vintia

INSPIRED ACCESS

Technical Documentation

Recreatex API

Documentation

API Change Log

Introduction

The **Change Log** provides an overview of all changes made to the web services.
This will make it more manageable for integrators of the web services to upgrade to a more recent version.

Legend

Type	Description
New	A new action is added
Bug	A bugfix
Improvement	An existing functionality is improved but the behaviour hasn't changed
Changed	An existing functionality it's behaviour has changed.

Example

Version	Date	Type	Method	Summary	Technical Details	Related JIRA Issue
EXAMPLE 3.4.0	29 Apr 2015	*Changed*	*Listcountries*	*Added Filter Parameter*	*A new Name Filter (string) is added to the Criteria*	***TEST-55***

Overview

Version	Change log
3.4.0	API Change Log - 3.4.0
3.4.1	API Change Log - 3.4.1
3.5.0	API Change Log - 3.5.0
3.5.2	API Change Log - 3.5.2
3.6.0	API Change Log - 3.6.0
3.6.1	API Change Log - 3.6.1
3.7.0	API Change Log - 3.7.0
3.7.1	API Change Log - 3.7.1
3.8.0	API Change Log - 3.8.0
3.8.1	API Change Log - 3.8.1
6.0.0	API Change Log - 6.0.0
6.0.1	API Change Log - 6.0.1
6.1.0	API Change Log - 6.1.0
6.1.1	API Change Log - 6.1.1
6.2.0	API Change Log - 6.2.0
6.2.1	API Change Log - 6.2.1
6.3.0	API Change Log - 6.3.0
6.3.1	API Change Log - 6.3.2

6.4.0	API Change Log - 6.4.0
6.4.1	API Change Log - 6.4.1
6.4.2	API Change Log - 6.4.2
7.0.0	API Change Log - 7.0.0
7.4.0	API Change Log - 7.4.0

API Change Log - 8.3.0

Version	Date	Type	Method	Summary	Technical Details	Related JIRA Issue
8.3.0	28 Aug 2023	Edit	FindSubscriptions	Added "ModifiedUntil" parameter to criteria		BACKOFFICE-22527 - Add "modifiedUntil" option in API call for findsubscriptions CLOSED
8.3.0	29 Aug 2023	Edit	All Basket endpoints	Added 3 properties to BasketPayment - OrderId - PayId - TrxId		BACKOFFICE-22711 - API - Enviso pay split payment: implement in Webshop CLOSED
8.3.0	29 Aug 2023	Edit	FindPersons GetPersonLLVInfoByLessonGroup	Added "Email" property to PersonLLVInfo		BACKOFFICE-22714 - CLONE - LLV voor lesgevers: emailadres klant CLOSED
8.3.0	30 Aug 2023	Edit	FindExpositions FindExpositionPrices	- Expositions - Exposition - Prices - ExpositionPrice - **ArticleGroupId** - **ArticleGroupName** - **HideItemGroups**		BACKOFFICE-22712 - API - EXPO: Allow grouping the tickets in an exhibition CLOSED
8.3.0	30 Aug 2023	Edit	FindExpositions FindExpositionArticlePrices	Added "AdditionalInformation" to ExpositionArticleOptional.		BACKOFFICE-21673 - Extra info toevoegen op webshop bij optioneel artikel (API) CLOSED
8.3.0	05 Sep 2023	Add	GetCashReportJson			BACKOFFICE-22463 - Support printing salesshift report on Adyen castle device => change API CLOSED

8.3.0	28 Sep 2023	Edit	Basket endpoints	Added new property to PlaceReservation • ContactId • Guid?		BACKOFFICE-22839 - API - BOOKINGS: Possibility to select a contactperson while making the reservation CLOSED
8.3.0	29 Sep 2023	Edit	FindPersonContacts	Added new property to Contact • PersonId • Guid		BACKOFFICE-22839 - API - BOOKINGS: Possibility to select a contactperson while making the reservation CLOSED
8.3.0	02 Oct 2023	Edit	FindSubscriptionArticles	Added new property to Criteria Includes • CardCreations • bool Added new property to SubscriptionArticle • CardCreations • List<CardCreation>		BACKOFFICE-22827 - Extend FindSubscriptionArticles: Add list of CardCreations to each Article in response CLOSED
8.3.0	06 Oct 2023	Edit	FindSales	Added new property "RefundReference" to Criteria		BACKOFFICE-22917 - Add SplitReference property to BasketPayment + SetSaleRefundStatus should look at SalesPaymentLines.RefundReference CLOSED
8.3.0	06 Oct 2023	Edit	Basket endpoints	Added 2 new properties to BasketPayment • SplitReference • RefundReference		BACKOFFICE-22917 - Add SplitReference property to BasketPayment + SetSaleRefundStatus should look at SalesPaymentLines.RefundReference CLOSED
8.3.0	11 Oct 2023	Edit	GetPointOfSales	Added a new property to PointOfSale • DivisionId • Guid		BACKOFFICE-19374 - Mpos: books in the sales point division instead of the division linked to shopID CLOSED

8.3.0	06 Nov 2023	Edit	FindExpositions FindExpositionPrices	Added 5 properties to ExpositionPrice FindExpositions: - Expositions - Prices - ArticleInfo1 - ArticleInfo2 - ArticleInfo3 - ArticleInfo4 - ArticleInfo5 FindExpositionPrices - ExpositionPrices - ArticleInfo1 - ArticleInfo2 - ArticleInfo3 - ArticleInfo4 - ArticleInfo5		BACKOFFICE-23023 - FindExpositionPrices: Add Info1 to Info5 to ExpositionPrice contract CLOSED

API Change Log - 8.3.1

Version	Date	Type	Method	Summary	Technical Details	Related JIRA Issue
8.3.1	03 Nov 2023	Edit	FindVendingMachineExpositions	Added list of ExpositionSubscriptionItems to each VendingMachineExposition		BACKOFFICE-23015 - Exhibitions - required membership on exhibition level -> KIOSK (API) CLOSED
8.3.1	03 Nov 2023	Edit	FindArticles	Added two booleans: - AllowPriceChangeWeb - AllowPriceChangeWebshop		BACKOFFICE-23021 - CLONE - Cadeaubon vrij te kiezen bedrag verkopen via webshop CLOSED
8.3.1	06 Nov 2023	Edit	FindPersons	Added OrganisationNumber to the person object		BACKOFFICE-23018 - Aansluiting Verenigingsloket: nieuw veld 'Organisatie nummer' (API) CLOSED

8.3.1	14 Nov 2023	Edit	FindPerson	Added OrganisationNumber to the search criteria • Request • Criteria • OrganisationNumber		BACKOFFICE-23069 - API - Aansluiting Verenigingsloket CLOSED
8.3.1	14 Nov 2023	Edit	FindExpositions	Added ExpositionTimePeriodCategories to the response		BACKOFFICE-23009 - EXPO: Group periods (API) CLOSED
8.3.1	22 Nov 2023	Edit	GetVendingMachineSettings	Added new property “AddNewMembershipsToExistingCard“ to VendingMachineSettings		BACKOFFICE-23089 - The possibility to sell new subscriptions via the self-service kiosk (API) CLOSED
8.3.1	22 Nov 2023	Edit	FindVendingMachineArticles	Added new property ForMembershipModule to FindVendingMachineArticlesSearchCriteria		BACKOFFICE-23089 - The possibility to sell new subscriptions via the self-service kiosk (API) CLOSED
8.3.1	23 Nov 2023	EDIT	FindEmployees	Add the following: • Criteria • **TaskId** • Includes • IncludeEmployeeSkills • Response • Employee • **IsSupervisor** • **EmployeeSkills** • **Id** • **SkillSubCategoryID** • **SkillCategoryID** • **EmployeeSkillObtained** • **EmployeeSkillExpires** • **EmployeeSkillComment** • **SkillSubCategoryShortName** • **SkillSubCategoryDescription** • **SkillSubCategoryLevel** • **SkillCategoryShortName** • **SkillCategoryDescription** • **SkillCategoryMandatory** • **SkillsID** • **SkillsShortName** • **SkillsDescription**		BACKOFFICE-22854 - TASKFLOW 2.0 : Add 'webtask'-webservices in WSDL CLOSED

8.3.1	24 Nov 2023	EDIT	FindEmployeeTasks	Added the following: • Criteria • Includes • **Availability** • Response • EmployeeTasks • **IsAvailable**		BACKOFFICE-22854 - TASKFLOW 2.0 : Add 'webtask'-webservices in WSDL CLOSED
8.3.1	24 Nov 2023	Add	FindEmployeeAvailabilities	Created a new call with the following: • Criteria • EmployeeId • From • Until • Response • List<EmployeeAvailability> • Id • EmployeeID • From • Until • Comment		BACKOFFICE-22854 - TASKFLOW 2.0 : Add 'webtask'-webservices in WSDL CLOSED
8.3.1	27 Nov 2023	Add	SaveEmployeeAvailability	Created a new call with the following: • Criteria • EmployeeAvailability • Id • EmployeeID • From • Until • Comment • Response • Result • Success • Message	Id is filled in if u want to update an existing one. leave empty to create a new one.	BACKOFFICE-22854 - TASKFLOW 2.0 : Add 'webtask'-webservices in WSDL CLOSED
8.3.1	27 Nov 2023	Add	DeleteEmployeeAvailability	Created a new call with the following: • Criteria • EmployeeAvailabilities (List<Guid>) • Response • Result • Success • Message		BACKOFFICE-22854 - TASKFLOW 2.0 : Add 'webtask'-webservices in WSDL CLOSED
8.3.1	27 Nov 2023	Add	GetTaskFlowSettings	Created a new call with the following • Response • WebTaskCanBePicked • EmployeeCanCancelConfirmedTasks		BACKOFFICE-22854 - TASKFLOW 2.0 : Add 'webtask'-webservices in WSDL CLOSED
8.3.1	27 Nov 2023	Edit	GetDigitalWallets	Added new includes all boolean? • OnlyTransactionsToday • ArrangementArticles • ComponentArticles • ZeroAmountLines • CancelledRecords		BACKOFFICE-23113 - CLONE - API KIOSK DIGIWALLET OVERVIEW IMPROVEMENT CLOSED

8.3.1	29 Nov 2023	Edit	GetVendingMachineTickets SetVendingMachineTicketsPrinted	When a startdate is available this is retrieved and added to the vendingMachineTickets The VendingMachineTicketsPrintedCriteria has a new bool? PrintOnlyToday	When PrintOnlyToday is not null an extra where is taken into account for setting PrintCount	**BACKOFFICE-23133** - CLONE - Add configuration on 'Print by order number' -flow **CLOSED**
8.3.1	13 Dec 2023	Add	HandleChannelEventByPlugin	When a channel input event (button pressed, magnetic loop, light on, etc) from a GT7 or PLC or other access control device, we can pass it from the RCX platform plugin towards the RCX Webservices, so that custom plugins of type IInputChannelListener could handle it		**PLUGINS-2045** - Meise - Parking solution RCX **CLOSED**
8.3.1	13 Dec 2023	Edit	FindArticles FindSalesZoneArticles	• FindArticles ◦ Added ShortName to Translations • FindSalesZoneArticles ◦ Bring out Translations		**BACKOFFICE-23209** - FindSalesZoneArticles: Add all translations to response **CLOSED**

API Change Log - 8.3.2

Version	Date	Type	Method	Summary	Technical Details	Related JIRA Issue
8.3.2	18 Jan 2024	Edit	ListPaymentMethods	Added property “TerminalPaymentMethodType” to BasketPaymentMethod	TerminalPaymentMethodType = CZAMType	https://gantner.atlassian.net/browse/BACKOFFICE-23351
8.3.2	25 Jan 2024	Edit	FindSalesZoneArticles	• Unified “Translation” contracts ◦ ArticleTranslation Translation ◦ ArticleGroupTranslation Translation • Added Translations to ArticleOption & ArticleOptionValue		https://gantner.atlassian.net/browse/BACKOFFICE-23399
8.3.2	12 Feb 2024	Edit	ListCountries	Added the CountryGroupId to the result		https://gantner.atlassian.net/browse/BACKOFFICE-23520

8.3.2	12 Feb 2024	Edit	FindExpositions	Added the finalRegistration to the exposition object - Id - Shortname - Name - ExpositionId - CountryGroupId - RegistrionClosurePeriodInMinutes - List of countries		https://gantner.atlassian.net/browse/BACKOFFICE-23520
8.3.2	14 Feb 2024	Edit	- FindPersonSubscriptionData - FindSubscriptionArticles - FindSubscriptions - GetValidSubscriptionsForAddressCard	Added the HigherPriceAtSale to the response		https://gantner.atlassian.net/browse/BACKOFFICE-23019
8.3.2	15 Feb 2024	Edit	Basket endpoints	Added property “OptionsPerQuantity“ to ArticleSale	Will multiply your ItemOptions with Quantity	https://gantner.atlassian.net/browse/BACKOFFICE-23535
8.3.2	26 Feb 2024	Edit	FindExpositions	Added property MinimumAge to Exposition		https://gantner.atlassian.net/browse/BACKOFFICE-23646
8.3.2	26 Feb 2024	Edit	- FindExpositions - FindExpositionArticlePrices - FindVendingMachineExpositionPeriodPrices	Added property MaximumQuantity for automatic articles		https://gantner.atlassian.net/browse/BACKOFFICE-23651
8.3.2	26 Feb 2024	Edit	FindExpositions	Added property itemGroupsSelected		https://gantner.atlassian.net/browse/BACKOFFICE-23657
8.3.2	04 Mar 2024	Edit	ListSalesSeriesOrders	Remake SalesSeriesOrder - List<SalesSeriesOrderItem> - List<SalesSeriesOrderOption> - List<SalesSeriesOrderPart>		BACKOFFICE-23463 - API: Extend the findSalesSeriesOrders api call CLOSED

API Change Log - 8.4.0

Version	Date	Type	Method	Summary	Technical Details	Related JIRA Issue
8.4.0	15 Mar 2024	Edit	FindCombiExpositions	Added property to CombiExpositionExpositions - CombiExpositionStepId Added extra select statement to get the CombiExpositionSteps		BACKOFFICE-23835 - CLONE - Combi exposition extensions: add exposition choice per step (Webshop) CLOSED
8.4.0	22 Mar 2024	Edit	All methods are described in the ticket	Added SMSAllowed		BACKOFFICE-23897 - CLONE - SMS: add new field 'SMSAllowed' on Address-level (webshop) CLOSED
8.4.0	02 Apr 2024	Edit	All Articles endpoints	Added Info fields - Info1 - Info2 - Info3 - Info4 - Info5		BACKOFFICE-23956 - API: Bring out info fields in FindArticles CLOSED
8.4.0	03 Apr 2024	Edit	FindPersonCards	Add “IncludeBlocked” to criteria		BACKOFFICE-23958 - FindPerson Cards: Criteria to exclude blocked CLOSED

8.4.0	11 Apr 2024	Edit	FindCombiExpositions CalculateCombiExpositionReservation	Added a list of Optional Article on combi ticket level to the result The list of SelectedCombiExpositionTicket has a list of Optional articles in the result		BACKOFFICE-23970 - CLONE - Combi exposition extensions: optional articles on ticket-level (webshop) CLOSED
8.4.0	15 Apr 2024	Edit	FindCombiExpositions	Added a list of expositionTimePeriodCategory to the CombiExpositionExpositions		BACKOFFICE-24039 - CLONE - Combi exposition extensions: other visualization for overlapping timeslots CLOSED
8.4.0	16 May 2024	Add	SaveEmployeeAvailabilities	Creates/Updates a list of EmployeeAvailabilities		BACKOFFICE-24009 - TASKFLOW 2.0 : Create & update availability in bulk to the API (recurring availability slot) CLOSED

Technical documentation - Recreatex API

1 Introduction

The ReCreateX Web Shop Webservices allow developers to create an own Web shop on top of an existing ReCreateX system. The Webservices abstract the ReCreateX domain logic and database into a simple interface.
The Webservices allow you to fill a basket (shopping basket) in order to finally register a check-out (sale).
The Webservices are implemented as SOAP Webservices to allow for interoperability.Some Webservice operations are also implemented as REST Webservices in order to avoid particular overhead (e.g. access to binary data such as images).
This document contains no profound details on the Webservices but only the general principles.For more details, we refer to the appropriate documents.

2 General Principles

Creating basket items

Creating basket items is the responsibility of the Webservices' client.
In order to create a basket item, however, certain start data a required (dependent on the type of basket item). There are two ways to get the necessary information:

- Browse
- Search

The Webservices give you sufficient operations to allow for both methods. Globally, it comes down to:

- Calling up lists of categories/types/ ... that can be used as search criteria.
- Looking up items using search criteria.Paging is always supported if large lists of results are possible.

Security

- Token based security
  - Activate the token based security
  - Build a token
- Encryption / Decryption algorithm.
  - Example key
- Encryption in Java
  - RSA Modulus (to use in Java)

Token based security

In version 6.3.0, token based security has been added to the WSDL.

> ⚠ **Password security**
> The token based security is built on top of the existing password security.
>
> That means that when a customer decides to use the token based security, a password is still required. The password will be used to generate the token.

Activation
When token based security is activated, one must be sure that **all the clients** are properly configured.

Activate the token based security

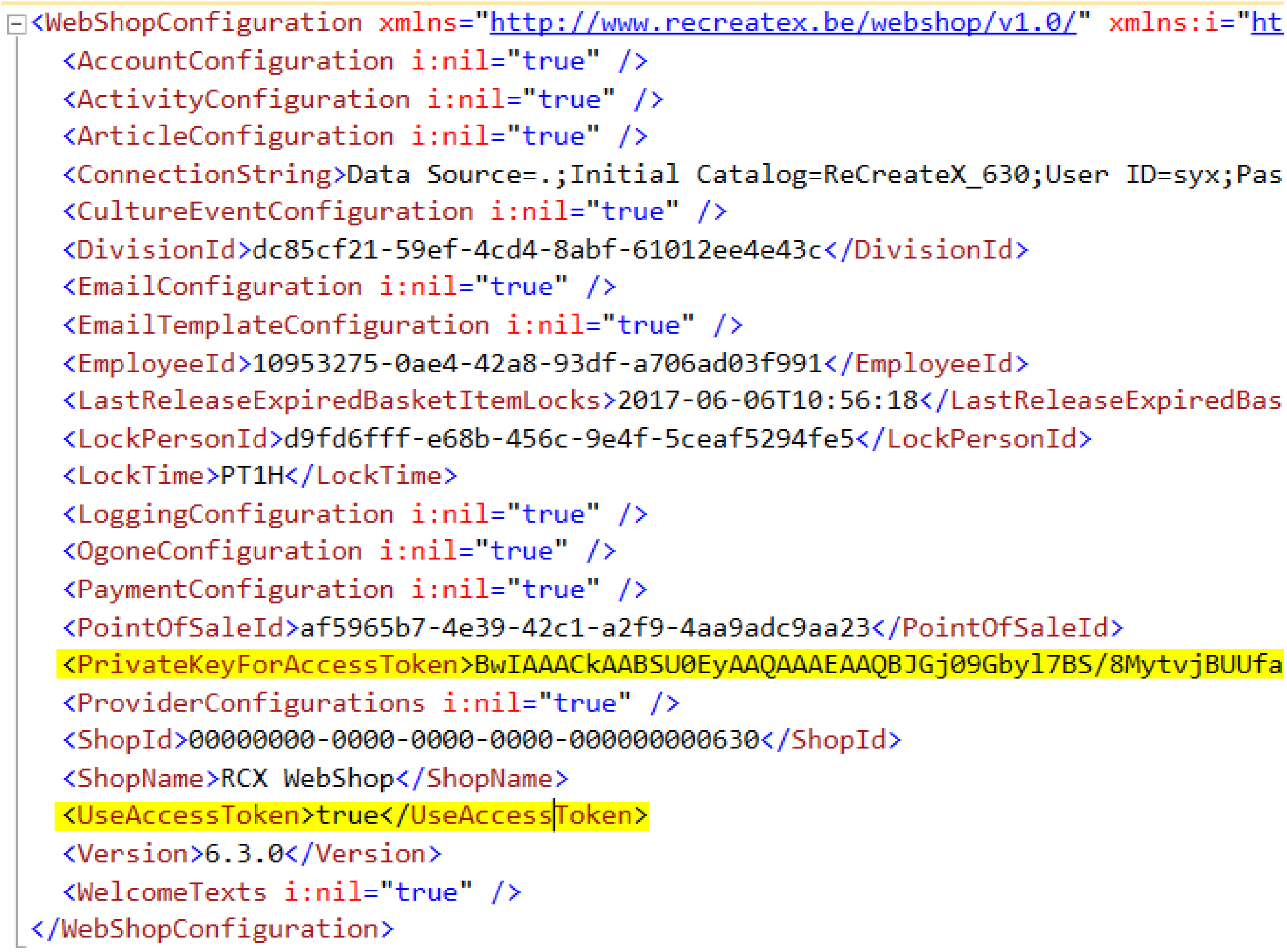
The configuration of the token based security is done in the 'WebShopConfiguration' that is stored in the WebShopConfiguration database.

Configuration

```
SELECT [Configuration]
  FROM [WebShopConfiguration].[dbo].[WebShopConfigurations]
  where ShopID = '00000000-0000-0000-0000-000000000630'
```

To activate the token based configuration, relevant fields should be configured

Field name	Description
UseAccessToken	Determines whether the token based security is activated. If that is the case, the access token must be passed with each request.
PrivateKeyForAccessToken	Access tokens are encrypted with an assymetric encryption algorithm before they are sent on the wire. A private key is used by the server (WSDL) to decrypt the access token. The client must encrypt the token with the public key.



```
<WebShopConfiguration xmlns="http://www.recreatex.be/webshop/v1.0/" xmlns:i="ht
  <AccountConfiguration i:nil="true" />
  <ActivityConfiguration i:nil="true" />
  <ArticleConfiguration i:nil="true" />
  <ConnectionString>Data Source=.;Initial Catalog=ReCreateX_630;User ID=syx;Pas
  <CultureEventConfiguration i:nil="true" />
  <DivisionId>dc85cf21-59ef-4cd4-8abf-61012ee4e43c</DivisionId>
  <EmailConfiguration i:nil="true" />
  <EmailTemplateConfiguration i:nil="true" />
  <EmployeeId>10953275-0ae4-42a8-93df-a706ad03f991</EmployeeId>
  <LastReleaseExpiredBasketItemLocks>2017-06-06T10:56:18</LastReleaseExpiredBas
  <LockPersonId>d9fd6fff-e68b-456c-9e4f-5ceaf5294fe5</LockPersonId>
  <LockTime>PT1H</LockTime>
  <LoggingConfiguration i:nil="true" />
  <OgoneConfiguration i:nil="true" />
  <PaymentConfiguration i:nil="true" />
  <PointOfSaleId>af5965b7-4e39-42c1-a2f9-4aa9adc9aa23</PointOfSaleId>
  <PrivateKeyForAccessToken>BwIAAACkAABSU0EyAAQAAAEAAQBJGj09Gbyl7BS/8MytvjBUUfa
  <ProviderConfigurations i:nil="true" />
  <ShopId>00000000-0000-0000-0000-000000000630</ShopId>
  <ShopName>RCX WebShop</ShopName>
  <UseAccessToken>true</UseAccessToken>
  <Version>6.3.0</Version>
  <WelcomeTexts i:nil="true" />
</WebShopConfiguration>
```

Build a token

If token based security is enabled, the client must generate a token and pass that with each request. The token is valid for 30 minutes.

The client can choose whether it

- generates a new token for each request
- re-uses a token for a certain time (max. validity is 30 minutes).

The token based security is built on top of the existing password security.

The client/customer should have received

- a public key which is needed to encrypt the token
- a secret password which is used to built the token

Steps

- The client must create the token-content.
- The token-content concatenates the secret password and the number of ticks that represents the current UTC date. A colon should be used as seperator.
- The next step is to encrypt the token with the public key.
- Finally, the encrypted token is encoded to a base64 string.

GenerateToken

```
    class Client
    {
        private readonly string _publicKey;
        private readonly string _password;

        public Client(string publicKey, string password)
        {
            _password = password;
            _publicKey = publicKey;
        }

        public string GenerateToken()
        {
            var completeToken = string.Format("{0}:{1}", _password,
DateTime.UtcNow.Ticks);
            var encryptedToken = EncryptionHelper.Encrypt(_publicKey,
completeToken);
            var token = Convert.ToBase64String(encryptedToken);
            return token;
        }
    }
```

Encryption / Decryption algorithm.

The encryption algorithm that is used to encrypt/decrypt the access token is based on RSA.

The code sample below can be used to generate private and public key pairs.

Encryption algorithm

```
    class EncryptionHelper
    {
        public static Tuple<string, string> CreateKeyPair()
        {
            var cspParams = new CspParameters { ProviderType = 1 };

            var rsaProvider = new RSACryptoServiceProvider(1024,
cspParams);

            var publicKey = Convert.ToBase64String(rsaProvider.
ExportCspBlob(false));
            var privateKey = Convert.ToBase64String(rsaProvider.
ExportCspBlob(true));

            return new Tuple<string, string>(privateKey, publicKey);
        }

        public static byte[] Encrypt(string publicKey, string data)
        {
            var cspParams = new CspParameters { ProviderType = 1 };
            var rsaProvider = new RSACryptoServiceProvider(cspParams);

            rsaProvider.ImportCspBlob(Convert.FromBase64String
(publicKey));

            var plainBytes = Encoding.UTF8.GetBytes(data);
            var encryptedBytes = rsaProvider.Encrypt(plainBytes, false);

            return encryptedBytes;
        }

        public static string Decrypt(string privateKey, byte[]
encryptedBytes)
        {
            var cspParams = new CspParameters { ProviderType = 1 };
            var rsaProvider = new RSACryptoServiceProvider(cspParams);

            rsaProvider.ImportCspBlob(Convert.FromBase64String
(privateKey));

            var plainBytes = rsaProvider.Decrypt(encryptedBytes, false);

            var plainText = Encoding.UTF8.GetString(plainBytes, 0,
plainBytes.Length);

            return plainText;
        }
    }
```

Example key

For testing reasons, it can be appropriate to have a private and public key that can be used to encrypt/decrypt the token.

Key	Value
Private key	BwIAAACkAABSU0EyAAQAAAEAAQBJGj09Gbyl7BS/8MytvjBUUfaktW984VHHW4lSI9y2OwaeOq4qqSD6IOHU9HL/QtwZ+wELq28eAOQSnr11hifMf6zWjIsCBHOEpLNJjL3wxjl7dUBEGMJOeZj2rmcf8v7lP/rpAtO/G8wKXhAIKLIFxcChkXwQKbQbHQ/FtX2bwn/CHUScK/rzYp9RRXv0dn7BwFgbs918/1CS8ZACOUSTGakdb72x/YIdtQ3bW1jnRigrnzljLf+J4xju5WJP+sg3L12EFJzY5ytr1YU7IYLNXr6FU+x037nheJXiiA9t9BGYaP8BJ2ZanwIgG1/rHz1xQODRcOJzcEnA1BoNteL3q8e4lYGXw56FcvO0JJyR0FoS93j5IDp/D0KO73u6fUD4RQ6NKQei61cklrX4PcmqUyRZRSov7mK7ygr4YaegGm1ddRkQv+X7OadXCQgorbeRWrAksz7mAd/a+Higu8zsFhAj/3Q7JkwKUxQ1gO4KzeptxpO5nhvfIfyb5B5duLCZgH5OnEWPar0xIyH7uWHSPE0KUMhKadn2dshVQwJGzJQKk/K9/v6of6s+Q2wqjlLuB8PcUYWo5sc6aVXdrqCZdVV1WyAOdE+Yizy0kFpC99kjXAGWaJz94rlJJyj6LhCo7DjNc1SufRRfmbFKDO5XLJLZDQGDotsF6cvej+o0B6KgytK6ibImOtklx3+vnMnktOKMVWiHBQ/keOzWUWy6sbZZCZImLjvykV8+gR1priK5lp4Nn+TQJ+/ZJUJCuJs=
Public key	BgIAAACkAABSU0ExAAQAAAEAAQBJGj09Gbyl7BS/8MytvjBUUfaktW984VHHW4lSI9y2OwaeOq4qqSD6IOHU9HL/QtwZ+wELq28eAOQSnr11hifMf6zWjIsCBHOEpLNJjL3wxjl7dUBEGMJOeZj2rmcf8v7lP/rpAtO/G8wKXhAIKLIFxcChkXwQKbQbHQ/FtX2bwg==

Encryption in Java

For implementation in Android and other reasons it may be useful to write the encryption in Java. The code to be used in that case is :

```
import java.util.Base64;

String password = "RCX7.1.1";
String modulusString = "wpt9tcUPHRu0KRB8kaHAxQWyKAgQXgrMG7
/TAun6P+X+8h9nrvaYeU7CGERAdXs5xvC9jEmzpIRzBAKLjNasf8wnhnW9nhLkAB5vqwsB+x
ncQv9y9NThIPogqSquOp4GO7bcI1KJW8dR4XxvtaT2UVQwvq3M8L8U7KW8GT09Gkk=";
String publicExponentString = "AQAB";

String time = "636917740295939460";
String token = password + ":" + time;

byte[] modulusBytes = DatatypeConverter.parseBase64Binary
(modulusString);
byte[] exponentBytes = DatatypeConverter.parseBase64Binary
(publicExponentString);

BigInteger modulus = new BigInteger(1, modulusBytes);
BigInteger publicExponent = new BigInteger(1, exponentBytes);

RSAPublicKeySpec rsaPubKey = new RSAPublicKeySpec(modulus,
publicExponent);
KeyFactory fact = KeyFactory.getInstance("RSA");
PublicKey pubKey = fact.generatePublic(rsaPubKey);

Cipher cipher = Cipher.getInstance("RSA");
cipher.init(Cipher.ENCRYPT_MODE, pubKey);
byte[] encryptedBytes = cipher.doFinal(token.getBytes("UTF-8"));

String encryptedStringBase64 = DatatypeConverter.printBase64Binary
(encryptedBytes);
System.out.println(encryptedStringBase64);
```

The modulusString is the corresponding of the example key here above. See next session to obtain the modulus of your public key.

When changing the time for the current time, make sure it is in UTC time zone in ticks.

RSA Modulus (to use in Java)

In Java you need the modulusString and exponent of the (C#) publicKey. It can be obtained through :

```
var cspParams = new CspParameters {};
var rsaProvider = new RSACryptoServiceProvider(cspParams);
rsaProvider.ImportCspBlob(Convert.FromBase64String(publicKey));

Console.WriteLine(rsaProvider.ToXmlString(false));
```

(This can be done separately or in the encrypt method.)

```
<RSAKeyValue>
        <Modulus>wpt9tcUPHRu0KRB8kaHAxQWyKAgQXgrMG7
/TAun6P+X+8h9nrvaYeU7CGERAdXs5xvC9jEmzpIRzBAKLjNasf8wnhnW9nhLkAB5vqwsB+x
ncQv9y9NThIPogqSquOp4GO7bcI1KJW8dR4XxvtaT2UVQwvq3M8L8U7KW8GT09Gkk=<
/Modulus>
        <Exponent>AQAB</Exponent>
</RSAKeyValue>
```

The basket operations

The **basket** contains all items a Web shop customer wants to buy (rent, book, ...). An item that can be added to the basket is called a **basket item.** Dependent on the ReCreateX modules you want to/can use, there are different types of basket items:

- The purchase of an article
- The rental of an article
- The registration for an activity
- …

It finally comes down to filling the basket with basket items in order to register a sale in the end **(check-out)**.
The content of the basket is the responsibility of the Webservice's users (clients).Only after check-out the basket is turned into a sale.

Check-out

This is the only operation that is really necessary to make a sale.First, the basket is validated and if no validation errors occur, the sale is processed.
If validation errors do occur, however, these first have to be solved after which the check-out operation has to be restarted.

Validation

Not all baskets can be sold. Certain domain rules can cause the sale of the basket to be cancelled. The validation of the basket consists of two parts:

- Validation of all basket itemse.g. the maximal number of articles to be bought has been exceeded
- Validation on the level of the basket as a wholee.g. the maximal sales amount has been exceeded

The result of a validation (of the basket as a whole or of an individual basket item) holds two characteristics:

- The name of the domain/business rule that has been violated.
- A text that indicates what it is about.

Practically, it will come down to visualising the text to the user so that he can take the necessary steps. (e.g. delete the basket item from the basket).If this is insufficient (for clients of the Webservice), you can take other actions on the basis of the name of the rule. (there is currently no list of rules available)

Validating basket items is an optional step that allows to already give feedback to the buyer before he checks out the basket completely.
Per basket item maximally one violated rule will be returned. It is thus possible that if a violated rule has been solved, a new violated rule will be found after validation.

Lock and Unlock basket items

It often happens that a **sales option** is taken on a basket item before the basket is checked out. Let's clarify this on the basis of an example in which we buy an article (and we keep the stock of those articles). We want to prevent that the buyer isn't informed until check-out that the article is no longer available (although there was sufficient stock on the moment he added his article to the basket). In order to prevent this, it is possible **to lock an article** (or other basket item) **for a certain time.** In case of an article, the stock is temporarily reduced.
The lock operation returns a **lock ticket** for each basket item. This ticket is a prove that a basket item has been locked.
The lock has an **expiry time.** The expiry times ensures that the lock system remains scalable.
The lock tickets have to be added to the basket at check-out. If a lock is already expired, the basket item is processed as if it didn't have a lock.
The **unlock** operation is an optional operation that allows you to release a lock earlier.
Locks are always automatically unlocked after the expiry time. If possible, however, it is always better to unlock an unnecessary lock as soon as possible.

The Service Context

All request to the Webservice contain a **service context.** The service context determines the context in which the Webservices will function. Since the Webservices can be configured for several ReCreateX Systems (databases), it may be necessary to specify which system you want to contact.If the Webservices operate in the context of only one ReCreateX system, the context is not relevant.
At this moment, the request consists of the following properties:

Property	Description	Comment
ShopID	Defines the unique shop identifier for the customer. All ReCreateX configuration will be done by the shop id. If not filled in, the default ShopID specified in the Webservice configuration file (web.config) will be used.	Mandatory
~~**DivisionId**~~	~~Defines the ReCreateX division id which will be used for API calls.~~ ~~*Only applicable when no ShopID is provided. (Recommended approach is using the ShopID!).*~~	Obsolete
Language	ISO639-1 language code which is used for getting all translatable ReCreateX records. *If not filled in, the default language will be used (NL)*	Optional
SessionId	If the basket functionality is used, a unique GUID per **shopping basket** has to be filled in under SessionID! If two different users work simultaneously on the same Website, it is necessary to know which requests belong to which shopping basket.	Mandatory when using basket
Origin	This property is introduced to allow the service to act differently depending on the origin. Origin is an enumerator (ServiceOrigin) with the available values: Default, Internet, VendingMachine, Desktop. *Ex: The values default & internet will check for all web dependencies (for example show on website).*	Optional
Password	Additional password security can be activated. When activated, we require that the password is passed in every API request.	Mandatory when activated
AccessToken	Security with access token can be activated. When activated, the token should be passed in every API request.	Mandatory when activated
Encode	When activated, all value content will be exposed HTML encoded.	Optional

Use of the domain objects

The Webservice operations always work with domain objects (underlyingly usually represented as XML) Such a domain object can contain a lot of useful information if sent back to the client in a response. If, however, a domain object is sent as a reference in the request, most information it holds will be irrelevant and it will be **sufficient to only fill in the identification of the object (ID property).**
e.g. If you call up all article categories, you will see in each category object an ID, a name, a description and optionally an image (all useful information). If you, however, want to look for articles in a specific article category, it is sufficient to only fill in the ID of the group in question in the object.

Versioning

This section only applies to the SOAP service (not to the JSON service).

Namespace

Previously, the version of the API was included in the namespace of the API.

Sample

As from version 6.0, the namespace does no longer contain the version of the API.

Sample

Documentation

The current version of the API can still be discovered within the documentation tag of the IWebShop contract definition:

Sample

```
<wsdl:portType msc:usingSession="false" name="IWebShop">
        <wsdl:documentation>
                <summary>
                        ReCreateX Web Shop Web Services (Contract
version 4.0)
                </summary>
        </wsdl:documentation>
```

API call

The version number can also be retrieved programmatically by calling the **GetCurrentWsdlVersion** method.
See GetCurrentWsdlVersion for more information.

3 General Webservices

The text below will discuss in detail how each Webservice operation has to be used.

Paging Criteria

This is a standard object that will be used to allow page numbers on Web pages.
The object is set up as follows.

Name	Type	Description	Remarks
PageIndex	Integer	The current page number.	Default = 0
PageSize	Integer	The number of items that has to be visualised.	Default = 10
Sorting	SortingCriteria	The way in which the results will be sorted.	

Sorting Criteria

Name	Type	Description	Remarks
PropertyName	String	The name of the property on which the system has to sort.	To be filled in obligatory.
IsAscending	Boolean	True if sorted in increasing order, false if sorted in decreasing order.	Default = false

xxxSpecifiedField

If a DotNet client is used to address the Webservices, it is important to also fill in the possible xxxSpecifiedField fields of the object if the linked fields (xxx) are used.
If you do not set this boolean to 'True', the value of the actual field will not be used.
This has to do with the way in which DotNet serializes the request and nullable types.
This does not have to be done with another programming language or environment (Java - PHP - etc.)

Note
As from API version 6.0, the xxxSpecifiedFields are no longer created in .Net when using a service reference!

4 General

AllowedToDeleteLockTicket

This call will return a boolean whether you are allowed to delete a lockticket or not

Request

The request consists of the Context and and the **AllowedToDeleteLockTicketCriteria** object.

Name	Type	Description	Remarks
LockTicketId	GUID	The ID of the lockticket you wish to check	

Response

The response contains a **AllowedToDeleteLockTicketresult** object.

Name	Type	Description	Remarks
AllowedToDeleteLockTicket	bool	This boolean will return true if you are allowed to delete the lockticket or false when you are not allowed to	

CreateCollectLaterLine

Calling up this Webservice operation will create a collect later line in the ReCreateX system

Request

The CreateCollectLaterLine consists of the service context object and a **CreateCollectLaterLineCriteria** object

CreateCollectLaterLineCriteria

Name	Type	Description	Remarks
QuantityToCollect	int	The quantity that must be collected	
QuantityCollected	int	The quantity that is already collected	
UniqueReferenceNumber	string	Unique reference number	
OrderDueDate	DateTime	When is the order overdue?	
ExternalItemCode	string	External item code	
Name	string	Name of the customer	
Street	string	Street of the customer	
City	string	City of the customer	
Country	string	Country of the customer	

Response

The response is a **CreateCollectLaterLineResult** object.

The **CreateCollectLaterLineResult** object looks as follows:

Name	Type	Description	Remarks
Succeeded	bool	Did the save succeed?	
ReasonForFailure	String	What is the reason for failure?	Only contains a value when *Succeeded* is false

FindReasons

Calling up this Webservice operation will load the reasons configured in ReCreateX.

Request

The FindReasons consists of the service context object and the **ReasonSearchCriteria** object.

ReasonSearchCriteria

Name	Type	Description	Remarks
ID	Guid?	Filter on the ID of the reason	
ReasonType	ReasonType?	Filter on the type	
Paging	PagingCriteria	Paging	
Includes	ReasonIncludes	Include certain data in the search	
PersonType	Enum PersonType	Filter on Person type	

ReasonIncludes

Name	Type	Description	Remarks
CancelCostSensitive	bool	Include the reasons that are cancel cost sensitive	
CancelCostPercentagePeriods	bool	Include the cancel cost percentage periods	

Response

The response is a collection of Reason objects.

The **Reason** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	The unique identifier on the reason	
Code	String	The code of the reason	
Name	String	The shortname of the reason	
ReasonType	ReasonType	The type of the reason	
CancelCostSensitive	bool	Indicates if the reason is cancel cost sensitive	
CancelCostAmount	decimal	Indicates what the cancel cost amount is	
CancelCostPercentagePeriods	IList<CancelCostPercentagePeriod>	An array of cancel cost periods for this specific reason	Only contains values when the ReasonType is of type Reservations (integer value 0)

The **CancelCostPercentagePeriod** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	Unique identifier	
Code	string	The code of the percentage period	
Description	string	The description of the percentage period	
FromLimit	bool	Indicates if the from limit is used	
UntillLimit	bool	Indicates if the until limit is used	

FromTimeScale	TimeScale	Defines what the timescale is	Enum object:
			Hours = 0,
			Days = 1,
			Weeks = 2,
			Months = 3

UntilTimeScale	TimeScale	Defines what the timescale is	See above
FromQuantity	int	Defines the value of the FromTimeScale	
UntilQuantity	int	Defines the value of the UntilTimeScale	
Percentage	int	The cancel cost percentage if this percentage period is used	
CustomerPriceGroupId	Guid	Defines for which price group this percentage period is applicable	

Example:

Example

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
FindReasonsResponse, ReCreateX.WebShop.WebServices.Contracts",
    "result": [
        {
            "$type": "ReCreateX.WebShop.WebServices.Contracts.Reason,
ReCreateX.WebShop.WebServices.Contracts",
            "code": "VERKEERDE BOEKING",
            "id": "9941b066-6211-e811-813f-a0481cb9dbf9",
            "name": "VERKEERDE BOEKING",
            "reasonType": 0,
                        "personType": 0,
            "cancelCostSensitive": true,
            "cancelCostAmount": 0,
            "cancelCostPercentagePeriods": [
                {
                    "$type": "ReCreateX.WebShop.WebServices.Contracts.
CancelCostPercentagePeriod, ReCreateX.WebShop.WebServices.Contracts",
                    "id": "d8b9a66e-6211-e811-813f-a0481cb9dbf9",
                    "code": "-24 u",
                    "description": "Annulatie binnen de 24u vooraf",
                    "fromLimit": true,
                    "untilLimit": false,
                    "fromTimeScale": 0,
                    "untilTimeScale": 0,
                    "fromQuantity": 24,
                    "untilQuantity": 0,
                    "percentage": 50,
                    "customerPriceGroupId": "e356fc61-dc00-4931-8d35-
2db59483d682"
                },
                {
                    "$type": "ReCreateX.WebShop.WebServices.Contracts.
CancelCostPercentagePeriod, ReCreateX.WebShop.WebServices.Contracts",
                    "id": "f2ec247a-6211-e811-813f-a0481cb9dbf9",
                    "code": "-1 maand",
                    "description": "Annulatie binnen de maand vooraf",
                    "fromLimit": true,
                    "untilLimit": true,
                    "fromTimeScale": 3,
                    "untilTimeScale": 0,
                    "fromQuantity": 1,
                    "untilQuantity": 24,
                    "percentage": 10,
                    "customerPriceGroupId": "e356fc61-dc00-4931-8d35-
2db59483d682"
                }
            ]
        }
    ]
}
```

In above example we see two cancel cost percentage periods:

- Annulation within 24 hours the user has to pay an annulation cost of 50% based on the original amount of the booking when the booking is cancelled 23 hours 59 minutes or less before the start of the booking
- Annulation within 1 month and +24 hours the user has to pay an annulation cost of 10% based on the original amount of the booking when the booking is cancelled 24 hours or more before the start of the booking

GetCurrentDivision

Calling up this Webservice operation will visualise the current division linked to the webshop configuration.

Request

The GetCurrentDivision consists of the service context object.

Response

The response is a Division object.

The **Division** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	Unique identifier of the division.	
Name	String	The name of the division	
Address	Address	The address of the division	
Phone	String	The phone of the division	
Fax	String	The fax of the division	
Email	String	The email of the division	
Website	String	The website of the division	
BankAccount	String	The bankaccount of the division	
Comments	String	The comments of the division	
Credential	Credential	The credential of the division	
IBAN	String	The iban of the division	
BIC	String	The bic of the division	

GetCurrentWsdlVersion

Calling up this Webservice operation will visualise the current version linked to the webshop configuration.
It will return both the version of the service (API) and the version of the storage (RCX).

Request

The GetCurrentDivision consists of the service context object and an empty criteria object (GetCurrentWsdlVersionSearchCriteria).

Response

The response is a GetCurrentWsdlVersionResult object.

The **GetCurrentWsdlVersionResult** object looks as follows:

Name	Type	Description	Remarks
ServiceVersion	String	Version number of the service.	= public version of the API
StorageVersion	String	Version number of the storage.	= public version of the RCX back office

> **Note**
> As from API 6.0, the service & storage version numbers will always be the same!

GetExternalMapping

Calling up this Webservice operation will return the mapping for an external id

Request

The GetExternalMapping consists of the service context object and the **GetExternalMappingCriteria** object.

GetExternalMappingCriteria

Name	Type	Description	Remarks
ExternalId	long	The external id	
Entity	string	The entity key	

Sample

```
{
  "Criteria": {
    "ExternalId": 100,
    "Entity": "CustomApp"
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000641",
    "Password": "MyVerySecretPassword"
  }
}
```

Response

The response contains a **ExternalMapping** object.

Name	Type	Description	Remarks
RcxId	Guid	The id in the ReCreateX system	
ExternalId	long	The external id	
Entity	string	The entity key	

Sample

```
{
    "ExternalMapping": {
        "RcxId": "40c19ddb-b0ee-e611-8160-78acc098f184",
        "Entity": "CustomApp",
        "ExternalId": 100
    }
}
```

GetInfoFieldCategories

Calling up this Webservice operation will load the configured info field categories configured in ReCreateX.

Request

The GetInfoFieldCategoriesRequest consists of the service context object.

Sample

```
{
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000641",
    "Password": "myVerySecretPassword"
  }
}
```

Response

The response is a collection of InfoFieldCategory objects.

The **InfoFieldCategory** object contains the following properties:

Name	Type	Description	Remarks
Id	Guid	Unique identifier	
Code	String	The code of the info field category	
Description	String	The description of the info field category	

Sample

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
GetInfoFieldCategoriesResponse, ReCreateX.WebShop.WebServices.
Contracts",
    "result": [
        {
            "$type": "ReCreateX.WebShop.Model.InfoFieldCategory,
ReCreateX.WebShop.Model",
            "id": "c281aa8b-5f06-e811-938a-78acc098f184",
            "code": "Measurements",
            "description": "The measurements of a specific person"
        },
        {
            "$type": "ReCreateX.WebShop.Model.InfoFieldCategory,
ReCreateX.WebShop.Model",
            "id": "652494ea-6d06-e811-938a-78acc098f184",
            "code": "Locatie",
            "description": "Locatie"
        }
    ]
}
```

GetInfoFields

Calling up this Webservice operation will load the configured info fields for a specific subject configured in ReCreateX.

Request

The GetInfoFields consists of the service context object and the **GetInfoFieldsCriteria** object.

GetInfoFieldsCriteria

Name	Type	Description	Remarks
SubjectId	Guid?	For which subject do you want to receive the info fields?	Optional
InfoFieldCategoryId	Guid?	For which category do you want to receive the info fields?	Optional
Includes	InfoFieldIncludes	Include additional information	

InfoFieldIncludes

Name	Type	Description	Remarks
Blocked	bool	Also include the blocked info fields	Default = false

Sample

```
{
  "Criteria": {
    "SubjectId": null,
    "InfoFieldCategoryId": null,
    "Includes": {
      "Blocked": false
    }
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000641",
    "Password": "myVerySecretPassword"
  }
}
```

Response

The response is a collection of InfoField objects.

The **InfoField** object is abstract and looks as follows:

Name	Type	Description	Remarks
Id	Guid	Unique identifier	
SubjectId	Guid	The subject id of the info field	
Code	String	The code of the info field	
Description	String	The description of the info field	
Blocked	bool	Is the info field blocked?	
CategoryId	Guid?	The ID of the linked category	If no category is linked, this value is NULL

SubjectType	InfoFieldSubjectType	The subject type for this specific InfoField	Person = 0,
			Place = 1,
			Infrastructure = 2

Based on which datatype is linked with the InfoField object, another class inherits from the InfoField object.

TextInfoField

The **TextInfoField** object contains the following additional properties:

Name	Type	Description
Value	string	Contains the value of the info field

NumericInfoField

The **NumericInfoField** object contains the following additional properties:

Name	Type	Description
Value	decimal	Contains the value of the info field

BooleanInfoField

The **BooleanInfoField** object contains the following additional properties:

Name	Type	Description
Value	bool	Contains the value of the info field

IntegerInfoField

The **IntegerInfoField** object contains the following additional properties:

Name	Type	Description
Value	integer	Contains the value of the info field

InfoFieldCategory

Name	Type	Description	Remarks
Id	Guid	Unique identifier	
Code	string	The code of the category	
Description	string	The description of the category	

Sample

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.GetInfoFieldsResponse, ReCreateX.WebShop.WebServices.Contracts",
    "result": [
        {
            "$type": "ReCreateX.WebShop.Model.NumericInfoField, ReCreateX.WebShop.Model",
            "value": 74,
            "id": "7b5e8bfa-9710-e811-938d-78acc098f184",
            "subjectId": "c80b0764-265b-4e56-900b-0fdfa2222e37",
            "code": "Weight",
            "description": "Weight",
            "blocked": false,
            "categoryId": "c281aa8b-5f06-e811-938a-78acc098f184",
            "subjectType": 0
        },
        {
            "$type": "ReCreateX.WebShop.Model.NumericInfoField, ReCreateX.WebShop.Model",
            "value": 176,
            "id": "824bdb00-9810-e811-938d-78acc098f184",
            "subjectId": "c80b0764-265b-4e56-900b-0fdfa2222e37",
            "code": "Height",
            "description": "Height",
            "blocked": false,
            "categoryId": "c281aa8b-5f06-e811-938a-78acc098f184",
            "subjectType": 0
        }
    ]
}
```

GetProductType

Request

Name	Type	Description
Context	ServiceContext	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
Criteria	GetProductTypeCriteria	The criteria used for GetProductType

GetProductTypeCriteria

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.

BasketItem	BasketItem	The BasketItem for which the ProductType should be retrieved

Result

GetProductTypeResult

Name	Type	Description	Technical Information
ProductType	String	Returns the configured ProductType	Currently one of 4 values • Empty string • foodanddrink • homeandgarden • giftsandflowers

SavePersonExternalMasterId

This call will save a new external master id to the given person

Request

The request consists of the Context and and the **SavePersonExternalMasterIdCriteria** object.

Name	Type	Description	Remarks
PersonId	GUID	The Id of the person you wish to edit	
ExternalMasterId	String	The external master id you wish to save	

Sample

```
{
  "Criteria": {
    "PersonId": "572BE298-86E8-E611-8497-4C72B941B3A7",
    "ExternalMasterId": "fqfsdf66633fdsf"
  },
  "Context": {
    "Language": "nl",
    "ShopId": "00000000-0000-0000-0000-000000000710"
  }
}
```

Response

The response contains a **SavePersonExternalMasterIdResult** object.

Name	Type	Description	Remarks
ValidationResults	ValidationResult	If there are any errors, they will appear here	

Sample

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
SavePersonExternalMasterIdResponse, ReCreateX.WebShop.WebServices.
Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
SavePersonExternalMasterIdResult, ReCreateX.WebShop.Model",
        "validationResults": [
            {
                "$type": "ReCreateX.WebShop.Model.ValidationResult,
ReCreateX.WebShop.Model",
                "_brokenRuleName": null,
                "message": null,
                "isValid": true
            }
        ]
    }
}
```

UpdateCollectLaterLine

Calling up this Webservice operation will update an existing collect later line in the ReCreateX system

Request

The UpdateCollectLaterLine consists of the service context object and a **UpdateCollectLaterLineCriteria** object

UpdateCollectLaterLineCriteria

Name	Type	Description	Remarks
OrderReference	string	Reference of the order	
ItemCode	string	The code of the item	
ItemQuantity	int	How many items are collected?	

Response

The response is a **UpdateCollectLaterLineResult** object.

The **UpdateCollectLaterLineResult** object looks as follows:

Name	Type	Description	Remarks
Succeeded	bool	Did the update succeed?	
ReasonForFailure	String	What is the reason for failure?	Only contains a value when *Succeeded* is false

GetShopIDConfiguration

Request

The request consists of the service context and has no criteria

Response

ShopIDConfigurationResult

Name	Type	Description	Remarks
Division	ConfigurationDivision	The configuration division linked to the shopid configuration	
Employee	ConfigurationEmployee	The configuration employee linked to the shopid configuration	
LastReleaseExpiredBasketItemLocks	DateTime	The date of last release expired locked basketitems	
LockPerson	ConfigurationPerson	The configuration person linked to the shopid configuration	
LockTime	string	A string of the locktime	
PointOfSale	ConfigurationPointOfSale	The configuration point of sale linked to the shopid configuration	
RemovePersonalData	Bool	A boolean for removing personal data	
ShopName	string	A string with the name of the shop	
Version	string	A string with the version number	

ConfigurationDivision

Name	Type	Description	Remarks
Division	Division	The division of the configuration	
DivisitonExists	Bool	Bool if the division exists in the database	

ConfigurationEmployee

Name	Type	Description	Remarks
EmployeeID	GUID	The employeeID of the configuration	
EmployeeExists	Bool	Bool if the employee exists in the database	

ConfigurationPerson

Name	Type	Description	Remarks
PersonId	GUID	The personId of the configuration	
PersonExists	Bool	Bool if the person exists in the database	

ConfigurationPointOfSale

Name	Type	Description	Remarks
PointOfSaleId	GUID	The point of sale id of the configuration	
PointOfSaleExists	Bool	Bool if the point of sale exists in the database	

ListSalesSeriesOrders

Calling up this Webservice operation will visualise the desired salesSeriesOrders on basis of the linked parameters.

Request

The request consists of the service context and a ListSalesSeriesOrdersSearchCriteriaobject.

ListSalesSeriesOrdersSearchCriteria

Name	Type	Description	Remarks
FromDate	DateTime	If filled in: Only salesSeriesOrders that have a salesHeaderdate after the given date will be shown.	
UntilDate	DateTime	If filled in: Only salesSeriesOrders that have a salesHeaderdate before the given date will be shown.	

Status	Int	If filled in: Only salesSeriesOrders for the given status will be shown.	
PointOfSaleId	GUID	If filled in: Only salesSeriesOrders for the given pointOfSaleId will be shown.	
SalesZoneId	GUID	If filled in: Only salesSeriesOrders for the given salesZoneId will be shown.	
SalesSerieID	GUID	If filled in: Only salesSeriesOrders for the given salesSerie will be shown.	
Includes	SalesSeriesOrderIncludes	This object will determine which characteristics of the salesseriesorders will be cadde up	

SalesSeriesOrderIncludes

Name	Type	Description	Remarks
ExtraInfo	Boolean	Include the items, option and parts	

Response

A list of salesSeriesOrders. A salesSeriesOrder consists of the following parts:

SalesSeriesOrder

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the salesSeriesOrder	
TokenNumber	int	The tokenNumber of the salesSeriesOrder	
SalesOrderStatus	SalesSeriesOrderStatus	The status of the salesSeriesOrder	
Note	string	The comment of the salesSeriesOrder	
OrderNumber	string	The orderNumber of the salesSeriesOrder	
Table	string	The table of the salesSeriesOrder	
OpenTime	DateTime	The time the salesSeriesOrder has been made	
ProcessedTime	DateTime	The time the salesSeriesOrder has been processed	
PickedUpTime	DateTime	The time the salesSeriesOrder has been picked up	
Items	List<SalesSeriesOrderItem>	A list of Items	
Options	List<SalesSeriesOrderoOption>	A list of option	
Parts	List<SalesSeriesOrderPart>	A list of parts of an arragement	

SalesSeriesOrderItem

A salesSeriesOrderItem is the item sold, this can be a normal item or an arrangement.

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the salesSeriesOrderitem	
Name	string	The name of the salesSeriesOrderitem	
Quantity	decimal	The quantity of the salesSeriesOrderitem	
IsArrangement	Boolean	Is the salesSeriesOrderitem an arrangementItem	
Note	string	The comment on the salesSeriesOrderitem	

SalesSeriesOrderOption

A salesSeriesOrderOption is an option for the item sold.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the salesSeriesOrderOption	
Name	string	The name of the salesSeriesOrderOption	
Note	string	The comment on the salesSeriesOrderOption	

SalesSeriesOrderPart

A salesSeriesOrderPart is a component of an arrangement.

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the salesSeriesOrderPart	
Name	string	The name of the salesSeriesOrderPart	
Note	string	The comment on the salesSeriesOrderPart	
Quantity	decimal	The quantity of the salesSeriesOrderitem	

5 Articles

Articles – Brief overview

The Webservice provides the possibility to search and process (sell) articles. Following methods can be used to provice regular article selling on your web site.

FindArticleCategories

Calling up this Webservice operation will visualise the desired ArticleCategories on the basis of the linked parameters.

Request

The FindArticlesCategories consists of the service context and an ArticleCategorySearchCriteria object. The latter will determine which ArticleCategories will be found.

ArticleCategorySearchCriteria

Name	Type	Description	Remarks
ArticleCategoryId	GUID	If filled in: Only call up the category with this ID.	
NamePattern	String	If filled in: Only articles that contain a name pattern will be visualised.	Default = empty (no name restriction).
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response is a list of ArticleCategories.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the Article.	
ShortName	String	The code or short name of the category.	
Description	String	The description of the category.	
Level	Int	The level within a hierarchy.	
ParentCategory	GUID	If existing, the higher category in the hierarchy.	If empty, this is the highest level.

FindArticles

Calling up this Webservice operation will visualise the desired Articles on the basis of the linked parameters.

Request

The FindArticlesRequest consists of the service context and an ArticleSearchCriteria object. The latter will determine which Articles will be found.

ArticleSearchCriteria

Name	Type	Description	Remarks
ArticleID	GUID	If filled in: Only call up the article with this ID.	
ArticleGroupId	GUID	If filled in: Only articles of the ArticleGroup with the specified ID are visualised.	Default = empty (articles of all groups).

NamePattern	String	If filled in: Only articles that contain the name pattern will be visualised.	Default = empty (no name restriction).
BuyerId	GUID	If filled in: Only articles that are relevant to a certain buyer are shown.	Not yet in use.Allows to filter the articles on e.g. age (alcoholic drinks).
ArticleTypes	ArticleTypes	Enumeration of ArticleTypes. The following values are possible: • Sale • Rental • All • None	If you, for example, only want to call up the e-purse articles, you put this on None and indicate the ChipKnipArticles under the Includes as true.
IncludeDetail	Boolean	If true, the options, ingredients and warehouses will be called up as well. (See 0: Response)	
Includes	ArticleIncludes	This object will determine which characteristics of the articles will be called up.(See 5.1.1.2: ArticleIncludes)	
Paging	PagingCriteria	See 4.2: Paging Criteria	
StockLocationId	GUID	If not filled in, the total stock is calculated per article for all warehouses at once. If filled in, only the stock of the specified warehouse is calculated.	
ArticleCategoryId	GUID	Only articles of a certain category will be called up.	Default = empty (articles of all categories).
PlaceId	GUID	If filled in, only articles that belong to the POSSalesList of the Place will be shown.	
ForVouchers	Bool?	If true, only the articles that are marked to show in vouchers will be called up.	
TraderId	Guid?	If filled in, only load articles which are linked to the reseller (company)	
Barcode	string	If filled in, only load articles linked with this barcode	
IgnoreActivePeriodsFilter	bool	If true, ignore the active periods filter	

ArticleIncludes

This object will determine which characteristics of the Articles will be called up. This is advisable since certain Web pages only require certain characteristics to be shown while in other cases comprehensive information is needed.

Name	Type	Description	Remarks
Price	Boolean	Find price yes or no.	
ImageUrl	Boolean	Find Image URL of the Article.	
Image	Boolean	Find Image of the Article.	
Group	Boolean	Find ArticleGroup of the Article.	
FreeArticles	Boolean	If False, the Articles of which the price is 0 will not be called up.	
Ingredients	Boolean	Is no longer used. Has been replaced by the ArticleSearchCriteria.IncludeDetail.	
Locations	Boolean	Is no longer used. Has been replaced by the ArticleSearchCriteria.IncludeDetail.	
Options	Boolean	Is no longer used. Has been replaced by the ArticleSearchCriteria.IncludeDetail.	
SoldOutArticles	Boolean	If False, the Articles of which the current stock is 0 will not be called up.	
RentArticles	Boolean	If False, rental articles will not be called up.	
SaleArticles	Boolean	If False, sales articles will not be called up.	
Stock	Boolean	Is no longer used. Has been replaced by the ArticleSearchCriteria.IncludeDetail.	
ArticleCategories	Boolean	If False, the categories will not be called up.	

ChipKnipArticles	Boolean	If False, e-purse articles will not be called up.	
Vat	Boolean	If true, the linked VAT of the article will be added in the response.	
Barcodes	Boolean	If true, include the barcodes linked with the article in the response.	
ActivePeriods	Boolean	If true, include the active periods linked with the article in the response.	
IsLessonGroupItem	Boolean	if true, returns only LessonGroupitems	
IsSubscriptionItem	Boolean	If true, returns only SubscriptionItems	

Response

The response is a list of Articles.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the Article.	
Code	String	The code of the Article.	
Name	String	The name of the Article.	
Description	String	The description of the Article.	
Price	Decimal	The sales price of the Article today, without taking into account price groups.	
Stock	Decimal	The current number in stock.If a warehouse has been specified in the request, the response will only contain the number in stock in the current warehouse.If no warehouse has been specified, the response will contain the total number of articles in stock in all warehouses.	
Image	Picture	The picture of the Article.	
ImageUrl	String	The link on which a picture of the ArticleGroup can be found.	
IsRental	Bool	Indicates whether the article is a rental article or not.	
Group	ArticleGroup	The ArticleGroup to which the article belongs.	
IsStockItem	Bool	Indicates whether the article is a stock article or not.	
DivisionId	GUID	The financial division to which the article belongs. Sales of this article will be booked on this division.	
Ingredients	IList<ArticleIngredient>	The list of ingredients out of which this article consists.	
Options	IList<ArticleOption>	The exclusive options that can be chosen when selling this article.	
StockLocations	IList<StockLocation>	The warehouses in which this article is available.	
Categories	IList<Category>	The list of categories to which this article belongs. The categories can share a hierarchy mutually.	
Attachment	Attachment	The article attachment	
AttachmentUrl	String	The article attachment URL	
ExtraImages	IList<Picture>	List of extra images	
ExtraImageUrls	IList<string>	List of extra image URLs	

ShortDescription	String	The short description of the Article	
MaximumSalesQuantity	int?	The max allowed sales for the article	▪ 0 is infinite ▪ NULL is infinite
IsLessonGroupItem	Bool	Indicates whether the article is a lessongroup article or not.	
UitpasEventId	String	Coupled with this uitpas event id.	
Vat	Vat	The linked vat	
Barcodes	IList<string>	The barcodes linked with the article	
CalculationType	ArticleCalculationType	Enumeration of calculation method. The possible values are: • PriceTimesHours = 0 • FixedPrice = 1 • PriceTimesQuantity = 2 • PriceTimesQuantityTimesHours = 3	The price is either fixed, either dependent on the quantity, the number of hours or the number of hours and quantity. • Price x Hour • Fixed • Price x Number • Price x Number x Hour
ActivePeriods	IList<ActivePeriod>	The list of active periods linked with this article	
HasExternalBarcodes	Bool	Indicates whether the article has external barcodes or not	
TicketDescription	String	The ticket description of the article	
AllowPriceChangeWeb	Bool	Indicates whether the article can have a custom price over the api	
AllowPriceChangeWebshop	Bool	Indicates whether the article can have a custom price true the webshop	
Info1	String	Informational field	
Info2	String	Informational field	
Info3	String	Informational field	
Info4	String	Informational field	
Info5	String	Informational field	

ActivePeriod

Name	Type	Description
Id	Guid	The identifier
From	DateTime?	Optional from date
Until	DateTime?	Optional until date

FindArticleSalesOrders

This Webservice operation allows to generate an overview or history of the sold articles. This is mainly used for the visualisation of 'My articles bought'.

Request

The request contains the service context and the ArticleSalesOrderSearchCriteria object.

ArticleSalesOrderSearchCriteria

This object indicates which data have to be called up.

Name	Type	Description	Remarks
Id	GUID?	Unique ID of the sale.	
PersonId	GUID?	Unique ID of the client.	
From	DateTime?	If filled in: Only the sales that have been registered after the given date will be visualised.	
Until	DateTime?	If filled in: Only the sales that have been registered before the given date will be visualised.	
Type	ArticleSalesOrderType	Enumeration that determines which type of articles is visualised: • All (= all types) • Sales (= sales articles) • Warranty • WaitingList • Service • ChipKnip • LessonGroup • Purchase • PriceGroup • Credit • Rental • Subscription • PurchaseCredit • Family • GiftCertificate • ConsumptionCoupon • FollowUp • SpendingCredit • ETicket	See 5.5.1.1: ArticleSalesOrderSearchCriteria
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response is a list of ArticleSalesOrder objects. An ArticleSalesOrder consists of the following fields.

ArticleSalesOrder

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the sales line.	
Description	String	Description of the sales line.	Is usually copied from the article.
Date	DateTime	Registration date of the sales line.	
PersonId	GUID	Unique ID of the person to whom the sale was made.	
Number	Long	Sales line number.	
SequenceNumber	Int	Sequence number of the sales line.	For sales lines within the same sale.
Quantity	Decimal	Quantity of articles sold.	
UnitPrice	Decimal	Unit price of the article.	
TotalPrice	Decimal	Total price of the sales line.	

FindPriceGroups

Request

The request consists of the service context and a PriceGroupSearchCriteria object.

Name	Type	Description	Remarks
Id	GUID	ID of the specific price group.	Nullable for all price groups.
Type	PriceGroupType	Type of the price group. A price group always belongs to a certain type. One of these types is 'general'. This type means that the price group is active everywhere. This is important for this PriceGroupType selection because, in case of selection of e.g. price groups of the type 'Ticketing', a list of price groups of the type 'General' and of the type 'Ticketing' will be visualised.	Nullable for all types.

Response

The response will contain a list of price groups that meet the criteria.

FindSalesZoneArticles

Request

Name	Type	Description
Context	ServiceContext	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
Criteria	FindSalesZoneArticlesCriteria	The criteria used for FindSalesZoneArticles

FindSalesZoneArticlesCriteria

Name	Type	Description
SalesZoneId	ServiceContext	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
ArticleId	Guid?	The ID of the article you want to retrieve
ArticleGroupId	Guid?	The ID of the article group you wish to retrieve the articles for
ArticleCategoryId	Guid?	The ID of the article category you wish to retrieve the articles for
Includes	SalesZoneArticleIncludes	
Paging	PagingCriteria	
CustomerID	Guid?	The ID of the customer that would buy the article

SalesZoneArticleIncludes

Name	Type	Description
ImageUrl	bool	Retrieves the Image url for the articles
Image	bool	Retrieves the Image for the articles
SoldOutArticles	bool	Indicates whether you wish to retrieve sold articles

Response

List of Articles - See FindArticles

GetArticlePriceInformation

Calling up this Webservice operation will visualise the detailed price information of an article/subscriptionarticle.

Request

The GetArticlePriceInformation consists of the service context and an ArticlePriceInformationSearchCriteria object.

ArticlePriceInformationSearchCriteria

Name	Type	Description	Remarks
ArticleId	GUID	Gets the detailed price of this article.	

CustomerId	GUID	Gets the detailed price of this article for this customer.	

Response

The response is an object of type ArticlePriceInformation.

Name	Type	Description	Remarks
TotalPrice	GUID	The total price of the article price.	
PriceGroup	String	The price group of the Article price.	
FamilyComposition	String	The family composition of the Article price.	
SubsidizationPrice	String	The subsidization price of the Article price.	
AdditionSupplementPrice	Decimal	The addition supplement price of the article price.	
DonationPrice	Decimal	The donation price of the article price.	

ListArticleGroups

This Webservice operation allows the user to visualise the article groups.

Request

The request contains the service context and the ArticleGroupSearchCriteria parameter. The service context has already been described above (see 3.1: The Service Context).

ArticleGroupSearchCriteria

This parameters indicates which data have to be called up.

Name	Type	Description	Remarks
ArticleTypes	ArticleTypes	This is an enumeration that consists of the following values. • Sale = 0 • Rental = 1 • All = 2	Default = Sale
IncludeImage	Bool	Determines if the image also has to be called up or not.	Default = false
IncludeImageUrl	Bool	Determines if the image URL also has to be called up or not.	Default = true

Response

The response is a list of ArticleGroups. An ArticleGroup consists of the following fields.

ArticleGroup

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the ArticleGroup.	
Code	String	The code of the ArticleGroup.	
Name	String	The name of the ArticleGroup.	
Description	String	The description of the ArticleGroup.	
Image	Picture	The picture of the ArticleGroup.	
ImageUrl	String	The link on which a picture of the ArticleGroup can be found.	
Blocked	Bool	Indicates if Article Group is Blocked	
QRMaxDifferentItems	int	Indicates how many different items are allowed on QR code webshop	

6 Basket (shopping basket)

Basket – Brief overview

This section contains general operations which are necessary when working with a shopping basket. A shopping basket is needed to perform registrations within ReCreateX, for example fitness reservations, ticketing seats, activity registrations, … .
A shopping basket contains a list of items (as defined in 6.2.1.1). The web site is responsible for the basket, which means that the web site has to compose and keep up with its baskets.
The basket can then be used by the web service to perform validations and registrations.

BasketItems

1.1.1.1 BasketItem

A basket item is an item in the shopping basket. Basket item is, however, the generic term that is used for the different types, such as ArticleSale, ExpositionPeriodReservation, ActivityReservation, CultureEventReservation, WaitingListSale, ...

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	The number.	
DivisionId	GUID	The division to which the item belongs.	
Name	String	The description of the article.	Read only Is only supported in case of • ActivityReservation • ArticleSale
RuleNamesToIgnore	List<String>	The list of validation rules that do not have to be tested when validating the shopping basket.	
UnitPrice	Decimal	The price for the basket item	
AdvancementPrice	Decimal	The advance price for the basket item	
PromotionRuleDiscountAmount	Decimal	The amount of discount given by promotion rules	

1.1.1.1.1 ArticleSale

An ArticleSale is a specific basket item that is used for the sale of Articles.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	The number.	
DivisionId	GUID	The division to which the item belongs.	
Article	Article	The Article that is sold.	Only the article ID has to be filled in.
UseCards	List<CustomerCardUsage>	The PersonCards which should be used when selling subscriptions	
GenLoyaltyCards	List<GenLoyaltyCard>	The loyalty cards which were used	
CustomerID	Guid?	Unique identifier of the sales customer	
CustomPrice	Decimal	The custom unit price which should be used	Only applicable when configured in RCX.
ExtraDescription	String	Extra description which will be visible in the sales lines	
SubscriptionCustomerID	Guid?	The customer which will be used when creating a subscription	If empty, the Basket customer is used
SubscriptionStartDate	DateTime?	The start date which will be used when creating a subscription	Only applicable when Article is configured correctly
SubscriptionEndDate	DateTime?	The end date which will be used when creating a subscription	Only applicable when Article is configured correctly
LessonGroupID	Guid?	Lessongroup which will be used when creating a subscription	If empty, this field is ignored and no lessongroup is coupled with your subscription.

SubscriptionContactID	Guid?	The contact which will be used when creating a subscription	When provided, fields extraInfo1, extraInfo2 & extraInfo3 of the subscription will be filled in with contact details
UitpasPrice	decimal?	The price that should be paid when using UiTPAS	
UitpasNumber	String	The uitpas number that must be used when processing the payment	
UitpasPriceGroup	String	The uitpas price group that must be used hwen processing the payment	
ProRataFactor	decimal?	The pro rata factor when it is a direct debit purchase	
NumberAddedWithApplyingPromoRules	int	Number of free added items	
LockTicket	LockTicket	The lock ticket	Only when subscribing for a lesson group.
LessonOptions	List<LessonOption>	The lessonoptions which are sold together with the lessongroup article.	see 1.1.1.1.19
DigitalWalletCharge	decimal?		
ExternalCardNumber	string	Card number used for external service (Sorteerstraatjes)	
CustomCardNumber	string	Card Number that should be used for created subscription	
ItemOptions	List<ArticleSaleOption>	Extra options linked to the ArticleSale	
OptionsPerQuantity	Bool	Will multiply the options with quantity.	

1.1.1.1.2 ExpositionPeriodReservation

An ExpositionPeriodReservation is a specific BasketItem that is only used for the sale of reservations for a specific exposition period.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	The number.	Can not be used in this specific case.
UnitPrice	Decimal	The unit price.	Can not be used in this specific case.
Division Id	GUID	The division to which the item belongs.	
ExpositionPeriodId	GUID	The ID of the Exposition Period for which one wants to make a reservation.	
Entries	List <ExpositionPeriodReservationEntry>	If one wishes to make a booking for an ExpositionPeriod, one can register on the basis of different prices. This object will allow this. (see further)	
ArticleSales	List<ArticleSale>	Extra articles that are sold in addition to the exposition.	See 6.2.1.2.1: ArticleSale
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.

OrderWithoutPayment	Bool	True = only order WITHOUT payment
		False = WITH payment
			Deprecated, please use PayLater on Basket instead

LanguageId	Guid?	The selected language for the reservation	The language can be used from RCX 5.4.0
ExpositionId	GUID	The ID of the Exposition for which one wants to make a reservation.	(Optional, will be filled at lock) IMPORTANT: when using enviso expositions/offers, this must be filled in by the client, as the ExpositionPeriodID is not an RCX ID, but an enviso ID.
Donation	Bool	True = Donation is included in the reservation	
GiftAid	Bool	True = GiftAid is included in the reservation	

AdvancementPrice	Decimal	The price of the advance payment	
AutomaticArticleSales	IList<ArticleSale>	Articles that are sold automatic with an exposition	
ArticleOptionalSales	IList<ArticleSale>	Articles that are optionally sold with the exposition	

1.1.1.1.3 ExpositionPeriodReservationEntry

This object contains a number and a linked price group.

Name	Type	Description	Remarks
ParticipantCount	Int	The number of participants for a certain price group.	
PriceGroupId	GUID	The ID of the specific price group.	
Participants	List<ExpositionPeriodReservationEntryParticipant>	The details of the participants	
Cards	List<ExpositionPeriodReservationEntryCard>	The details of the Cards	
PromotionRuleDiscount	Decimal	The amount of discount applied by promotion rules	
Amount	Decimal	How much the ExpositionPeriodREservationEntry costs	
CustomerCardUsage	CustomerCardUsage	Used for mapping the ticket to a specific Person/Card	

1.1.1.1.4 ExpositionPeriodReservationEntryParticipant

This object contains the details of a participant/visitor.

Name	Type	Description	Remarks
Name	String	The name of the participant	
FirstName	String	The first name of the participant	
BirthDate	DateTime	The birth date of the participant	
Email	String	The email of the participant	
Phone	String	The phone of the participant	
Street	String	The street of the participant	
HouseNumber	String	The house number of the participant	
Zipcode	String	The zipcode of the participant	
City	String	The city of the participant	
Country	String	The country of the participant	
UitpasNumber	String	The uitpas number of the participant	
UitpasPrice	String	The uitpas price of the participant's ticket	
CountryGroupId	String	The country group's id	

1.1.1.1.5 CultureEventReservation

A CultureEventReservation is a specific BasketItem that is only used for the sale of tickets for a cultural event.

A CultureEventReservation always has to be locked before a check-out or validation can be done.

This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
CultureEventId	GUID	The unique identifier of the cultural event.	

ReservationDate	DateTime	The date of the reservation.	
Comments	String	Remarks that can be added to the reservation.	
Entries	CultureEventReservationEntry	A list of CultureEventReservationEntries.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
Options	List<CultureEventOptionReservation>	The list of options that have been added to this reservation.	Is not yet used.
ShipTickets	bool	Determines if tickets should be sent by a postal service (can cost extra).	
MustBeANew Reservation	bool	Defines whether the basket item should result in a new ticketing reservation base record.	Default all CultureEventReservation objects are being bundled in one ticketing reservation in RCX.
TraderId	Guid?	When adding a culture event reservation as a reseller, add the ID of the reseller here	
CultureSubscriptionId	Guid	The unique identifier of the linked culture subscription	
CultureSeriesId	Guid	The unique identifier of the linked culture series	
CustomerPriceGroupId	Guid?	The unique identifier of the customer price group	
CustomerId	Guid?	The unique identifier of the linked customer	
GiveDonation	bool	Determines if donation will be giver or not	
Type	CultureSubscriptionEventType	Determines the type of the reservation when used within a ticketing membership reservation	See CultureSubscriptionEntry object

1.1.1.1.6 CultureEventReservationEntry

A CultureEventReservationEntry is the abstract class that will contain the data that specify how many and which tickets will exactly be bought for that specific cultural event.

There are two possible ways to book seats:

- **Best-Available Seats**
  The seats are allocated according to a certain algorithm. You specify how many seats you want and in which price class and the algorithm will try to choose the best seats for you. (Preferably next/close to each other.)
- **Explicit Seats**

You choose the seat you want yourself (exact place).

BestAvailableSeatsCultureEventReservationEntry

Name	Type	Description	Remarks
ParticipantCount	Int	The number of participants.	
PriceGroupId	GUID	The unique identifier of the price group for this entry.	

ExplicitSeatsCultureEventReservationEntry

Name	Type	Description	Remarks
SeatId	GUID	The ID of the seat you want to lock/book.	
PriceGroupId	GUID	The unique identifier of the price group for this entry.	

Example of entries: A father buys 3 tickets for a show. 1 of these tickets is for himself, 2 are for his children. Consequently, he has 2 CultureEventReservationEntries, namely 1 with price group adults and ParticipantCount 1, and a second entry with price group children and ParticipantCount 2.

1.1.1.1.7 CultureEventWaitingListReservation (as of RCX 5.0.5)

A CultureEventWaitingListReservation is a specific BasketItem that is only used for the registration on the waiting list of a cultural event. This type of BasketItem is always free and can not be locked. This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
CultureEventId	GUID	The unique identifier of the cultural event.	
ReservationDate	DateTime	The date of the registration.	
Comments	String	Remarks that can be added to the registration.	
Quantity	Int	Number of desired seats.	
UnitPrice	Decimal	Unit price for the reservation.	
SitePlaceQuantity	Int	Number of desired site places.	

1.1.1.1.8 CultureSubscriptionEntry (as of RCX 5.1.0)

A CultureSubcriptionEntry is a specific BasketItem that is used for the purchase of a culture subscription. When purchasing a culture subscription, you actually buy a bunch of CultureEventReservations in one time, but you ar e limited to the events that are in the allowed list. Some events can be fixed and some can be chosen. This type of BasketItem has to be locked before a check-out or validation can be done. If you lock the subscription, you also lock all the CultureEventReservations. Do not lock the CultureEventReservations one by one! This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
CultureSubscriptionID	GUID	The unique identifier of the culture subscription.	
EventReservations	List<CultureEventReservation>	A list of culture event reservations that are part of the subscription	
PriceGroupID	String	The pricegroup that has to be applied to all reservations (only 1 pricegroup for one subscription!)	
UnitPrice	Decimal	Unit price for the subscription.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items

1.1.1.1.9 WaitingListSale (as of RCX 5.0.5)

A WaitingListSale is a specific BasketItem that is only used for the registration of a person on the waiting list.

A WaitingListSale always has to be locked before a check-out or validation can be done.

This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	Number of students to be registered per object.	Always has to be 1.
UnitPrice	Decimal	The unit price of the linked price article.	
WaitingListId	GUID	ID of the waiting list for which a registration is made.	
ItemId	GUID	ID of the price article (from WaitingList).	
WaitingListLessonGroups	List <LessonGroup>	List of the chosen preferential lesson groups with filled in priority.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
PersonId	GUID	If filled in, this person is registered on the waiting list, if empty, the logged in client is registered at check-out.	As of Service v2.0.1 with Storage v5.0.5.2
CustomerId	GUID?	The person which creates the waiting list reservation. If empty, the default lock person is used.	

1.1.1.1.10 FitnessActivityDayReservation (as of RCX 5.0.5)

A FitnessActivityDayReservation is a specific BasketItem that is only used for the registration for a specific fitness activity.

A FitnessActivityDayReservation always has to be locked before a check-out or validation can be done.

This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	Number of registrations	Always has to be 1.
UnitPrice	Decimal	The unit price of the linked fitness activity.	
FitnessActivityDayId	GUID	ID of the fitness activity day for which a registration is made.	
FitnessActivitySlots	IList<GUID>	List of the chosen slots.	Not obligatory.
PersonId	GUID?	ID of the person who registers.	For non-members, see 10.4: FindFitnessActivities
SubscriptionId	GUID?	ID of the subscription by means of which the person registers.	Not used for non-members.
WaitingList	Bool	Register for the waiting list or not?	True if a day is full and if one wants to register for the waiting list.
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
FitnessActivityId	Guid	ID of the fitness activity for which a registration is made.	(Optional, will be filled at lock)

1.1.1.1.11 RentalReservation

A RentalReservation is a specific BasketItem that is only used for the rental of rental articles.

A RentalReservation always has to be locked before a check-out or validation can be done.

This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	Number of articles.	
UnitPrice	Decimal	The unit price of the linked rental article.	
ArticleId	GUID	ID of the rental article.	
From	DateTime	Rental start date and time.	
Until	DateTime	Rental end date and time.	
Delivery	DateTime?	Date and time of collection of the rental articles.	Not obligatory.
Return	DateTime?	Date and time of return of the rental articles.	Not obligatory.
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.

OrderWithoutPayment	Bool	True = only order WITHOUT payment
		False = WITH payment	

RentalPrice	Decimal	The rental price of the total reservation.	
Comments	String	Extra info of a rental article reservation.	
Customer	Person	The person who rents the rental articles.	
RentalUnitPrice	Decimal	The unit price for the rental reservation.	Filled at recalculate (not used when creating a rental reservation)
RentalQuantity	Int	The quantity for the rental reservation	
RentalHours	Decimal	Number of hours for the rental reservation.	
RentalDays	Decimal	Number of days for the rental reservation.	
RentalNights	Decimal	Number of nights for the rental reservation.	

SingleReservation	Bool	Make a separate rental order for the rental reservation?	Default = False (multiple rental reservations are bundled within one order)
GroupReservationByDate	Bool	Make a separate rental order for each different period.	Default = False (multiple rental reservations are bundled within one order)
ToDeliver	Bool	Mark the reservation as “To deliver” or not.	Only applicable when DeliverChoice is enabled on the RentalArticle
RentalHeaderId	Guid	The id of the rental header that is linked to the rental reservation	Filled at lock/validate
RentalLineIds	IList<Guid>	The ids of the rental lines that are linked to the rental reservation	Filled at lock/validate
ActivityId	Guid?	The id of the rental activity.	
Location	String	The location of the rental reservation.	
ContactId	Guid?	The id of the linked contact person.	
RelationId	Guid?	The id of the linked relation.	
DesiredQuantity	int	Desired number of articles	
GeneralComment	string	General comment about the reservation	

1.1.1.1.12 ActivityReservation (as of RCX 5.0.5)

An ActivityReservation is a specific BasketItem that is only used for the registration of a person for an activity.

An ActivityReservation always has to be locked before a check-out or validation can be done.

This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Activity	Activity	Activity object for which one wants to register.	
Customer	Person	Invoice customer of the registration.	NULL if anonymous registration.
ActivityParticipantReservations	List<ActivityParticipantreservation>	List of ActivityParticipantReservation. (See 6.2.1.2.12.1: ActivityParticipantReservation)	
Comments	String	Remarks concerning the registration.	
TotalPrice	Decimal	The temporary total price of the registration, can still be modified after log-in in case of anonymous registrations.	
OptionPrice	Decimal	The temporary option price of the registration, can still be modified after log-in in case of anonymous registrations.	
UnitPrice	Decimal	Unit price of the activity.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
Quantity	Integer	Number of articles	Not applicable
Status	Integer	0 = reserved temporarily 0 = waiting list 2 = reserved (default)	Only if status = 2, an amount will be linked and the result will be a posted reservation, if not, it will be a non-posted reservation.

1.1.1.1.12.1 ActivityParticipantReservation

An ActivityParticipantReservation is in fact the reservation of a participant within a registration. It is possible to register several participants per registration (e.g. a father registers his 3 children for 1 activity)

Name	Type	Description	Remarks
ParticipantId	GUID	ID fo the participant.	PersonId
Comment	String	Remarks concerning the participant of the registration.	
PriceGroupId	GUID	ID of the price group of this participant	• GUID. Empty: in case of registration on activity price

			• NULL: in case of price calculation on the basis of the customer's price group • Filled in GUID: in case of a price group chosen by the customer See point 18.4: FindPriceGroupActivityByActivity
ParticipantActivityDays	List<ActivityPartReservation>	List of registered days with each time the chosen options. (See 6.2.1.2.12.1.1: ActivityPartReservation	
UitpasPrice	Decimal?	The price that should be paid when using UiTPAS	
UitpasNumber	String	The uitpas number that must be used when processing the payment	
UitpasPriceGroup	String	The uitpas price group that must be used when processing the payment	

1.1.1.1.12.1.1 ActivityPartReservation

An ActivityPartReservation is the combination of an activity day and a list of the chosen option.

Name	Type	Description	Remarks
ActivityPart	ActivityPart	See 0: ActivityPart	
ActivityOptions	List<ActivityOption>	List of options. See 18.3.2.3: ActivityOption	
IsSelected	Bool	Present on that specific day?	

1.1.1.1.13 SubscriptionProlongation (as of RCX 5.1.0)

A SubscriptionProlongation is a specific BasketItem that is only used to renew a subscription. A SubscriptionProlongation does not have to be locked. A Validation, however, is required before a check-out can be processed.

This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	Number.	Has to be 1.
UnitPrice	Decimal	The unit price of the linked subscription.	
SubscriptionId	GUID	ID of the subscription that has to be renewed.	
UitpasEventId	string	The uitpas event id	
UitpasPrice	decimal?	The uitpas tariff	
UitpasNumber	string	The uitpas number	
UitpasPriceGroup	string	The uitpas price group	

1.1.1.1.14 FileActivityPeriodReservation (as of RCX 5.2.0)

A FileActivityPeriodReservation is a specific BasketItem that is only used for making files.

A FileActivityPeriodReservation always has to be locked before a check-out or validation can be done.

This object contains the following elements.

Name	Type	Description	Remarks
Id	Guid	Unique identifier of the BasketItem.	
UnitPrice	Decimal	The unitprice of the BasketItem.	
Quantity	Integer	The number.	Can not be used in this specific case.
FileActivityId	Guid	ID of the file activity of the BasketItem.	
FileTypeId	Guid	ID of the file type of the BasketItem.	
FileAgeGroupId	Guid	ID of the file age group of the BasketItem.	
Entries		List of FileActivityPeriodReservationEntry objects.	

	List<FileActivityPeriodReservationEntry>		
ExecutionDate	DateTime	ExecutionDate of the file activity period reservation.	
ExecutionTime	TimeSpan	ExecutionTime of the file activity period reservation.	
ExecutionEndDate	DateTime	ExecutionEndDate of the file activity period reservation.	
ExecutionEndTime	TimeSpan	ExecutionEndTime of the file activity period reservation.	
NumberOfVisitors	Int	Number of visitors that would participate.	
Comment	String	Remarks concerning the registration.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
Description	String	The description of the file.	
LocationAddressID	Guid	The id of the location of the subfile.	

1.1.1.1.15 FileActivityPeriodReservationEntry

Name	Type	Description	Remarks
ArticleId	Guid	The ID of the specific article.	
PriceGroupId	Guid?	The ID of the specific price group.	
Quantity	Int	The number of articles for a certain pricegroup.	

1.1.1.1.16 ChildCareCentreEntryReservation

A ChildCareCentreEntryReservation is a specific BasketItem that is used for making a child care entry.

A ChildCareCentreEntryReservation always has to be locked before a check-out or validation can be done.

This object contains the following elements.

Name	Type	Description	Remarks
Id	Guid	Unique identifier of the BasketItem.	
UnitPrice	Decimal	The unitprice	Can not be used in this specific case. Always 0.
Quantity	Integer	The number.	Can not be used in this specific case. Always 1.
ChildCareEntryPeriodId	Guid	ID of the child care entry period.	
ChildCareCentreId	Guid	ID of the child care centre.	
PersonId	Guid	ID of the person for the reservation.	
ChildCareCentrePeriodIds	List<Guid>	List of unique child care centre period identifiers.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
LockPersonId	Guid?	ID of the person which is creating the reservation.	A parent can enter a reservation for his child.
ChildCareCentreWaitingListPeriodIds	List<Guid>	List of unique child care centre period identifiers to register on the waiting list.	

1.1.1.1.17 SubscriptionConversion (as of RCX 6.1.0)

A SubscriptionConversion is a specific BasketItem that is only used to convert a subscription. A SubscriptionConversion does not have to be locked. A Validation, however, is required before a check-out can be processed.

This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	Number.	Has to be 1.
UnitPrice	Decimal	The unit price of the linked subscription.	
SubscriptionId	GUID	ID of the subscription that has to be converted.	
SubscriptionConversionArticleId	GUID	ID of the subscription article that the subscription will convert into.	

1.1.1.1.18 FamilySubscriptionAdjustment (as of RCX 6.1.0)

A FamilySubscriptionAdjustment is a specific BasketItem that is only used to adjust a family subscription. A FamilySubscriptionAdjustment does not have to be locked. A validation, however, is required before a check-out can be processed.

This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	Number.	Has to be 1.
UnitPrice	Decimal	The price of the adjustment.	
SubscriptionId	GUID	ID of the subscription that has to be adjusted.	
CustomerCardUsages	CustomerCardUsage	An usage has to be added foreach person that must be adjusted.	

1.1.1.1.19 LessonOption

A lessonOption contains following elements.

Name	Type	Description	Remarks
OptionID	GUIDQUID	Unique identifier of the LessonGroupOption.Unique	
Quantity	Integer	Number of options to reserve.	

1.1.1.1.20 CombiExpositionReservation

An CombiExpositionReservation is a specific BasketItem that is only used for the sale of reservations for combiExposition existing of multiple expositions and a possible extra article

Name	Type	Description	Remarks
ExpositionReservations	List<ExpositionPeriodReservation>	List of ExpositionPeriodReservation objects	
CombiExpositionId	Guid	ID of the combiExposition	
LockTicket	LockTicket		
CustomerId	GUID?	Nullable Guid	
ExtraArticleSales	List<ArticleSale>	List of ArticleSale objects	
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	The number.	Can not be used in this specific case.
UnitPrice	Decimal	The unit price.	Can not be used in this specific case.

1.1.1.1.21 CombiOrganisedVisitRebooking

A CombiOrganisedVisitRebooking is a specific BasketItem that is only used for the sale of a rebooking of a combiOrganisedVisit.

Name	Type	Description	Remarks
OrganisedVisitId	Guid	ID of the OrganisedVisit	

OrganisedVisitPeriodTransfer	CombiOrganisedVistPeriodTransfer	An object that contains a list of PeriodId's	
LockTicket	LockTicket		
Id	Guid	Id of the basketitem	
Quantity	Integer	The number	Can not be used
UnitPrice	Decimal	The unit price	Can not be used

CheckoutBasket

This operation will generate the sale on the basis of the shopping basket. Before the sale will be generated, the same validation as the one of ValidateBasket will take place. Moreover, a CustomerId (the ID of the Person) now also has to be added to the basket.
The check-out will take into account the lock tickets if these are present in the BasketItem.

Request

The request consists of the service context and the Basket that will be checked out.

Response

The response is a CheckOutBasketResult, which consists of a BasketValidationResult and a CheckOutResultState.

Name	Type	Description	Remarks
BasketValidationResult	BasketValidationResult	See 6.2.2.2: BasketItemValidationResult	
ResultState	CheckOutResult	An enumeration that visualises the result of the check-out process: • None • Success • Failed	
SalesItems	List<SalesItems>	A list of the items bought.	
SalesOrderNumber	string	A unique reference to the sales which is created during checkout	The order number can be used for easily finding sales in RCX back office (using POS or reporting)
SalesSeriesId	Guid?	The unique identifier of the sales series which is created during checkout	A sales series is a group of sales which are split over multiple divisions
InvoiceId	Guid?	The unique identifier of the invoice which is created during checkout when paying by invoice	
HasCollectLaterLines	bool	Indicates if the SalesItems object has CollectLaterLines	

SalesItem

A sales item is an item that has been sold in the Web shop. These sales items allow you to generate an overview of the sold articles. Moreover, these sales items will also contain the barcodes or references to barcodes.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the sold article.	
Quantity	Int	The quantity	
Name	String	The name of the sold article.	
UnitPrice	Decimal	The unit price of the sold article.	
BarCodes	List<String>	A list of barcodes linked to the sold article.	
Code	string	Short description of the sold article	
UnitPriceWithoutVat	decimal	The unit price without vat	
SalesDetails	List<SalesItem>	Details of the sale	
SalesHeaderId	Guid	The Id of the SalesHeader	
Date	DateTime	Creation date of the sale	
PriceGroup	PriceGroup	The price group linked with the sales item.	Only contains a value with the call GetCreditSalesByDivisionCard

Amount	decimal	The amount of the sale	
SalesNumber	long?	The number of the SalesHeader	
Arrangement	bool	Indicates if the article is an arrangement article.	

Per article type there can be a specific sales item that only has additional properties.

ExpositionSalesItem

Name	Type	Description	Remarks
ExpositionId	GUID	The unique identifier of the exposition.	
ExpositionPeriodId	GUID	The unique identifier of the exposition period.	
OrganisedVisitId	GUID	The unique identifier of the visit.	
OrganisedVisitNo	long	Number of the organised visit.	

ArticleSalesItem

Name	Type	Description	Remarks
ArticleId	GUID	The unique identifier of the sold article.	
DirectDebitProcessing	bool	Defines if the sold article has direct debit processing.	
IsMembership	bool	Is the sale a membership?	
UitpasPaymentError	UitpasPaymentError	If a tariff of UiTPAS is used but the payment did not succeed, then this field contains a value with more information.	

UitpasPaymentError

Name	Type	Description	Remarks
UitpasNumber	string	The uitpasnumber that was used	
UitpasEventId	string	The eventId that was used	
PriceGroup	string	The pricegroup that was used	
Type	UitpasPaymentErrorType	The type of error	PaymentFailed = 0, PaymentPriceIsWrong = 1
Price	decimal	The price that was used	

CultureEventReservationSalesItem

Name	Type	Description	Remarks
Seats	List<SeatAllocation>	A list of reserved seats.	
CultureEventReservationId	GUID	The ID of the created reservation.	This ID is used by the DocumentService in order to generate tickets.
Sites	List<SiteAllocation>	A list of allocated sites.	

ActivityReservationSalesItem

Name	Type	Description	Remarks
EntryId	Guid	The ID of the Entry	
OrderId	Guid	The ID of the order	
UitpasPaymentErrors	IList<UitpasPaymentError>	If a tariff of UiTPAS is used but the payment did not succeed for a participant, then this collection will contain a record with more information.	

WaitingListSalesItem

Name	Type	Description	Remarks
ArticleId	Guid	The ID of the Article	

FitnessActivityDayReservationSalesItem

Name	Type	Description	Remarks
FitnessActivityDayReservationId	Guid	The ID of the fitness activity day reservation	

RentalReservationSalesItem

Name	Type	Description	Remarks
RentalReservationId	Guid	The ID of the rental reservation	

PlaceReservationSalesItem

Name	Type	Description	Remarks
ReservationId	Guid	The ID of the reservation	

RecurringPlaceReservationSalesItem

Name	Type	Description	Remarks
SerieId	Guid	The ID of the serie	

SubscriptionProlongationSalesItem

Name	Type	Description	Remarks
SubscriptionId	GUID	The unique identifier of the renewed subscription.	

FileSalesItem

Name	Type	Description	Remarks
FileActivityId	GUID	The unique identifier of the file activity.	
FileId	GUID	The unique identifier of the file.	

ChildCareCentreEntryReservationSalesItem

Name	Type	Description	Remarks
ChildCareCentreEntryId	GUID	The unique identifier of the child care centre entry.	

OrganisedVisitRebookingSalesItem

Name	Type	Description	Remarks
SubscriptionId	GUID	The unique identifier of the subscription.	

CreatePaymentMethod

This API call creates a new payment method.

Request

The Request consists of the service context, with one extra parameter "Code".

Name	Type	Description	Remarks
Code	String	The unique code of the payment method	Max 20 characters
Provider	String	The provider that provides the payment method	Optional

			Max 50 characters

Response

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the newly created payment method	

ExtendLockPeriod

This operation allows the end user to prolong a lock ticket in time. The time by which the lock is prolonged is determined in the configuration of the Webservice.

Request

The request consists of the service context and a list of LockTickets. The lock tickets have to be added to the correct type.

ExtendLockPeriodByTime

This operation allows the end user to prolong a lock ticket by a certain time. It is also possible to prolong the lock ticket infinite. This means the lock ticket will never expire. To do this, provide a negative time span in the request (ex: -1).

Request

The request consists of the service context, a list of LockTickets and a time span. The lock tickets have to be added to the correct type.

ListLockedBasketItems

This operation loads all locked basket items. This can be usefull when users log in and want to load the basket items that they created in a previous session.

Request

The request consists of the service context and a **ListLockedBasketItemsCriteria** object.

Name	Type	Description	Remarks
PersonId	Guid	The person for which the locked basket items should be loaded	
Includes	ListLockedBasketItemsIncludes	Determines which data (which basket items) are included in the response.	

The **ListLockedBasketItemsIncludes** object can be built up using the following criteria:

Name	Type	Description	Remarks
ActivityReservations	Bool	Include the activity reservations or not.	
ExpositionPeriodReservations	Bool	Include the exposition reservations or not.	
CultureEventReservations	Bool	Include the culture event reservations or not.	
CultureSubscriptionEntries	Bool	Include the culture event subscriptions or not.	
WaitingListSales	Bool	Include the waiting list reservations or not.	
FitnessActivityDayReservations	Bool	Include the fitness reservations or not.	
RecurringPlaceReservations	Bool	Include the recurring place reservations or not.	
OptionalRecurringPlaceReservations	Bool	Include the optional recurring place reservations or not.	
PlaceReservations	Bool	Include the place reservations or not.	
OptionalPlaceReservations	Bool	Include the optional place reservations or not.	
RentalReservations	Bool	Include the rental reservations or not.	
ChildCareCentreEntryReservations	Bool	Include the child care reservations or not.	
CultureSeriesReservations	Bool	Include the culture series reservations or not.	
StockArticles	Bool	Include the stock articles or not	

CombiExpositionReservations	Bool	Include the combi exposition reservations or not	

Response

The response will contain a **ListLockedBasketItemsResult** object which looks as follows:

Name	Type	Description	Remarks
SessionId	Guid	The new session id which is required to be used for the new basket.	
LockBasketResult	LockBasketResult	The result set containing the basket items.	See 6.6.2

ListPaymentMethods

This operation generates a list of the payment methods that are known in ReCreateX (e.g. Visa, payment card, ...).

Request

The Request consists of the service context and a ListPaymentMethodsCritiria object. The object looks as follows:

ListPaymentMethodsCriteria

Name	Type	Description	Remarks
PointOfSaleId	Guid?	The unique identifier of the point of sale	

Response

The response consists of a list of BasketPaymentMethods.
Such a **BasketPaymentMethod** looks as follows:

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the BasketPaymentMethod.	
Code	String	The unique code of the BasketPaymentMethod.	
Name	String	The description of the BasketPayment Method.	
DefaultCashPaymentMethod	Bool	Indicates if the BasketPayment Method is set as the default payment method for cash payments	
DefaultIncassoPaymentMethod	Bool	Indicates if the BasketPayment Method is set as the default payment method for incasso payments	
CreditCardType	int	The type of credit card	
PrintVendingTicket	bool?	Indicates if the vending ticket should be printed when using this payment method	
ToGoAccount	bool?	Indicates if the to go account is activated for this payment method	
WalletPaymentMethod	bool	Indicates if the BasketPayment Method is set as the default payment method for wallet (= chip knip) payments	
InvoicePaymentMethod	bool	Indicates if the BasketPayment Method is set as the default payment method for invoice payments	
NotRefundedPaymentMethod	bool	Indicates if the BasketPayment Method is set as the default payment method for not refunded payments	
CountMoney	bool	Indicates if the amount should be counted when using this payment method	
FixedPaymentAmount	bool	Indicates if the payment amount is fixed or not.	
DigitalWalletPaymentMethod	bool	Indicates if the payment is a Digital Wallet Payment or not	
AmountMayExceedTheRemainingBalance	bool	Indicates if the amount may exceed the remaining balance	
WorldLinePaymentProductId	string	The WorldLine payment product id that the PaymentMethod is linked to	

If a PaymentMethod has direct debit processing, an **IncassoBasketPaymentMethod** looks as follows:

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the IncassoBasketPaymentMethod.	

Code	String	The unique code of the IncassoBasketPaymentMethod.	
Name	String	The description of the IncassoBasketPayment Method.	
DefaultCashPaymentMethod	Bool	Indicates if the BasketPayment Method is set as the default payment method for cash payments	
DefaultIncassoPaymentMethod	Bool	Indicates if the BasketPayment Method is set as the default payment method for incasso payments	
PaymentMethod	String	The payment method representation for the corresponding ReCreateX online payment method	
Brand	String	The brand representation for the corresponding ReCreateX online payment method	
CreditCardType	int	The type of credit card	
PrintVendingTicket	bool?	Indicates if the vending ticket should be printed when using this payment method	
ToGoAccount	bool?	Indicates if the to go account is activated for this payment method	
WalletPaymentMethod	bool	Indicates if the BasketPayment Method is set as the default payment method for wallet (= chip knip) payments	
InvoicePaymentMethod	bool	Indicates if the BasketPayment Method is set as the default payment method for invoice payments	
NotRefundedPayment Method	bool	Indicates if the BasketPayment Method is set as the default payment method for not refunded payments	
AdditionalIncassoCosts	List<IncassoCost>	List of costs incurred by direct debit payment.	Extra costs for direct debit in 1x, 2x or 3x.

Direct debit processing is only possible in case of reservation of a cultural event and can not be combined with other payments or other types of BasketItems.

An **IncassoCost** looks as follows:

Name	Type	Description	Remarks
Name	String	The name of the IncassoCost.	
Cost	Decimal	The value of the IncassoCost.	
NumberOfPayments	Int32	The number of times in which the payment is made.	Is 1, 2 or 3.

ListTransportArticles

This operation generates a list of the transport articles that are known in ReCreateX. In ReCreateX RentalParameterSetup you can choose the possible transport articles for the webshop. These articles should be proposed to the user if the user has added a RentalReservation to its basket. When the user choses a transport article, that article will be added to its rentalreservation at Checkout of the basket.

Request

The request consists of the service context and a **TransportArticleSearchCriteria** object.

Name	Type	Description	Remarks
Rental	Bool	If the basket contains items of the module Rental than the transport options from the module Rental should be shown.	
Includes	TransportArticleIncludes	Determines which data are included in the response and/or which extra records will be present in the returnset.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

The **TransportArticleIncludes** object can be built up using the following criteria:

Name	Type	Description	Remarks
ImageUrl	Bool	Find the image or not.	
Image	Bool	Find the image or not.	
ArticleGroup	Bool	Find the data of the article group or not.	
Attachment	Bool	Find the attachments or not.	

Response

The response will contain a list of TransportArticle objects that meet the criteria.
The **TransportArticle** object looks as follows:

Name	Type	Description	Remarks
Module	Int	The number of the module for which the transport article can be chosen.	
ArticleId	Guid	ID of the article of the transport article.	
Article	Article	Article of the transport article.	

A TransportCost looks as follows:

Name	Type	Description	Remarks
Name	String	The name of the TransportCost.	
Cost	Decimal	The value of the TransportCost.	
Module	Int32	The number of the module of the transport article.	
ItemID	Guid	The ID of the transport article article.	

LockBasketItems

The Service Method Lock Basket items will lock the items that can be locked. In order to lock those items, a validation will precede.

Request

The request consists of the service context and of a list of basket items that have to be locked.

Response

The response consists of a LockBasketResult.

Name	Type	Description	Remarks
IsLocked	Boolean	Indicates whether the item is locked or not.	
List<BasketItem>	List<BasketItem>	A list of basket items of which the lock ticket will be filled in on the basket item.	
ValidationResult	ValidationResult		

The lock ticket type that will be added to the basket item depends on the basket item itself.
A basic lock ticket looks as follows.

LockTicket

Name	Type	Description	Remarks
ExpirationTime	DateTime	Date until which the lock is valid.	
Id	Guid	The ID of the reservation.	

CultureEventReservationLockTicket

This lock ticket has all the characteristics of a basic lock ticket.
Additional fields:

- list of SeatAllocations which indicates which seats have been temporarily booked by this lock operation.
- List of site Allocations.
- Boolean that indicates if the seats are kept together.

SeatAllocation

A SeatAllocation is an allocated seat for a cultural event and looks as follows:

Name	Type	Description	Remarks
CardNumber	String	The card number that is linked to the reservation.	
CultureEventId	Guid	The identifier of the event for which a reservation has been made.	

CultureEventReservationId	Guid	The unique identifier for which the seat has been reserved.	
CustomerId	Guid	The identifier of the customer that has made the reservation.	
Price	Decimal	The price of the reserved seat.	
SeatBlockCode	String	The zone in which the seat is located.	
SeatCode	String	The unique code of the seat within the row.	
SeatId	Guid	The unique code of the seat in the venue.	SeatId
Status	SeatAllocationStatus (enum)	The status of the seat allocation	
Id	Guid	The unique identifier	
ShowOnWeb	bool	Determine if the seat can be shown on web	
CustomerPriceGroupId	Guid	The identifier of the chosen customer price group	
PromotionRuleDiscountAmount	decimal	The discount given by promotion rules	

ExpositionReservationLockTicket

This lock ticket has all the characteristics of a basic lock ticket.
Additional fields:

Name	Type	Description	Remarks
ExternalBarcodeIds	List<Guid>		
EnvisoReservationIds	List<Guid>	Enviso reservation ids	Only when a reservation has been made
EnvisoOrderId	Guid	Enviso Order ID	Only when an order has been created

LockPosBasketItems

The Service Method Lock Basket items will lock the items that can be locked. In order to lock those items, a validation will precede.

> This is an alternative method to LockBasketItems, and should only be used if you make use of the WaiterPadService.
>
> Using this method results in different behaviour, for example: this method also creates POSSalesLines.

Request

The request consists of the service context, waiterpadcontext and of a list of basket items that have to be locked.

The WaiterPadContext contains information like employee, point of sale, table (optional) and macId.

Response

The response consists of a LockBasketResult.

Name	Type	Description	Remarks
IsLocked	Boolean	Indicates whether the item is locked or not.	
List<BasketItem>	List<BasketItem>	A list of basket items of which the lock ticket will be filled in on the basket item.	
ValidationResult	ValidationResult		

RecalculateBasket

This function will calculate the correct price of the basket on the basis of a given basket and return it in a response holding the correct prices.

Request

The request of the RecalculateBasket consists of a service context and a basket.

Response

The Response is the recalculated basket.

ResetLockPeriod

This operation allows the end user to reset an infinite lock ticket. This operation will only succeed when the ticket has been extended infinite.

Request

The request consists of the service context and a list of LockTickets. The lock tickets have to be added to the correct type.

UnLockBasketItems

This method will delete the lock of the basket items that have a lock.
For more information on locking and unlocking, see 3.3.3: Lock and Unlock basket items.

> ⊗ This method can only be used for objects which were locked by the **LockBasketItems** method and are not yet expired.
> If the objects are not locked or are expired, then a proper message will return that the unlock was not successful.

Request

The request consists of the service context and a list of LockTicket objects.
Only the ID of each LockTicket has to be filled in.

Response

The response contains a **UnlockBasketItemsResult** object.

UnlockBasketItemsResult

Name	Type	Description	Remarks
IsValid	bool	Determines the result of the unlock process	• TRUE if all unlock operations for all lock tickets were successful. • FALSE if at least one unlock operation was not successful.
ValidationResult	IList<UnlockBasketItemResult>	List of underlying validation results	

UnlockBasketItemResult

Name	Type	Description	Remarks
IsValid	bool	Determines the result of the unlock process	• TRUE if the unlock operation for the corresponding lock ticket was successful. • FALSE if the unlock operation for the corresponding lock ticket was not successful.
LockTicket	LockTicket	The lock ticket which corresponds to the unlock operation	
ValidationResult	ValidationResult	The validation result which contains more detailed information about the lock process	

ValidateBasket

If the shopping basket has been filled on the client side, it first has to be validated to check whether it can be processed or not. For this purpose, the ValidateBasket operation is used.

Request

The request consists of the service context and the basket object itself.

Basket

A basket is the collection of basket items, payment methods and a buyer (if this one is already known).

Name	Type	Description	Remarks
Items	List<BasketItems>	A list of items that are in the shopping basket.	See also further.
Price	Decimal	The total price of the shopping basket.	Has to be the same as the total amount of the payments.
Payments	List<Payments>	A list of payment methods and linked amount.	See also further. At least 1 payment has to be present for validation.
CustomerId	GUID	Unique identifier that will be used to identify the buyer. • The Person ID of the logged in user.	Field obligatory for validation.
DeliveryAddressId	GUID	Unique identifier that will be used for the registration of a delivery address.	Not obligatory.
AdditionalCosts	List<AdditionalCost>	A list of additional costs that are in the shopping basket.	
OrderId	String	Unique order number for online payments.	Is taken over in the RCX payment.
PayLater	Boolean	Allows paying later (not all basketitems support this)	
ExcludeCosts	ExcludeCosts	Specifies whether certain costs are excluded	
AdditionalDiscounts	List<AdditionalDiscount>	A list of additional discounts that are in the shopping basket.	
TrxId	String	Unique transaction number for online payments	Is taken over in the RCX payment (from RCX 5.4.0)
PayId	String	Unique payment number for online payments	Is taken over in the RCX payment (from RCX 5.4.0)
ContactId	GUID	Unique identifier that will be used to identify the contact person of the buyer (user which is logged in).	
Comment	String	Comment that will be used within the sales headers.	
CouponCodes	List<string>	List of coupon codes that apply to the basket.	
GiftCertificates	List<string>	List of gift certificates that apply to the basket.	

ExcludeCosts

The ExcludeCosts object defines whether certain costs are excluded.

Name	Type	Description	Remarks
ExcludeAdministrativeCost	Bool	Exclude administrative costs	
ExcludeReservationCost	Bool	Exclude reservation costs	
ExcludeExpositionReservationCost	Bool	Exclude exposition reservation costs	
ExcludeShippingCost	Bool	Exclude shipping costs	

BasketItem

A basket item is an item in the shopping basket. Basket item is, however, the generic term that is used for the different types, such as ArticleSale, ExpositionPeriodReservation, ActivityReservation, CultureEventReservation, WaitingListSale, ...

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	The number.	
DivisionId	GUID	The division to which the item belongs.	
Name	String	The description of the article.	Read only Is only supported in case of • ActivityReservation • ArticleSale
RuleNamesToIgnore	List<String>		

		The list of validation rules that do not have to be tested when validating the shopping basket.	
UnitPrice	Decimal	The price for the basket item	
AdvancementPrice	Decimal	The advance price for the basket item	

ArticleSale

An ArticleSale is a specific basket item that is used for the sale of Articles.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	The number.	
DivisionId	GUID	The division to which the item belongs.	
Article	Article	The Article that is sold.	Only the article ID has to be filled in.
UseCards	List<CustomerCardUsage>	The PersonCards which should be used when selling subscriptions	
GenLoyaltyCards	List<GenLoyaltyCard>	The loyalty cards which were used	
CustomerID	Guid?	Unique identifier of the sales customer	
CustomPrice	Decimal	The custom unit price which should be used	Only applicable when configured in RCX.
ExtraDescription	String	Extra description which will be visible in the sales lines	
SubscriptionCustomerID	Guid?	The customer which will be used when creating a subscription	If empty, the Basket customer is used
SubscriptionStartDate	DateTime?	The start date which will be used when creating a subscription	Only applicable when Article is configured correctly
SubscriptionEndDate	DateTime?	The end date which will be used when creating a subscription	Only applicable when Article is configured correctly
LessonGroupID	Guid?	Lessongroup which will be used when creating a subscription	
SubscriptionContactID	Guid?	The contact which will be used when creating a subscription	
UitpasPrice	decimal?	The price that should be paid when using UiTPAS	
UitpasNumber	String	The uitpas number that must be used when processing the payment	
UitpasPriceGroup	String	The uitpas price group that must be used hwen processing the payment	
ProRataFactor	decimal?	The pro rata factor when it is a direct debit purchase	
NumberAddedWithApplyingPromoRules	int	Number of free added items	
LockTicket	LockTicket	The lock ticket	
LessonOptions	List<LessonOption>	The lessonoptions which are sold together with the lessongroup article.	
DigitalWalletCharge	decimal?		
ExternalCardNumber	string	Card number used for external service (Sorteerstraatjes)	
CustomCardNumber	string	Card Number that should be used for created subscription	
ItemOptions	List<ArticleSaleOption>	Extra options linked to the ArticleSale	

ExpositionPeriodReservation

An ExpositionPeriodReservation is a specific BasketItem that is only used for the sale of reservations for a specific exposition period.

Name	Type	Description	Remarks

Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	The number.	Can not be used in this specific case.
UnitPrice	Decimal	The unit price.	Can not be used in this specific case.
DivisionId	GUID	The division to which the item belongs.	
ExpositionPeriodId	GUID	The ID of the Exposition Period for which one wants to make a reservation.	
Entries	List<ExpositionPeriodReservationEntry>	If one wishes to make a booking for an ExpositionPeriod, one can register on the basis of different prices. This object will allow this. (see further)	
ArticleSales	List<ArticleSale>	Extra articles that are sold in addition to the exposition.	See 6.2.1.2.1: ArticleSale
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.

OrderWithoutPayment	Bool	True = only order WITHOUT payment
		False = WITH payment
			Deprecated, please use PayLater on Basket instead

LanguageId	Guid?	The selected language for the reservation	The language can be used from RCX 5.4.0
ExpositionId	GUID	The ID of the Exposition for which one wants to make a reservation.	(Optional, will be filled at lock)
Donation	Bool	True = Donation is included in the reservation	
GiftAid	Bool	True = GiftAid is included in the reservation	
AdvancementPrice	Decimal	The price of the advance payment	
CustomerId	Guid?	The person which creates the expositionperiod reservation. If empty, the default lock person is used.	
PersonalizedMessage	String	Personalized message	

ExpositionPeriodReservationEntry

This object contains a number and a linked price group.

Name	Type	Description	Remarks
ParticipantCount	Int	The number of participants for a certain price group.	
PriceGroupId	GUID	The ID of the specific price group.	
Participants	List<ExpositionPeriodReservationEntryParticipant>	The details of the participants	

ExpositionPeriodReservationEntryParticipant

This object contains the details of a participant/visitor.

Name	Type	Description	Remarks
Name	String	The name of the participant	
FirstName	String	The first name of the participant	
BirthDate	DateTime	The birth date of the participant	

CultureEventReservation

A CultureEventReservation is a specific BasketItem that is only used for the sale of tickets for a cultural event.
A CultureEventReservation always has to be locked before a check-out or validation can be done.
This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
CultureEventId	GUID	The unique identifier of the cultural event.	

ReservationDate	DateTime	The date of the reservation.	
Comments	String	Remarks that can be added to the reservation.	
Entries	CultureEventReservationEntry	A list of CultureEventReservationEntries.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
Options	List<CultureEventOptionReservation>	The list of options that have been added to this reservation.	Is not yet used.
ShipTickets	bool	Determines if tickets should be sent by a postal service (can cost extra).	

CultureEventReservationEntry

A CultureEventReservationEntry is the abstract class that will contain the data that specify how many and which tickets will exactly be bought for that specific cultural event.
There are two possible ways to book seats:

- **Best-Available Seats**The seats are allocated according to a certain algorithm. You specify how many seats you want and in which price class and the algorithm will try to choose the best seats for you. (Preferably next/close to each other.)
- **Explicit Seats**

You choose the seat you want yourself (exact place).
BestAvailableSeatsCultureEventReservationEntry

Name	Type	Description	Remarks
ParticipantCount	Int	The number of participants.	
PriceGroupId	GUID	The unique identifier of the price group for this entry.	

ExplicitSeatsCultureEventReservationEntry

Name	Type	Description	Remarks
SeatId	GUID	The ID of the seat you want to lock/book.	
PriceGroupId	GUID	The unique identifier of the price group for this entry.	

Example of entries: A father buys 3 tickets for a show. 1 of these tickets is for himself, 2 are for his children. Consequently, he has 2 CultureEventReservationEntries, namely 1 with price group adults and ParticipantCount 1, and a second entry with price group children and ParticipantCount 2.

CultureEventWaitingListReservation (as of RCX 5.0.5)

A CultureEventWaitingListReservation is a specific BasketItem that is only used for the registration on the waiting list of a cultural event. This type of BasketItem is always free and can not be locked. This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
CultureEventId	GUID	The unique identifier of the cultural event.	
ReservationDate	DateTime	The date of the registration.	
Comments	String	Remarks that can be added to the registration.	
Quantity	Int	Number of desired seats.	
UnitPrice	Decimal	Unit price for the reservation.	
SitePlaceQuantity	Int	Number of desired site places.	

CultureSubscriptionEntry (as of RCX 5.1.0)

A CultureSubcriptionEntry is a specific BasketItem that is used for the purchase of a culture subscription. When purchasing a culture subscription, you actually buy a bunch of CultureEventReservations in one time, but you ar e limited to the events that are in the allowed list.

Some events can be fixed and some can be chosen. This type of BasketItem has to be locked before a check-out or validation can be done. If you lock the subscription, you also lock all the CultureEventReservations. Do not lock the CultureEventReservations one by one! This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
CultureSubscriptionID	GUID	The unique identifier of the culture subscription.	
EventReservations	List<CultureEventReservation>	A list of culture event reservations that are part of the subscription	
PriceGroupID	String	The pricegroup that has to be applied to all reservations (only 1 pricegroup for one subscription!)	
UnitPrice	Decimal	Unit price for the subscription.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items

WaitingListSale (as of RCX 5.0.5)

A WaitingListSale is a specific BasketItem that is only used for the registration of a person on the waiting list.
A WaitingListSale always has to be locked before a check-out or validation can be done.
This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	Number of students to be registered per object.	Always has to be 1.
UnitPrice	Decimal	The unit price of the linked price article.	
WaitingListId	GUID	ID of the waiting list for which a registration is made.	
ItemId	GUID	ID of the price article (from WaitingList).	
WaitingListLessonGroups	List <LessonGroup>	List of the chosen preferential lesson groups with filled in priority.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
PersonId	GUID	If filled in, this person is registered on the waiting list, if empty, the logged in client is registered at check-out.	As of Service v2.0.1 with Storage v5.0.5.2
CustomerId	GUID?	The person which creates the waiting list reservation. If empty, the default lock person is used.	

FitnessActivityDayReservation (as of RCX 5.0.5)

A FitnessActivityDayReservation is a specific BasketItem that is only used for the registration for a specific fitness activity.
A FitnessActivityDayReservation always has to be locked before a check-out or validation can be done.
This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	Number of registrations	Always has to be 1.
UnitPrice	Decimal	The unit price of the linked fitness activity.	
FitnessActivityDayId	GUID	ID of the fitness activity day for which a registration is made.	
FitnessActivitySlots	IList<GUID>	List of the chosen slots.	Not obligatory.
PersonId	GUID?	ID of the person who registers.	For non-members, see 10.4: FindFitnessActivities
SubscriptionId	GUID?	ID of the subscription by means of which the person registers.	Not used for non-members.
WaitingList	Bool	Register for the waiting list or not?	True if a day is full and if one wants to register for the waiting list.

LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
FitnessActivityId	Guid	ID of the fitness activity for which a registration is made.	(Optional, will be filled at lock)

RentalReservation

A RentalReservation is a specific BasketItem that is only used for the rental of rental articles.
A RentalReservation always has to be locked before a check-out or validation can be done.
This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	Number of articles.	
UnitPrice	Decimal	The unit price of the linked rental article.	
ArticleId	GUID	ID of the rental article.	
From	DateTime	Rental start date and time.	
Until	DateTime	Rental end date and time.	
Delivery	DateTime?	Date and time of collection of the rental articles.	Not obligatory.
Return	DateTime?	Date and time of return of the rental articles.	Not obligatory.
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.

OrderWithoutPayment	Bool	True = only order WITHOUT payment
		False = WITH payment	

RentalPrice	Decimal	The rental price of the total reservation.	
Comments	String	Extra info of a rental article reservation.	
Customer	Person	The person who rents the rental articles.	
RentalUnitPrice	Decimal	The unit price for the rental reservation.	Filled at recalculate (not used when creating a rental reservation)
RentalQuantity	Int	The quantity for the rental reservation	
RentalHours	Decimal	Number of hours for the rental reservation.	
RentalDays	Decimal	Number of days for the rental reservation.	
RentalNights	Decimal	Number of nights for the rental reservation.	
SingleReservation	Bool	Make a separate rental order for the rental reservation?	Default = False (multiple rental reservations are bundled within one order)
GroupReservationByDate	Bool	Make a separate rental order for each different period.	Default = False (multiple rental reservations are bundled within one order)
ToDeliver	Bool	Mark the reservation as "To deliver" or not.	Only applicable when DeliverChoice is enabled on the RentalArticle

ActivityReservation (as of RCX 5.0.5)

An ActivityReservation is a specific BasketItem that is only used for the registration of a person for an activity.
An ActivityReservation always has to be locked before a check-out or validation can be done.
This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Activity	Activity	Activity object for which one wants to register.	
Customer	Person	Invoice customer of the registration.	NULL if anonymous registration.
ActivityParticipantReservations	List<ActivityParticipantreservation>	List of ActivityParticipantReservation. (See 6.2.1.2.12.1: ActivityParticipantReservation)	
Comments	String	Remarks concerning the registration.	
TotalPrice	Decimal	The temporary total price of the registration, can still be modified after log-in in case of anonymous registrations.	

OptionPrice	Decimal	The temporary option price of the registration, can still be modified after log-in in case of anonymous registrations.	
UnitPrice	Decimal	Unit price of the activity.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
Quantity	Integer	Number of articles	Not applicable
Status	Integer	0 = reserved temporarily0 = waiting list 2 = reserved (default)	Only if status = 2, an amount will be linked and the result will be a posted reservation, if not, it will be a non-posted reservation.

ActivityParticipantReservation

An ActivityParticipantReservation is in fact the reservation of a participant within a registration. It is possible to register several participants per registration (e.g. a father registers his 3 children for 1 activity)

Name	Type	Description	Remarks
ParticipantId	GUID	ID fo the participant.	PersonId
Comment	String	Remarks concerning the participant of the registration.	
PriceGroupId	GUID	ID of the price group of this participant	• GUID. Empty: in case of registration on activity price • NULL: in case of price calculation on the basis of the customer's price group • Filled in GUID: in case of a price group chosen by the customer See point 18.4: FindPriceGroupActivityByActivity
ParticipantActivityDays	List<ActivityPartReservation>	List of registered days with each time the chosen options. (See 6.2.1.2.12.1.1: ActivityPartReservation	

ActivityPartReservation

An ActivityPartReservation is the combination of an activity day and a list of the chosen option.

Name	Type	Description	Remarks
ActivityPart	ActivityPart	The activity part	
ActivityOptions	List<ActivityOption>	List of options.	
IsSelected	Bool	Present on that specific day?	

SubscriptionProlongation (as of RCX 5.1.0)

A SubscriptionProlongation is a specific BasketItem that is only used to renew a subscription. A SubscriptionProlongation does not have to be locked. A Validation, however, is required before a check-out can be processed.
This object contains the following elements.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the BasketItem.	
Quantity	Integer	Number.	Has to be 1.
UnitPrice	Decimal	The unit price of the linked subscription.	
SubscriptionId	GUID	ID of the subscription that has to be renewed.	

FileActivityPeriodReservation (as of RCX 5.2.0)

A FileActivityPeriodReservation is a specific BasketItem that is only used for making files.
A FileActivityPeriodReservation always has to be locked before a check-out or validation can be done.
This object contains the following elements.

Name	Type	Description	Remarks
Id	Guid	Unique identifier of the BasketItem.	

UnitPrice	Decimal	The unitprice of the BasketItem.	
Quantity	Integer	The number.	Can not be used in this specific case.
FileActivityId	Guid	ID of the file activity of the BasketItem.	
FileTypeId	Guid	ID of the file type of the BasketItem.	
FileAgeGroupId	Guid	ID of the file age group of the BasketItem.	
Entries	List<FileActivityPeriodReservationEntry>	List of FileActivityPeriodReservationEntry objects.	
ExecutionDate	DateTime	ExecutionDate of the file activity period reservation.	
ExecutionTime	TimeSpan	ExecutionTime of the file activity period reservation.	
ExecutionEndDate	DateTime	ExecutionEndDate of the file activity period reservation.	
ExecutionEndTime	TimeSpan	ExecutionEndTime of the file activity period reservation.	
NumberOfVisitors	Int	Number of visitors that would participate.	
Comment	String	Remarks concerning the registration.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
Description	String	The description of the file.	
LocationAddressID	Guid	The id of the location of the subfile.	

FileActivityPeriodReservationEntry

Name	Type	Description	Remarks
ArticleId	Guid	The ID of the specific article.	
PriceGroupId	Guid?	The ID of the specific price group.	
Quantity	Int	The number of articles for a certain pricegroup.	

ChildCareCentreEntryReservation

A ChildCareCentreEntryReservation is a specific BasketItem that is used for making a child care entry.
A ChildCareCentreEntryReservation always has to be locked before a check-out or validation can be done.
This object contains the following elements.

Name	Type	Description	Remarks
Id	Guid	Unique identifier of the BasketItem.	
UnitPrice	Decimal	The unitprice	Can not be used in this specific case. Always 0.
Quantity	Integer	The number.	Can not be used in this specific case. Always 1.
ChildCareEntryPeriodId	Guid	ID of the child care entry period.	
ChildCareCentreId	Guid	ID of the child care centre.	
PersonId	Guid	ID of the person for the reservation.	
ChildCareCentrePeriodIds	List<Guid>	List of unique child care centre period identifiers.	
LockTicket	LockTicket		See 3.3.3: Lock and Unlock basket items.
LockPersonId	Guid?	ID of the person which is creating the reservation.	A parent can enter a reservation for his child.

Payments

The user has the possibility to link one or several payment methods to the basket. For this purpose BasketPayments are used. A BasketPayment looks as follows.
Important: If direct debit payment has been chosen, an IncassoBasketPayment has to be used! If not, no direct debit contract will be created which will lead to invalid payments.

Name	Type	Description	Remarks
Amount	Decimal	The amount of the payment.	

Currency	String	The name of the currency.	Is not yet used.
PaymentMethodId	GUID	The ID of the payment method as known in ReCreateX.	
ExtraInfo1	String	The first of three info fields the user can provide to register extra info for the sales process.	
ExtraInfo2	String	The second of three info fields the user can provide to register extra info for the sales process.	
ExtraInfo3	String	The third of three info fields the user can provide to register extra info for the sales process.	
PaymentTerminalText	String	The feedback from a payment terminal that was provided when the user has successfully payed on a pin terminal. This feedback is registered in the database during the checkout and can be included in a proof of payment.	

IncassoBasketPayment
An IncassoBasketPayment consists of a BasketPayment but has some additional properties.

Name	Type	Description	Remarks
NumberOfIncassoPayments	Int	The number of direct debit payments chosen by the user.	

PersonCardPayment
A PersonCardPayment is used as dummy payment when the user actually wants to put the amount due on his to go account. The settlement of the account happens on a later time in ReCreateX backoffice.
To be able to put the payment on a to-go account, the account has to be identifief by a person's divisioncard.
This is why the PersonCardId is needed in the PersonCardPayment.
The PersonCardID can be obtained by a call to FindPersonCards or GetCard. Take the id from the card property of the PersonCard. If one or more cards are found, the PersonCardId is the Id property of the returned PersonCard in the Response.

Name	Type	Description	Remarks
PersonCardId	Id	The DivisionCardID for the person	This is NOT a PersonCardID, but a CardID!

Response

Validating a basket will lead to a ValidateBasketResponse which holds a BasketValidationResult. A BasketValidationResult consists of a BrokenRuleName, a message and a BasketItemValidationResult.
The BrokenRuleName will each time return the name of the ValidationRule that could not be met.
If the BrokenRuleName is empty, one can conclude that each ValidationRule has been met and that the validation succeeded. Subsequently, one no longer has to take into account the BasketItemValidationResults and/or the message either.

BasketValidationResult

Validating a basket consists of the validation of the basket itself and of the BasketItems.

Name	Type	Description	Remarks
BrokenRuleName	String	Contains the name of the rule that could not be met. Empty if validation was successful.	
Message	String	Message linked to the BrokenRuleName.	
BasketItemValidationResults	List<BasketItemValidationResults>	List of validations of BasketItems that could not be validated successfully. (see further)	

BasketItemValidationResult

Just as BasketValidationResult this object also contains a BrokenRuleName and a linked message. The ValidationBasketItem is added here, which contains the validated basket item.

Name	Type	Description	Remarks
BrokenRuleName	String	Contains the name of the rule that could not be met. Empty if validation was successful.	
Message	String	Message linked to the BrokenRuleName.	
ValidatedBasketItem	BasketItem	The BasketItem that could not be successfully validated.	

ValidateBasketItem

This function can be compared to ValidateBasket. The only difference is that only 1 item is given here instead of several items in case of ValidateBasket.

7 Discount codes (coupons / vouchers / gift certificates)

CouponCalculate

This operation will calculate the discount you receive for the current shopping basket using one or multiple discount codes.

Request

The request consists of the service context and a CouponCalculateCriteria object. The CouponCalculateCriteria object contains the basket in which you need to fill the CouponCodes property.

Name	Type	Description	Remarks
Basket	Basket	The basket for which you want to calculate the discount	

Response

The response consists of an CouponCalculateResult that indicates whether the discount code(s) are calculated correctly.

Name	Type	Description
Status	CouponStatus	The status for the coupon code(s). One of the following enumerations: • Success = 0 • CouponNotFound = 1 • CouponCanNotBeUsedByAddressGroup = 2 • CouponCanNotBeUsedInDivision = 3 • CouponIsNotValidYet = 4 • CouponExpired = 5 • CouponUsedInInvalidTimeframe = 6 • CouponUsedOnWrongDayOfWeek = 7 • CouponCanNotBeUsedInThisSalesChannel = 8 • CouponMaximumNumberOfUsagesReached = 9 • CouponNotApplicableToSales = 10 • CouponSalesSeriesAlreadyUsed = 11 • CouponsCanNotBeUsedTogether = 12 • CouponInsufficientCredits = 13 • CouponIsNoVoucher = 14
Discounts	List<CouponDiscount>	List of the discounts that are ready to be reserved

The CouponDiscount object consists of following elements:

Name	Type	Description	Remarks
DiscountAmount	Decimal	The discount amount you will receive	
CouponID	GUID	The unique coupon identifier	
CouponCode	String	The coupon code	
CouponDescription	String	The coupon description	

CouponRelease

This operation will release the discount you reserved in a previous transaction using the CouponReserve method.

Request

The request consists only of the service context. The release is executed using the unique Session ID of the ServiceContext. The session ID needs to be unique per shopping basket!

Response

The response returns a CouponReleaseResult which contains the following properties

Name	Type	Description
IsSuccess	Bool	Indiciates if the release had succeeded
Message	string	When the release failed, this field contains a value

CouponReserve

This operation will reserve the discount you receive for the current shopping basket using one or multiple discount codes. Once reserved, the discount can only be applied by the Checkout method or released by the CouponRelease method.

Request

The request consists of the service context and a CouponReserveCriteria object. The CouponReserveCriteria object contains the basket in which you need to fill the CouponCodes property.

Name	Type	Description	Remarks
Basket	Basket	The basket for which you want to calculate the discount	

Response

The response consists of an CouponReserveResult that indicates whether the discount code(s) are calculated reserved.

Name	Type	Description
Status	CouponStatus	The status for the coupon code(s). One of the following enumerations: • Success = 0 • CouponNotFound = 1 • CouponCanNotBeUsedByAddressGroup = 2 • CouponCanNotBeUsedInDivision = 3 • CouponIsNotValidYet = 4 • CouponExpired = 5 • CouponUsedInInvalidTimeframe = 6 • CouponUsedOnWrongDayOfWeek = 7 • CouponCanNotBeUsedInThisSalesChannel = 8 • CouponMaximumNumberOfUsagesReached = 9 • CouponNotApplicableToSales = 10 • CouponSalesSeriesAlreadyUsed = 11 • CouponsCanNotBeUsedTogether = 12 • CouponInsufficientCredits = 13 • CouponIsNoVoucher = 14
Discounts	List<CouponDiscount>	List of the discounts that are ready to be reserved
CouponReservations	List<GUID>	List of unique identifiers of all reserved coupons

Discount codes – Brief overview

Coupons

As from ReCreateX 5.5.0, it is possible to define discount codes. These codes can be used within the online process sales.
Its usage consists of following operations:

Calculate	Calculates the discount for one or multiple discount codes
Reserve	Reserves the discount for one or multiple discount codes
Release	Releases the discount for one or multiple discount codes (undo reserve)

Vouchers

As from ReCreateX 5.8.0, it is possible to define vouchers. Vouchers can be used to buy specific articles within the web shop.

VoucherValidate	Validates the voucher codes and returns the items that can be bought using these voucher codes

Gift certificates

As from ReCreateX 5.12.1, it is possible to use gift certificates within the API. These codes can be used within the online process sales. Gift certificates have a fixed amount and can be used more than once until the amount is completely spent.

The only operation you need (to calculate the discount) is the following:

Calculate	Calculates the discount for one or multiple gift certificates

When doing a CheckoutBasket, the gift certificates which are provided will be automatically handled correctly.

FindGiftCertificates

With this you can retrieve all gift certificates for a certain customer.

Request

The request consists of the service context and a **Criteria** object.

In this Criteria, the following properties exist

Name	Type	Description	Remarks
CustomerId	Guid	The unique identifier of the customer you wish to retrieve the gift certificates for.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response consists of a **FindGiftCertificatesResult** that contains all the gift certificates.

Name	Type	Description
GiftCertificates	List<GiftCertificate>	List of the gift certificates

GiftCertificate

The **GiftCertificate** object consists of following elements:

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the gift certificate	
Customer	Person	The customer that bought the gift certificate	
PersonCard	PersonCard	The info of the person and the card	
Article	Article	The article linked to the gift certificate	
Amount	Decimal	The amount of value of the gift certificate	
PurchaseDate	Datetime	The purchase date of the gift certificate	
ValidFrom	DateTime	The date of when the gift certificate becomes valid	
ValidTill	DateTime	The date of when the gift certificate expires	
Coupon	Bool	Indicates whether the gift certificate is a coupon	
SalesSeriesID	Guid?	The identifier of the sales series	
ShortName	String	The short name of the gift certificate	
Description	String	The description of the gift certificate	
TicketDescription	String	The ticket description of the gift certificate	
ExtraDescription	String	The extra description of the gift certificate	
Info1	String	The info1 value of the gift certificate	
TemplateId	Guid	The template id of the gift certificate	
Valid	Bool	Indicates whether the gift certificate is currently valid	

GiftCertificateCalculate

This operation will calculate the discount you receive for the current shopping basket using one or multiple gift certificates.

Request

The request consists of the service context and a **GiftCertificateCalculateCriteria** object. The GiftCertificateCalculateCriteria object contains the basket in which you need to fill the GiftCertificates property.

Name	Type	Description	Remarks
Basket	Basket	The basket for which you want to calculate the discount	

Response

The response consists of an **GiftCertificateCalculateResult** that indicates whether the gift certificate(s) are calculated correctly.

Name	Type	Description
Status	GiftCertificateStatus	The status for the gift certificate(s). One of the following enumerations: • Success = 0 • GiftCertificateNotFound = 1 • GiftCertificateHasNoBalance = 2
Discounts	List<GiftCertificateDiscount>	List of the gift certificates that are ready to be reserved

The **GiftCertificateDiscount** object consists of following elements:

Name	Type	Description	Remarks
Number	String	The gift certificate number	
DivisionCardID	Guid	The unique identifier of the linked division card	
RemainingAmount	Decimal	The remaining amount for the gift certificate	

SetCouponPrinted

This operation will set the coupon as printed.

Request

The request consists of the service context and the SetCouponPrintedCriteria.

Name	Type	Description	Remarks
Coupons	List<Coupons>	List of coupon ID's which will be marked as printed	

Response

No response available.

SetGiftCertificatePrinted

With this API call you can set the GiftCertificate as Printed.

When executed, it will fill in the current date and time as the PrintDate.

Request

The request consists of the service context and a **Criteria** object.

In this Criteria, the following properties exist

Name	Type	Description
Criteria	SetGiftCertificatePrintedCriteria	The unique identifier of the customer you wish to retrieve the gift certificates for.

SetGiftCertificatePrintedCriteria

Name	Type	Description
Id	Guid	The unique identifier of the Gift Certificate for which you want to set the PrintDate

VoucherValidate

This operation will validate one or more voucher codes and return the items that can be bought using these codes.

Request

The request consists of the service context and a VoucherValidateCriteria object. The VoucherValidateCriteria object contains following properties.

Name	Type	Description
PersonId	Guid?	The person who uses the voucher codes.
VoucherCodes	List<string>	The list of voucher codes.

Response

The response consists of an VoucherValidateResult object that indicates whether the voucher code(s) are valid and which items can be bought using them.

Name	Type	Description
VoucherStates	List<VoucherStatus>	The voucher validation states.
CouponDetails	List<CouponDetail>	List of items that can be bought using the given voucher codes

The **VoucherStatus** object defines if a specific voucher code is valid.

Name	Type	Description
VoucherCode	String	The voucher code that was validated
Status	CouponStatus	The status for the coupon code(s). One of the following enumerations: • Success = 0 • CouponNotFound = 1 • CouponCanNotBeUsedByAddressGroup = 2 • CouponCanNotBeUsedInDivision = 3 • CouponIsNotValidYet = 4 • CouponExpired = 5 • CouponUsedInInvalidTimeframe = 6 • CouponUsedOnWrongDayOfWeek = 7 • CouponCanNotBeUsedInThisSalesChannel = 8 • CouponMaximumNumberOfUsagesReached = 9 • CouponNotApplicableToSales = 10 • CouponSalesSeriesAlreadyUsed = 11 • CouponsCanNotBeUsedTogether = 12 • CouponInsufficientCredits = 13 CouponIsNoVoucher = 14

The **CouponDetail** object defines the items for which a specific voucher code can be used.
The CouponDetail object has multiple overrides:

- CouponDetailArticle
- CouponDetailSubscription
- CouponDetailExposition

Name	Type	Description
SubjectID	Guid	The ID of the item (article, subscription, exposition).

SalesType	CouponSalesType	The type of the item. • Article = 0, • Subscription = 1, • Exposition = 2, • CultureEvent = 3, • Activity = 4, • Place = 5, • Other = 6,
PriceTypeID	Guid	The price type ID of the item (ex: price of an exposition)
Description	String	Item description

CouponDetailArticle

Name	Type	Description
Article	Article	The linked article.

CouponDetailSubscription

Name	Type	Description
SubscriptionArticle	SubscriptionArticle	The linked subscription article.

CouponDetailExposition

Name	Type	Description
Exposition	Exposition	The linked exposition. (When using product types within the voucher, only the specific price will be returned)

CancelFashionCheque

This will cancel a redemption of a fashioncheque based on the given criteria

Request

The request consists of the service context and a GetStatusFashionChequeCriteria object.

Name	Type	Description	Remarks
CardNumber	string	The cardnumber of the Fashioncheque you want to retrieve information for	
VerificationCode	string	The security code linked to the card.	May be null.
Amount	decimal	The amount u want to redeem	
Currency	string	The currency used to redeem	
ActivationRequestId	string	the id used for redemption that can later be reused to cancel a redemption	This is self generated. You are the master. Not Fashioncheque

Response

Name	Type	Description
FashionCheque	FashionCheque	The Fashioncheque with its information

FashionCheque

Name	Type	Description	Remarks
Status	string	The status of the card	

ExpireDate	DateTime	The expiration date of the card	
RefillsRemaining	int	The amount of times the card can be refilled.	
TransactionsRemaining	int	The amount of transactions that still can be preformed	
InitialValue	decimal	The value the card had when activated	
MinValue	decimal	The minimum value the card can have on activation	
MaxValue	decimal	The maximum value the card can have on activation	
HasVerificationCode	bool	Tells us if the card has a verification code	
ProductIdentifier	string	The identifier for the product	
Number	string	The number of the card	
VerificationCode	string	The verification code of the card	
Amount	MoneyInfo	keeps the current amount and the currency	

Amount

Name	Type	Description	Remarks
Value	decimal	The value still left on the card	
Currency	string	the currency of the value left on the card. for example "EUR"	

GetStatusFashionCheque

This will retrieve information about a given FashionCheque cardnumber.

Request

The request consists of the service context and a GetStatusFashionChequeCriteria object.

Name	Type	Description	Remarks
CardNumber	string	The cardnumber of the Fashioncheque you want to retrieve information for	
VerificationCode	string	The security code linked to the card.	May be null.

Response

Name	Type	Description
FashionCheque	FashionCheque	The Incert voucher with its information

FashionCheque

Name	Type	Description	Remarks
Status	string	The status of the card	
ExpireDate	DateTime	The expiration date of the card	
RefillsRemaining	int	The amount of times the card can be refilled.	
TransactionsRemaining	int	The amount of transactions that still can be preformed	
InitialValue	decimal	The value the card had when activated	
MinValue	decimal	The minimum value the card can have on activation	
MaxValue	decimal	The maximum value the card can have on activation	
HasVerificationCode	bool	Tells us if the card has a verification code	
ProductIdentifier	string	The identifier for the product	
Number	string	The number of the card	
VerificationCode	string	The verification code of the card	
Amount	MoneyInfo	keeps the current amount and the currency	

Amount

Name	Type	Description	Remarks
Value	decimal	The value still left on the card	
Currency	string	the currency of the value left on the card. for example "EUR"	

GetStatusIncertVoucher

This will retrieve information about a given Incert voucher code.

Request

The request consists of the service context and a GetStatusIncertVoucherCriteria object.

Name	Type	Description	Remarks
VoucherCode	string	The code of the Incert voucher you want to retrieve information for	

Response

Name	Type	Description
Voucher	IncertVoucher	The Incert voucher with its information

IncertVoucher

Name	Type	Description	Remarks
VoucherCode	string	The code of the voucher	
VoucherName	string	The name of the voucher	
VoucherDescription	string	The description of the voucher	
InitialAmount	decimal	The initial amount of the voucher	
CurrentAmount	decimal	The current amount of the voucher	
Currency	string	The currency of the voucher	
CustomerFullName	string	The customer's full name of the voucher	
ValidTill	DateTime	The datetime of how when the voucher will no longer be valid	
Status	int	The status of the voucher	
PartlyRedeemable	bool	Indicates whether the voucher can be partly redeemed or not	
Validity	Validity	The validity of the voucher	
PosSalesLineId	Guid	The PosSalesLine of the voucher sale	
RedemptionsHistory	• List<IncertRedemptionHistory>	The historic list of redemptions for the voucher	

Validity

- VoucherCodeNotFound = 0,
- Active = 1
- Cancelled = 2
- FullyRedeemed = 3
- VoucherTypeNotAllowed = 4
- VoucherNotPaidYet = 5
- VoucherAlreadyExpired = 6
- VoucherNotValidForThisStation = 7
- VoucherNotActivated = 8
- VoucherNotValidForThisRateCode = 9

IncertRedemptionHistory

Name	Type	Description	Remarks
RedemptionID	string	The identifier of the redemption	
RedemptionDate	datetime	The date of the redemption	
RedemptionValue	decimal	The value of the redemption	
RedemptionComment	string	The comment of the redemption	
RedemptionStationID	int	The identifier of the station of the redemption	
RedemptionStationName	string	The name of the station of the redemption	
BookingID	string	The identifier of the booking of the voucher	
InclusiveName	string		
InclusiveID	string		

RedeemFashionCheque

This will redeem a fashioncheque based on the given criteria

Request

The request consists of the service context and a GetStatusFashionChequeCriteria object.

Name	Type	Description	Remarks
CardNumber	string	The cardnumber of the Fashioncheque you want to retrieve information for	
VerificationCode	string	The security code linked to the card.	May be null.
Amount	decimal	The amount u want to redeem	
Currency	string	The currency used to redeem	
ActivationRequestId	string	the id used for redemption that can later be reused to cancel a redemption	This is self generated. You are the master. Not Fashioncheque

Response

Name	Type	Description
FashionCheque	FashionCheque	The Fashioncheque with its information

FashionCheque

Name	Type	Description	Remarks
Status	string	The status of the card	
ExpireDate	DateTime	The expiration date of the card	
RefillsRemaining	int	The amount of times the card can be refilled.	
TransactionsRemaining	int	The amount of transactions that still can be preformed	
InitialValue	decimal	The value the card had when activated	
MinValue	decimal	The minimum value the card can have on activation	
MaxValue	decimal	The maximum value the card can have on activation	
HasVerificationCode	bool	Tells us if the card has a verification code	
ProductIdentifier	string	The identifier for the product	
Number	string	The number of the card	
VerificationCode	string	The verification code of the card	
Amount	MoneyInfo	keeps the current amount and the currency	

Amount

Name	Type	Description	Remarks
Value	decimal	The value still left on the card	
Currency	string	the currency of the value left on the card. for example "EUR"	

RedeemIncertVoucher

This will redeem a specific amount of an Incert voucher

Request

The request consists of the service context and a RedeemIncertVoucherCriteria object.

Name	Type	Description	Remarks
VoucherCode	string	The code of the Incert voucher you want to retrieve information for	
Amount	decimal	The amount that is wished to be redeemed of the voucher	

Response

Name	Type	Description
RedeemInfo	IncertRedeemInfo	The information returned about the redemption

IncertRedeemInfo

Name	Type	Description	Remarks
RedemptionId	string	The identifier of the redemption	
AmountRedeemed	decimal	The amount that was redeemed for the voucher	
NewAccountBalance	decimal	The newly updated balance of the voucher after redemption	

CancelIncertVoucher

This will attempt to cancel a redemption of an Incert voucher

Request

The request consists of the service context and a CancelIncertVoucherCriteria object.

Name	Type	Description	Remarks
RedemptionID	string	The ID of the redemption that is wished to be cancelled	

Response

Name	Type	Description
CancelInfo	IncertCancelInfo	The information returned about the cancellation of the redemption

IncertCancelInfo

Name	Type	Description	Remarks
Success	bool	Indicates whether the cancel of the redemption was successful	
VoucherCode	string	The code of the voucher that the redemption was cancelled for	
CurrentAmount	decimal	The current amount of the voucher	

8 Profile

AuthenticateManager

This operation will authenticate a manager on the basis of his user name and password. The manager authentication is used for extra configurations of the Website. (CMS management)

Request

The request consists of the service context and a Credential object. The **Credential** object looks as follows.

Name	Type	Description	Remarks
UserName	String	The Login with which the manager will authenticate himself.	
Password	String	The password with which the manager will authenticate himself.	

Response

The response consists of an **AuthenticationResult** that indicates whether the user is authenticated successfully or not.

Name	Type	Description	Remarks
HasSucceeded	Boolean	Indicates if the user has been successfully authenticated or not.	
Error	AuthenticationError	An enumeration that returns the error of the AuthenticationProcess. Possible values are: • None • InvalidParameters • InvalidCredential	For security purposes no correct information is given here.
PersonId	GUID		Is not used.
LastLoginDate	DateTime?		Is not used.

AuthenticateNationalNumber

This operation will authenticate a user on the basis of his national register number. It is the Website's responsibility to limit the use of this method to once. This can be done by obliging the user to choose a user name and a password using this method during his first log-in! If the user already has a user name within ReCreateX, he will not be able to authenticate using this method.

Request

The request consists of the service context and a text object for the national register number.

Name	Type	Description	Remarks
NationalNumber	String	The national register number with which the user will authenticate himself.	

Response

The response consists of an AuthenticationResult that indicates whether the user is authenticated successfully or not.

Name	Type	Description	Remarks
HasSucceeded	Boolean	Indicates if the user has been successfully authenticated or not.	
Error	AuthenticationError	An enumeration that returns the error of the AuthenticationProcess. Possible values are: • None • InvalidParameters • UnknownUser • BlockedUser • HasUserName	For security purposes no correct information is given here.
PersonId	GUID	The ID of the authenticated user.	
LastLoginDate	DateTime?	Indicates when the user authenticated himself the last time.	If empty: first time.

AuthenticateToken

This operation will authenticate a user on the basis of a unique token. A token can be requested by the **RequestAuthenticationToken** method.

Request

The request consists of the service context and a unique GUID representing the token.

Name	Type	Description	Remarks
Token	Guid	The token with which the user will authenticate himself.	

Response

The response consists of an AuthenticationResult that indicates whether the user is authenticated successfully or not.

Name	Type	Description	Remarks
HasSucceeded	Boolean	Indicates if the user has been successfully authenticated or not.	
Error	AuthenticationError	An enumeration that returns the error of the AuthenticationProcess. Possible values are: • None • InvalidToken • InvalidTokenTime • InvalidParameters • UnknownUser • BlockedUser • InvalidCredential	For security purposes no correct information is given here.
PersonId	GUID	The ID of the authenticated user.	
LastLoginDate	DateTime?	Indicates when the user authenticated himself the last time.	If empty: first time.

AuthenticateUser

This operation will authenticate a user on the basis of his user name and password.

Request

The request consists of the service context and a Credential. The Credentials look as follows.

Name	Type	Description	Remarks
UserName	String	The Login with which the user will authenticate himself.	
Password	String	The password with which the user will authenticate himself.	

Response

The response consists of an AuthenticationResult that indicates whether the user is authenticated successfully or not.

Name	Type	Description	Remarks
HasSucceeded	Boolean	Indicates if the user has been successfully authenticated or not.	
Error	AuthenticationError	An enumeration that returns the error of the AuthenticationProcess. Possible values are: • None • InvalidParameters • UnknownUser • BlockedUser • InvalidCredential	For security purposes no correct information is given here.
PersonId	GUID	The ID of the authenticated user.	
LastLoginDate	DateTime?	Indicates when the user authenticated himself the last time.	If empty: first time.

DeleteAuthenticationToken

This operation deletes a specific authentication token. After using this method, the token will no longer be valid.

Request

The request consists of the service context and a unique GUID representing the token.

Name	Type	Description	Remarks

Token	Guid	The token to be deleted.	

Response

The response consists of an DeleteAuthenticationResult that indicates whether the token was deleted or not.

Name	Type	Description	Remarks
Deleted	Boolean	Indicates if the token was deleted. When this is False, this can have multiple reasons: • Token is not present/valid • Token was already deleted • …	

DeleteCeviPersons

This operation is used to delete an user in the ReCreateX system.
This operation may only be used by the CEVI service.

Request

The request consists of the service context and a list of referenceNumbers as a string.
The characteristics used for a CeviPerson object look as follows:

Response

The response will return a DeleteCeviPersonsResult object. This object contains the possibly ValidationResults that indicate what went wrong.

DeleteCeviPersonsResult

Name	Type	Description	Remarks
ValidationResults	List<ValidationResult>		

DeletePersonDeliveryAddress

This method is used for deleting a delivery address.

Request

The request consists of the service context and the unique ID of the object to be deleted.

Name	Type	Description	Remarks
RelationId	GUID	ID of the relation.	

Response

The response contains a **ValidationResult** object. This object looks as follows:

Name	Type	Description	Remarks
BrokenRuleName	String	The error code.	
Message	String	The standard error.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	

DeletePersonRelation

Request

The Request consists of the service context and a DeletePersonRelationCriteria object.

DeletePersonRelationCriteria

Name	Type	Description	Remarks

RelationId	GUID	ID of the relation.	
RemoveSubscriptions	Bool	Remove the family subscriptions when removing the relation	

Response

The response contains a **ValidationResult** object. This object looks as follows:

Name	Type	Description	Remarks
BrokenRuleName	String	Name of the validation rule.	
Message	String	Error message.	
IsValid	Bool	Rule has been successfully processed.	If false, the other properties will contain more details.

FindClubLocationContractTypes

This method is used for calling up the possible contract types for club locations.

Request

The request consists of a service context and a ClubLocationContractTypesCriteria object.
The properties used for a ClubLocationContractTypesCriteria object look as follows:

Name	Type	Description	Remarks
Id	GUID?	Unique ID of the contact.	Nullable, not obligatory.
Keyword	String	Keyword to search for. Leave empty to ignore.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response visualises a list of ClubLocationContractType objects that meet the given criteria.
This ClubLocationContractType looks as follows:

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the contract type.	
Code	String	Code of the contract type.	
Description	String	Description of the contract type.	

PersonFunction

This is a description of the PersonFunction object.

Name	Type	Description	Remarks
ID	Guid	The ID of the function of the person.	
ShortName	String	The shortname of the function of the person.	
Description	String	The description of the function of the person.	

FindInvoices

This method is used for finding invoices and to get a document through the document service

Request

The request consists of a service context and a FindInvoicesCriteria object.
The properties used for a FindInvoicesCriteria object look as follows:

Name	Type	Description	Remarks
Id	GUID?	Unique ID of the contact.	Nullable, not obligatory.
PersonId	GUID?	Filter by person	Advised, but not obligatory

Paging	PagingCriteria	See 4.2: Paging Criteria	
StartDate	DateTime?	Only invoices with an invoicedate starting from this date	Not obligatory, but advised
EndDate	DateTime?	Only invoices with an invoicedate until this date	Not obligatory, but advised
IncludeDetails	Boolean	Include saleslines in the invoice object	Default false
AllDivisions	Boolean	Include invoices for all divisions	Default false, only include for the division linked to the shopid

Response

The response visualises a list of Invoice objects that meet the given criteria.
This Invoice object looks as follows:

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the Invoice	
InvoiceNumber	String	Number of the invoice	
InvoiceDate	DateTime	The invoice date	
ExpirationDate	DateTime	The expiration date	
InvoicePaymentDate	DateTime	Date when the invoice was payed	
Customer	String	Name of the customer	
CustomerCity	String	City of the customer	
Amount	Decimal	Total amount of the invoice	
Payed	Decimal	Already payed amount	
OGMNumber	String	OGM Number	
Status	Int	0 = fully payed, others are either not payed or partially payed	
ToPay	Decimal	Amount that still has to be payed	
PaymentMethods	String	Payment methods used to pay	
CustomerStreet1	String	Street 1 of the customer	
CustomerStreet2	String	Street 2 of the customer	
CustomerBox	String	Box of the customer	
CustomerZipCode	String	Zipcode of the customer	
Date	DateTime		
SalesLines	IList<InvoiceSalesLine>	List of InvoiceSalesLine objects	Only included when IncludeDetails is true
CustomerStreetNumber	String	Street number of the customer	

InvoiceSalesLine

This object is mostly used for visualisation of the Invoice document itself, but could be used to get some data.
The InvoiceSalesLine object looks as follows:

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the Invoice	
InvoiceId	GUID	Id of the invoice	
Type	Int	Type of the salesline	
Sequence	Int	Sequence number	
Item	String	Name of the item that was sold	
UnitOfMeasure	String	Unit of measure of the item	
Description	String	Description of the salesline	
Quantity	Decimal	Quantity	
UnitPrice	Decimal	Price per unit	
DiscountAmount	Decimal	Discount	

VATPercentage	Decimal	Percentage of the VAT	
VATAmount	Decimal	Amount of the VAT	
Amount	Decimal	Total amount without VAT	
Arrangement	Boolean	Is this an arrangement?	
ArrangementPart	Boolean	Is this part of an arrangement?	
SalesNumber	String	Number of this sale	
CultureEventDescription	String	Description of the culture event	Multilingual
CultureEventStartTime	DateTime	Starttime of the culture event	
PriceGroupId	GUID	Id of the pricegroup	
PriceGroup	String	Name of the pricegroup	
LineAmount	Decimal	Amount including VAT	
CultureEventShortName	String		
PriceGroupDescription	String	Description of the pricegroup	Multilingual
CultureEventStartDate	DateTime		
CultureEventId	GUID		
CultureEventReservationNumber	Long		
ItemDescription	String	Description of the item	Multilingual
OrganisedVisitNumber	Long		
ExpositionDescription	String		
OrganisedVisitStartDate	Datetime		
OrganisedVisitEndDate	DateTime		
OrganisedVisitStartTime	DateTime		
OrganisedVisitEndTime	DateTime		
ReservationNumber	Long		
ReservationPlace	String	Place of the reservation	
ReservationInfrastructure	String	Infrastructure of the reservation	
ReservationFrom	DateTime		
ReservationTo	DateTime		
ActivityId	GUID		
ActivityShortName	String		
ActivityDescription	String		Multilingual
CalculationType	String	Description of the calculation type	Multilingual

GetPersonNotifications

This method is used for getting a list of notification for a certain person. Callling this method will also trigger the creation of new notifications.

Request

The request consists of a service context and a PersonNotificationSearchCriteria object.
The properties used for a PersonNotificationSearchCriteria object look as follows:

Name	Type	Description	Remarks
PersonId	GUID?	Filter by person	Advised, but not obligatory
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response visualises a list of PersonNotification objects that meet the given criteria.
This PersonNotification object looks as follows:

Name	Type	Description	Remarks

Id	GUID	The unique identifier of the notification	
PersonId	GUID	Id of the person	
NotificationType	int	Type of the notification	**0 = Undefined** **1 = Birthday** **2 = LLV Graduation**
CreationDate	DateTime	The creation date of this notification	
EventDate	DateTime	Date when the event occurred that triggered the notification	
NotifiedDate	DateTime?	The date when the user was notiefied	If null, the user has not yet been notified
SubjectId	GUID?	Id for the object this notification is about	**Optional**
SubjectInfo	PersonNotificationSubjectInfo	Extra information about the subject	

PersonNotificationSubjectInfo

This object contains extra information about the subject of a notification.
It looks as follows:

Name	Type	Description	Remarks
GraduationInfo	GraduationInfo	Extra info for type 2 notifications	

GraduationInfo

This object contains extra information about type 2 notifications (LLV Graduation).
It looks as follows:

Name	Type	Description	Remarks
DiplomaName	String	The name of the diploma the student graduated for	
LessonGroupId	GUID	The ID of the lessongroup the student was in when he graduated	
PersonId	GUID	The ID of the person that graduated	This can also be the id of the person from a child relation

FindContactRelations

Request

The request consists of a service context and a FindContactRelationsCriteria object.
The properties used for a FindContactRelationsCriteria object look as follows:

FindPersonsCriteria

Name	**Type**	**Description**	**Remarks**
ContactId	GUID?	Id to find persons linked to the contact	
PersonId	GUID?	Id to find persons linked to the contact based on a person	
Includes	PersonIncludes	Determines which parts of the person will be included.	
IncludesImage	Boolean	Determines of blocked persons have to be called up.	
IncludeImageUri	Boolean	National register number of the user you want to search.	

PersonIncludes

This object allows to exclude certain parts of a person.

See FindPerson for more information about the PersonIncludes object

Response

The respone visualises a list of Person objects that meet the FindPersonsCriteria.

See SavePerson for more information about the Person object.

FindETickets

This method is used for finding all sold e-tickets based on the sales.

Request

The request consists of the service context and a **ETicketSearchCriteria** object.

Name	Type	Description	Remarks
PersonId	GUID	Unique ID of the person for which you want to receive all e-ticket information.	Give Empty Guid to ignore the person
From	DateTime?	If filled in: Only e-tickets that were created after the given date will be returned.	
Until	DateTime?	If filled in: Only e-tickets that were created before the given date will be returned.	
SalesLineId	GUID?	If filled in: e-tickets regarding this SalesLineId	
SubscriptionId	GUID?	If filled in: e-tickets regarding this subscription	
OrganisedVisitId	GUID?	If filled in: e-tickets regarding this organised visit	
Paging	PagingCriteria	Response will only contain data from chosen page	
Includes	ETicketSearchCriteriaIncludes	Property that allows you to include various data in the response	

ETicketSearchCriteriaIncludes

Name	Type	Description
IncludeImage	Bool	Indicates whether the image should be retrieved

Response

The response will contain a list of **Eticket** objects.

Sale

Name	Type	Description	Remarks
Id	GUID	Unique ID of the e-ticket.	
TicketNo	String	Number of the e-ticket.	
FirstName	String	First name of the linked person.	
MiddleName	String	Middle name of the linked person.	
Name	String	Last name of the linked person.	
Initials	String	Initials of the linked person.	
ContactTitle	String	Contact title of the linked person.	
TicketDescription	String	The e-ticket description.	
TotalPrice	Decimal	Total price which was payed for the e-ticket.	
ItemPriceInc	Decimal	Price VAT inclusive.	
ItemPriceExc	Decimal	Price VAT exclusive.	
CreationDate	DateTime	The date the e-ticket was created.	
TemplateId	Guid	Unique ID of the e-ticket template.	
CardNumber	String	Card number of the e-ticket.	
WebImage	Byte[]	Image for the e-ticket.	

ExpositionName	String	Name of the linked exposition.	(if available)
ExpositionFrom	DateTime	Start date of the linked exposition.	(if available)
ExpositionUntil	DateTime	End date of the linked exposition.	(if available)
ExtraInfo1	String	Extra information.	
ExtraInfo2	String	Extra information.	
ExtraInfo3	String	Extra information.	
Quantity	Decimal	Number of e-tickets.	
StartDate	DateTime	Start date of the e-ticket.	
EndDate	DateTime	End date of the e-ticket.	
DonationPrice	Decimal	Donation price	
ItempriceIncWithoutDonation	Decimal	Item price inc without donation	
ItemPriceExcWithoutDonation	Decimal	Item price exc without donation	
ExpositionTypeDescription	String	Description of the exposition type	
ExpositionTypeShortName	String	Shortname of the exposition type	
ExpositionInfo1	String	Info field 1 exposition	
ExpositionInfo2	String	Info field 2 exposition	
ExpositionInfo3	String	Info field 3 exposition	
ExpositionInfo4	String	Info field 4 exposition	
ExpositionInfo5	String	Info field 5 exposition	
ExpositionComment	String	Comment exposition	
NumberOfDays	Decimal	Number of days	
NumberOfTurns	Decimal	Number of turns	
RemainingTurns	Decimal	Remaining turns	
Valid	Bool	Is valid	
Street	String	Street	
Number	String	Number	
ZipCode	String	Zipcode	
City	String	City	
Country	String	Country	
Box	String	Box	
Active	Bool	Is active	
ExpositionId	Guid	ID Exposition	
Contact	Contact	Contact	
AnonymousPerson	Anonymous	Anonymous	
OrderNumber	String	Ordernumber	
Coupon	String	Coupon	
PersonalizedMessage	String	Personalized message	
Audience	Audience	The audience	
ArticleId	Guid	The article id	
OnlinePaymentOrderId	String	The online payment order id	
ExpositionPeriodComment	String	The comment of the ExpositionPeriod linked to the ETicket	
VisitNo	long	The Visit number linked to the ETicket	
LessonGroupShortNames	string	Concatenation of all LessonGroup ShortNames separated by |	
LessonGroupDescriptions	string	Concatenation of all LessonGroup Descriptoins separated by |	

ExternalTicketData	ExternalTicketData	External ticket data (e.g. NMBS)	
CombiExpositionDescription	string	Description of the Combi Exposition linked to the ETicket	
CombiExpositionTicketDescription	string	TicketDescription of the Combi Exposition linked to the ETicket	

ExternalTicketData

Name	Type	Remarks
Barcode	string	
DepartureStationName	string	
DestinationStationName	string	
TravelDate	DateTime	

JourneyType	JourneyType	Unknown = 0,
		SingleTrip = 1,
		RoundTrip = 2

TravelClass	TravelClass	Unknown = 0,
		FirstClass = 1,
		SecondClass = 2

ShortSmsCode	string	
LongSmsCode	string	
ProductId	long	
TicketNumber	string	

FindPerson

Request

The request consists of a service context and a FindPersonsCriteria object.
The properties used for a FindPersonsCriteria object look as follows:

FindPersonsCriteria

Name	Type	Description	Remarks
ActivationId	GUID?	ID to activate the user account.	Is not yet used.
Id	GUID?	The unique ID of the user you want to search.	Nullable, not obligatory.
UserName	String	The user name of the user you want to search.This is the same UserName as the one of the Credential object.	Not obligatory.
Paging	PagingCriteria	See 4.2: Paging Criteria	
Email	String	The e-mail address of the person you want to search.	
Includes	PersonIncludes	Determines which parts of the person will be included.	
IsBlocked	Boolean	Determines of blocked persons have to be called up.	
NationalNumber	String	National register number of the user you want to search.	
ResetPasswordId	GUID?	The unique code to reset a password on which you want to search.	
LessonGroupId	GUID?	If filled in, only the students of this lesson group will be called up.	
IncludeImage	Bool?	If true, the image is called up.	
Name	String	All persons of who the family name contains the given part are called up.	

FirstName	String	All persons of who the first name contains the given part are called up.	
ExternalMasterId	String	Calls up the persons with this exact ExternalMasterID.	
IncludeImageURI	Bool?	If true, an URL to the imaging service is given	
ExcludeCredentials	Bool?	If true, the credential object (username & password) is not returned	
OnlyPersonsWithEmptyPassword	Bool?	If true, only persons with an empty password are returned	
DateFromWhenOnlyPersonsWithEmptyPassword	DateTime?	If OnlyPersonsWithEmptyPassword is true, only persons with an empty password that were not updated after this date are returned	
GetAnonymousPerson	Bool?	If true, only the person that is used for temporary locking is returned	
LLVRightsEmployeeID	Guid?	The employee that is used for the LLV rights	
GroupID	Guid?	The person group ID of the user(s) you want to find	
Type	PersonType?	The person type of the user(s) you want to find • Unknown • Individual • Group • Company • Location • Physician • School	
AgeCategoryId	Guid?	Filter on persons without a birthdate or a birthdate/age within the age category settings	
`ModifiedSince`	DateTime?	Only persons that were updated after this date are returned	This date is included
AddressSubCategoryId	Guid?	Only persons that are linked to the subcategory will be returned	
AddressCategoryId	Guid?	Only persons that are linked to the category will be returned	
OrganisationNumber	string	the address with this organisation number will be returned	

PersonIncludes

This object allows to exclude certain parts of a person.

Name	Type	Description	Remarks
Categories	Bool	Calls up the subcategory of a person.	
PriceGroups	Bool	Calls up the price groups of a person.	
PersonLLVInfos	Bool	Calls up the specific lesson group info of a person.	
PersonLLVTransferHistories	Bool	Calls up the course management transfer history of a person.	
PersonLLVCurrentLessonGroups	Bool	Calls up the current lesson groups of a person.	
Relations	Bool	Calls up all children-relations of a person.	
ChildCareInfo	Bool	Calls up extra information concerning after-school child care.	
ParentRelations	Bool	Calls up all parent-relations of a person.	
Doctor	Bool	Calls up the related doctor person.	
ClubInfo	Bool	Calls up the club information	Only available for Type = Group
ChildCareInfoForChildren	Bool	Calls up extra information concerning after-school child care for all relations	Only applicable when include Relations = True
LlvCredits	Bool	Calls up the total LLV credits for a person.	
CapScanAddress	Bool	Calls up the CapScan Address	
CompanyInfo	Bool	Calls up the company information linked to the person	
CompanyPeople	Bool	Calls up the people linked to the company	

Response

The response visualises a list of Person objects that meet the FindPersonsCriteria.

See SavePerson for more information about the Person object.

FindPersonBase

Request

The request consists of a service context and a FindPersonsBaseCriteria object.
The properties used for a FindPersonsBaseCriteria object look as follows:

FindPersonsBaseCriteria

Name	Type	Description	Remarks
Id	GUID?	The unique ID of the user you want to search.	Nullable, not obligatory.
Paging	PagingCriteria	See 4.2: Paging Criteria	
Name	String	All persons of who the family name **equals** the given part are called up.	
FirstName	String	All persons of who the first name **equals** the given part are called up.	
ZipCode	String	All persons of who the zipcode **equals** the given part are called up.	
Street	String	All persons of who the street **equals** the given part are called up.	

Response

The response visualises a list of PersonBase objects that meet the FindPersonsBaseCriteria .

PersonBase

Name	Type	Description	Remarks
Id	Guid	Identifier of the person	
Name	PersonName	The PersonName object contains all info about the name of the person (firstname, name, initials, middlename, ...)	
Address	Address	The Address object contains all info about the address of the person (street, houseno, zipcode, ...)	
BirthDate	DateTime?	The birthdate of the person	Not mandatory -> Nullable DateTime object
Email	string	The email of the person	

FindPersonContacts

This method is used for calling up contacts of a person.

Request

The request consists of a service context and a PersonContactSearchCriteria object.
The properties used for a PersonContactSearchCriteria object look as follows:

Name	Type	Description	Remarks
Id	Guid?	Unique ID of the contact.	Nullable, not obligatory.
PersonId	Guid?	ID of the person.	Nullable, not obligatory.
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	PersonContactIncludes		

The properties used for a PersonContactIncludes object look as follows:

Name	Type	Description	Remarks
IncludeImage	Bool	Include the full byte array of the image	
IncludeImageUrl	Bool	Include the image URL (to be used in the image service)	
IncludeRelations	Bool	Include the related persons	

Response

The response visualises a list of Contact objects that meet the given criteria.
This object looks as follows:

Name	Type	Description	Remarks
Id	GUID	Unique ID of the contact.	
Title	PersonTitle	The title of the contact.	e.g. doctor, sir, …
Initials	String	The initials of the contact.	
Name	PersonName	The name of the contact.	
Function	PersonFunction	The function of the contact.	
FunctionComments	String	The comment of the function of the contact.	
Location	Address	The adres of the contact.	
Telephone1	String	Telephone1 of the contact.	
Telephone2	String	Telephone2 of the contact.	
Telephone3	String	Telephone3 of the contact.	
Fax	String	The fax of the contact.	
Email	String	The email of the contact.	
Comment	String	Comment of the contact.	
RegisterNumber	String	The registernumber of the contact.	
IsGeneral	Bool	Shows if the contact is used for general communication.	
IsInvoice	Bool	Shows if the contact is used for communication concerning invoices.	
IsReservation	Bool	Shows if the contact is used for communication concerning reservations.	
IsMailing	Bool	Shows if the contact is used for mailings.	
Disabled	Bool	Indicates if the contact is a disabled person or not.	
Gender	Gender	The gender of the contact.	Enumerator: Male=0, Female=1, Unknown=2
BirthDate	DateTime?	The birth date of the contact.	
Image	Byte array	The image of the contact.	
ImageUrl	String	Url to the imaging service for this contact.	
IsTicketing	Boolean	Shows if the contact is used for communication concerning ticketing.	
IsExhibitions	Boolean	Shows if the contact is used for communication concerning exhibitions.	
IsEmployee	Boolean	Shows if the contact is used for communication concerning employee.	
IsAdministrator	Boolean	Shows if the contact is used for communication concerning administrator.	
Credential	Credential	The credentials of the contact	
Relations	List<Person>	The relations of the contact	To which persons is the contact linked
PersonId	Guid	The person who they are a contact of	

PersonTitle

This is a description of the PersonTitle object.

Name	Type	Description	Remarks
ID	Guid	The ID of the title of the person.	
ShortName	String	The shortname of the title of the person.	
Description	String	The description of the title of the person.	

FindPersonDeliveryAddresses

This method is used for calling up delivery addresses (for a specific customer or not).

Request

The request consists of a service context and a DeliveryAddressSearchCriteria object.
The properties used for a DeliveryAddressSearchCriteria object look as follows:

Name	Type	Description	Remarks
Id	GUID?	Unique ID of the delivery address.	Nullable, not obligatory.
PersonId	GUID?	ID of the person.	Nullable, not obligatory.
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response visualises a list of DeliveryAddress objects that meet the given criteria.
This object looks as follows:

Name	Type	Description	Remarks
Id	GUID	Unique ID of the delivery address.	
PersonId	GUID	ID of the person linked to the delivery address.	
Description	String	Description of the delivery address.	
Address	Address	Details of the address data.	See 8.8.1.3 for details concerning the Address object.

FindSales

This method is used for finding detailed sales information.

Request

The request consists of the service context and a **FindSalesCriteria** object.

FindSalesCriteria

Name	Type	Description	Remarks
Id	GUID	Unique ID of the sale.	
CustomerId	GUID	Id of the customer (person).	
From	DateTime	If filled in: Only sales that were registerd after the given date will be returned.	
Until	DateTime	If filled in: Only s ales that were registerd before the given date will be returned.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
OnlinePaymentOrderId	String	If filled in: Only sales that have this online payment order id	
OrderNumber	String	If filled in: Only sales that have this order number	
EmployeeShiftId	GUID	The ID of the shift	
EmployeeId	Guid?	When the employee is provided, you can only load the maximum sales this employee can see in history (setting in ReCreateX)	
PointOfSaleId	Guid?	Only load the sales sold on this point of sale	
RefundReference	String	Only returns sales when one of their payment lines contains the mentioned refund reference	

Response

The response will contain a list of **Sale** objects.

Sale

Name	Type	Description	Remarks
Id	GUID	Unique ID of the sale.	
Number	Long	Sales number.	
Date	DateTime	Date of sale registration.	
CustomerId	GUID	Id of the customer (person).	
Lines	List<SalesLine>	List of SalesLines which define the sales details.	
SalesSeriesID	Guid	The serie linked with the sale	
PointOfSaleId	Guid	The unique Id of the Point Of Sale it was sold on	
PaymentLines	List<PaymentLine>	List of PaymentLines which define the sales payment details	

The **SalesLine** object looks as follows:

SalesLine

Name	Type	Description	Remarks
Id	GUID	Unique ID of the sales line.	
SequenceNumber	Int	Sequence number of the sales line within the sale.	
Quantity	Decimal	Number of bought products.	
UnitPrice	Decimal	Unit price of the sales line.	
TotalPrice	Decimal	Total price of the sales line.	
Description	String	Detailed information about the sales line.	
MainOrganisedVisitId	Guid	The OrganisedVisit of the SalesLine's main salesline	

A SalesLine can also be of type

- ArticleSalesLine
- ActivityReservationSalesLine
- ChildCareSalesLine
- ExhibitionSalesLine

with following additional properties:

ArticleSalesLine

Describes the sales line of an article sale.

Name	Type	Description	Remarks
ArticleId	GUID	Unique ID of the article.	

ActivityReservationSalesLine

Describes the sales line of an activity reservation.

Name	Type	Description	Remarks
ActivityReservationId	GUID	Unique ID of the activity reservation.	

ChildCareSalesLine

Describes the sales line of a child care presence.

Name	Type	Description	Remarks
ChildCareCentreId	GUID	Unique ID of the child care centre.	
ChildCareCentreCode	String	Code of the child care centre.	
ChildCareCentreName	String	Name of the child care centre.	

PersonId	GUID	Unique ID of the person.	
PersonName	String	Name of the person.	
Day	DateTime	Day of the child care presences.	

ExhibitionSalesLine

Describes the sales line of an article sale.

Name	Type	Description	Remarks
ExhibitionId	GUID	Unique ID of the Exhibition.	

PaymentLine

Name	Type	Description	Remarks
Id	GUID	Unique ID of the payment line.	
PaymentMethodId	GUID	Unique ID of the payment method	
Amount	Decimal	Amount paid in the payment	
RoundedAmount	Decimal	Rounded amount paid in the payment	
IsDigitalWalletPayment	bool	Determines whether the payment is Digital Wallet	
IsGiftCertificatePayment	bool	Determines whether the payment is GiftCertificate	
OnlinePaymentOrderId	string	The Id of the online paymentOrder	
OnlinePaymentTrxId	string	The Id of the online paymentTrx	
OnlinePaymentPayId	string	The Id of the online PaymentPayId	

ForgotPassword

Request

The request consists of a service context and the login of the user.

Response

The response is a ForgotPasswordResponse that consists of an EmailAddress and a Password.

Name	Type	Description	Remarks
EmailAddress	String	The e-mail address of the user.	
Password	String	The password of the user.	

GenLoyaltyCardIsValid

This method is used to validate a general loyalty card (e.g. "A-kaart"). The card number is validated and the profile is checked.

Request

The request consists of the service context and a **GenLoyaltyCardIsValidCriteria** object.

Name	Type	Description	Remarks
CardNumber	String	General loyalty card number	
Profile	String	General loyalty card profile string to check	

Response

The response will contain a **GenLoyaltyCardIsValidResult** object.

Name	Type	Description	Remarks
Valid	Bool	Valid card and profile check	
InvalidCardReason	GenLoyaltyInvalidCardReason	Reason for invalid card Enumeration of following values: • None • InvalidCardNumber • CardDisabled • ProfileMismatch • ProfileExpired • InternalError	If Valid = True, then None
ParticipantId	String	Participant id. string	Participant = card owner

GetPersonChipKnip

This method is used for calling up the e-purse (and if required its history) of a certain person.

Request

The request consists of the service context and a **PersonChipKnipSearchCriteria** object.

Name	Type	Description	Remarks
PersonId	Guid	ID of the person of who you want to call up the e-purse data.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	PersonChipKnipIncludes	Determines which parts of the data will be included.	

The **PersonChipKnipIncludes** object looks as follows:

Name	Type	Description	Remarks
History	Bool	Determines if the entire history has to be included.	
ChildCareInfo	Bool	Determines if extra information concerning the child care module has to be included.	

Response

The response will contain a **PersonChipKnip** object.

Name	Type	Description	Remarks
PersonId	Guid	ID of the person.	
Balance	Decimal	Current e-purse balance.	
History	IList<PersonChipKnipLine>	Historic overview of the e-purse transactions.	
ChildCareInfo	PersonChipKnipChildCareInfo	Extra child care information.	
SalesId	Guid	Unique ID of the corresponding sale.	Not always filled

The **PersonChipKnipLine** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	Unique ID of the e-purse transaction.	
Date	DateTime	Date and time of the transaction.	
Amount	Decimal	Amount that was charged during this transaction (e.g.: +100 or -50).	
Balance	Decimal	Balance after this transaction (= e-purse balance at that moment).	

The **PersonChipKnipChildCareInfo** object looks as follows:

Name	Type	Description	

			Remarks
ChildCareCommission	Decimal	The standard commission of the customer. If the e-purse balance falls below this barrier, the system will ask to upgrade the e-purse to this amount.	
ChildCareTreshold	Decimal	The barrier of the e-purse commission. If the e-purse balance falls below this barrier, the system will ask to upgrade the e-purse to the standard commission of the customer.	

GetPersonCredits

This method is used for calling up the credits (and if required the history of the credits) of a certain person.

Request

The request consists of the service context and a **PersonCreditsSearchCriteria** object.

Name	Type	Description	Remarks
PersonId	GUID	ID of the person of who you want to call up the credits.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	PersonCreditsIncludes	Determines which parts of the data will be included.	

The **PersonCreditsIncludes** object looks as follows:

Name	Type	Description	Remarks
History	Bool	Determines if the entire history has to be included.	

Response

The response will contain a **PersonCredits** object.

Name	Type	Description	Remarks
PersonId	GUID	ID of the person.	
Balance	Int	Current credits balance.	
History	IList<PersonCreditLine>	Historic overview of the credits.	

The **PersonCreditLine** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	Unique ID of the credits transaction.	
Date	DateTime	Date and time of the transaction.	
Amount	Int	Amount that was charged during this transaction (e.g.: +10 or -5).	
Balance	Int	Balance after this transaction (= credits balance at that moment).	

GetResellerSettings

This method is used for calling up the reseller settings.

Request

The request consists of the service context and a **GetResellerSettingsCriteria** object.

Name	Type	Description	Remarks
ResellerId	GUID	ID of the reseller (company).	
Includes	ResellerSettingsIncludes	Object with all the includes	

ResellerSettingsIncludes

Name	Type	Description	Remarks
IncludeArticlePeriods	bool	Determines if the article periods will be included in the response	

Response

The response will contain a **ResellerSettings** object.

Name	Type	Description	Remarks
Active	bool	Is the reseller website active	
WebPeriod	Period	Determines the period when the reseller website is active	
WelcomeText	string	The welcome text of the reseler	
Banner	string	The banner of the reseller	
LoginRequired	bool	Determines if the contacts must login	
ArticlePeriods	IList<ResellerProductPeriod>	List with all the article periods	
UseArticles	bool	Is the reseller allowed to use the module articles?	
UseExpositions	bool	Is the reseller allowed to use the module expositions?	
UseTicketing	bool	Is the resller allowed to use the module ticketing?	

ResellerProductPeriod

Name	Type	Description	Remarks
ProductId	Guid	ID of the article	
Period	Period	Determines the availability period of the article	
TicketsForEmployee	int	Determines how many tickets the employee can buy	
TicketsForCompany	int	Determines how many tickets the reseller (company) can buy	

ListBankBICs

This method is used to list all bank BIC numbers. These BIC numbers are used within the person object.

Request

The request consists of a service context and a FindBankBICsCriteria object.
The properties used for a FindBankBICsCriteria object look as follows:

Name	Type	Description	Remarks
Id	GUID?	Unique ID of the BIC number.	Nullable, not obligatory.
Bank	String	Name of the Bank to which the BIC number applies.	
CountryId	GUID?	Unique UD of the country to which the BIC number applies.	Nullable, not obligatory.
BIC	String	The BIC number itself	

Response

The response visualises a list of BankBIC objects that meet the given criteria.
This object looks as follows:

Name	Type	Description	Remarks
Id	GUID	Unique ID of the BIC.	
Bank	String	Name of the bank to which the BIC number applies.	
Country	String	Country code to which the BIC number applies	
BIC	String	The BIC number itself	

ListCategories

The ListCategories call is used to call up a list of categories.

Request

The ListCategoriesRequest contains the standard service context object.

Response

The ListCategoriesResponse contains a list of Category objects.

Category

Name	Type	Description	Remarks
Id	Guid	The id of the category	
Code	string	The code of the category	
Name	string	The name of the category	
Subcategories	ICollection<SubCategory>	The id of the AddressCategory	
MinSubCategories	int	The minimum of subcategories	
MaxSubCategories	int	The maximum of subcategories	
ExtraDescription	string	Extra description	

SubCategory

Name	Type	Description	Remarks
Id	Guid	The id of the subcategory	
Code	string	The code of the subcategory	
Name	string	The name of the subcategory	
CategoryId	Guid	The id of the category	
Category	Category	The category	
ExtraDescription	string	Extra description	

Profile – Brief overview

This section contains operations concerning the user profiles.
This consists of authentication operations and user profile operations.

RequestAuthenticationToken

This operation will return a token for a specific user. The token is encrypted as a unique GUID by which the can then authenticate using the **AuthenticateToken** method. A token is only valid for a certain time span.

Request

The request consists of the service context and a RequestAuthenticationTokenRequest object.
The properties used for a RequestAuthenticationTokenRequest object look as follows:

Name	Type	Description	Remarks
Credentials	Credential	Contains the user name and password combination one wants to authenticate. See amongst others 8.3.1.	
UniqueKey	String	An optional key representing the token request.	
Duration	Int	Number of seconds that the token remains valid.	

Response

The response consists of an RequestAuthenticationTokenResult that indicates whether the user is authenticated successfully or not.

Name	Type	Description	Remarks
AuthenticationResult	AuthenticationResult	Contains details about the authentication that was performed. See amonst others 7.4.2.	

Token	Guid	The unique token that can be used to authenticate.	This is only filled when authentication was succesfull.

ResetPersonPassword

Request

The request consists of a service context and a ResetPersonPasswordCriteria object containing all ids for the persons which need to be reset.

Name	Type	Description	Remarks
PersonIdList	List<Guid>	The list of person Ids	

Response

The response is a ResetPersonPasswordResult object.

Name	Type	Description	Remarks
HasSucceeded	Bool	Indicates if all passwords have been successfully reset.	
List<Person>	List<Person>	List of all persons for which the password was reset.	

SaveCeviPersons

This operation is used to create or modify a user in the ReCreateX system.
This operation may only be used by the CEVI service. Please use the regular SavePerson operation.

Request

The request consists of the service context and a CeviPerson object .
The characteristics used for a CeviPerson object look as follows:

CeviPerson

Name	Type	Description	Remarks
Name	PersonName	The name of the user.	
Address	Address	The address of the user.	
BirthDate	DateTime?	Date of birth	
Gender	Gender	Gender	
DateDeceased	DateTime?	Deceased date	
HeadOfFamily	bool?	Is the person head of family?	
Parent	bool?	Is the person a parent?	
RelationCode	string	Type of relation	
RelationText	string	Type of relation	
Reference	string	Reference	
FirstNames	string	Contains the first name of the person	
RegisterNumber	string	The registernumber of the preson	

Response

The response will return a SaveCeviPersonsResult object. This object contains the possibly ValidationResults that indicate what went wrong.

SavePersonResult

Name	Type	Description	Remarks
ValidationResults	List<ValidationResult>		

SavePerson

This operation is used to create or modify a user in the ReCreateX system.

Request

The request consists of the service context, a Person object and a SavePersonOptions object.
The characteristics used for a Person object look as follows:

Person

Name	Type	Description	Remarks
ActivationId	GUID	Unique ID to activate the user account.	Is not used in Webservices.
Id	GUID	The unique identifier of the user.	Will be empty at creation.
Code	String	A unique code of the user.	
Name	Person Name	The name of the user.	
Address	Address	The address of the user.	
Settings	Person Settings	Extra settings of the user.	
Email	String	The e-mail address of the user.	
Email2	String	The secondary e-mail address	
BirthDate	DateTime?	The birth date of the user.	Nullable
Gender	Gender	The gender of the user.	Enumerator: Male=0, Female=1, Unknown=2
Language	String	The ISO code of the user's language.	e.g. 'NL', 'FR', …
Picture	Byte array	The image of the user.	Submitting a picture with an empty/null byte array will keep the original picture. Submitting null or a picture as pictureobject will replace the old picture.
Phone	String	Telephone of the user.	
Phone2	String	Secondary telephone number	
CellPhone	String	Mobile phone of the user.	
BankAccount	String	Bank account number of the user.	
NationalNumber	String	National register number of the user.	National register number in Belgium.
Comments	String	Remarks of the user.	
Credential	Credential	Login data of the user.	To be filled in obligatory.
IsBlocked	Boolean	Indicates if the user is blocked or not.	
Type	Person Type	Indicates which type is concerned: • Unknown • Individual • Group • Company • Location • Physician • School	
Parent	Boolean	Indicates if the user is a parent or not.	
Disabled	Boolean	Indicates if the user is a disabled person or not.	
HeadOfFamily	Boolean	Indicates if the user is the head of the family or not.	
Relations	List<Relation>	Family relations of the user linked to this person. (>)	e.g. Children of the person
ExternalMasterID	String	ExternalMasterID field of the customer card.	
ParentRelations	List<Relation>	Family relations of the user that arrive at this person. (<)	e.g. Parents of the person

PictureUrl	String	Url to the imaging service for this person	
Title	Person Title	The title of the person	e.g. doctor, sir, …
IBAN	String	IBAN account number of the person	
BIC	BankBIC	BIC number of the person	
AllowEdit	Boo l	Is the person allowed to edit it's details via web?	
IsWebBlocked	Bool	Is the person blocked for web usage?	
Group	Person Group	The person group details	
Doctor	Person	The person's doctor	
Website	String	The website URL for the person	
ClubInfo	ClubInfo	Details about the club information	Only available when the Type is "Group"
PasswordIsChangedByUserOnWeb	Bool	Is the password changed?	
ExternCustomerNumber	String	The external customer number	
LastPasswordUpdate	DateTime?	The date & time the password was last modified	
TotalLlvCredits	Decimal	The total llv credits for this person.	
VatNumber	string	The VAT number	
EnterpriseNumber	string	The enterprise number	Only applicable when Type is not "Individu"
BlockSynchronisation	Bool	Determines if the address data will NOT be synchronized with the population register	
`CreationDate`	DateTime?	The date and time of the creation of the person	
LastUpdate	DateTime?	The date and time the person was last updated	
OrganisationNumber	String	The number of your organisation	

PersonName

Name	Type	Description	Remarks
First	String	The first name of the user.	
Middle	String	The middle name of the user.	
Last	String	The family name of the user.	
Initials	String	The initials of the user	

Address

Name	Type	Description	Remarks
Street	String	The street of the user's address.	
Number	String	The house number of the user's address.	
Box	String	The box of the user's address.	
ZipCode	String	Postal code of the user's address.	
Town	String	Town of the user's address.	
Country	String	ISO code of the country of the user's address.	e.g. 'BE', 'NL', 'FR', …
LocationName	String	For the moment the name of the location is only filled in for the location an Activity level.	

PersonSettings

Name	Type	Description	Remarks
SubscribeMailingList	Boolean	Register for the newsletter.	
Subcategories	List<Subcategory>	List of subcategories to which the person belongs.	
PriceGroups	List<PersonPriceGroup>	List of the price groups linked to the person.	
PersonLLVInfos	List<PersonLLVInfo>	List of specific swimming lesson information.	As of RCX 5.0.5
PersonLLVTransferHistories	List<PersonLLVTransferHistory>	List of the transfer history of the customer.	As of RCX 5.0.5
PersonLLVCurrentLessonGroups	List<LessonGroup>	List of all lesson groups for which the person has registered.	As of RCX 5.0.5
PersonChildCareInfo	PersonChildCareInfo	Object with extra information concerning after-school child care.	
IsInvoiceMailingAllowed	Bool?	Indicates if invoices can be sent by email	

PersonPriceGroup

Name	Type	Description	Remarks
Id	GUID	Unique ID	
Type	PersonPriceGroupType	Enumeration of the ReCreateX module in which this price group is active. • PlaceReservations = 0 • Ticketing = 1 • Entries = 2 • Sales = 3 • Rental = 4 • Wellness = 5	• Facility bookings • Ticketing • Registrations • Sales (& expositions) • Rental • Fitness/wellness
PriceGroup	PriceGroup	The price group.	
StartDate	DateTime	Start date of the period during which the price group is active.	Nullable
EndDate	DateTime	End date of this period.	Nullable
WeekDay	WeekDay	Enumeration of the day of the week on which the price group is active. • EveryDay • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday • Sunday	• Every day • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday • Sunday
From	String	String notation of the hour on which this price group will be active.	Notation 00:00:00 = PT00H00M00S
Till	String	End time.	Notation 23:59:59 = PT23H59M59S

PersonLLVInfo

Name	Type	Description	Remarks
Id	GUID	Unique ID	
NumberPresent	Int	1. presences in lesson group.	
StartDate	Datetime	Start date lesson group.	
LastDate	Datetime	Last registration	
StartDateLLV	Datetime	Start date swimming lessons.	
NumberPresentLevel	Int	1. presences on this level.	
StartDateLevel	Datetime	Start date on this level.	
NumberAbsent	Int	1. absences in lesson group.	

NumberRegistrations	Int	1. registrations on this level.	
SwimmingComment	String	Swimming lesson remarks of the student.	

PersonLLVTransferHistory

Name	Type	Description	Remarks
Id	GUID	Unique ID	
Date	Datetime	Transfer date.	
FromGroup	String	Description of the lesson group out of which the student has been transferred.	
FromGroupCode	String	Code of the lesson group out of which the student has been transferred.	
ToGroup	String	Description of the lesson group to which the student has been transferred.	
ToGroupCode	String	Code of the lesson group to which the student has been transferred.	

PersonChildCareInfo

Name	Type	Description	Remarks
Id	GUID	Unique ID of the person.	
School	ChildCareSchool	Details concerning the school of the person.	
Tariff	ChildCareTariff	Details concerning the tariff used for the price calculation for this person.	
LeaveAlone	Bool	May the child leave the child care centre alone?	
HasMedication	Bool	Does the child take medication?	
MedicationName	String	Name of the medication.	
MedicationDose	String	Medication dosis.	
MedicationExtraInfo	String	Extra information concerning the medication.	
HasAllergy	String	Does the child have an allergy?	
AllergyName	String	Name of the allergy.	
AllergyExtraInfo	String	Extra information concerning the allergy.	
OtherMedicalInfo	String	Extra information concerning the medical data.	
ChildCareGroup	String		
ChildCareLeaveWith	String		
ChildCareTariffs			
LeaveWithThirdParty	Bool	Child is allowed to leave the childcare with a third party	
HealthInsuranceNumber	string	The health insurancenumber of the child	
MustMakeHomework	bool	Child must do its homework in the childcare	
ChildCareExtraInfoParents	IList<ChildCareExtraInfoParent>	List of ChildCareExtraInfoParent objects	

ChildCareSchool

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the school.	
Code	String	Code of the school.	
Name	String	The name of the school.	
Location	AddressLocation	Location of the school.	See 0: Response

ChildCareTariff

Name	Type	Description	Remarks

Id	GUID	The unique identifier of the tariff.	
Code	String	Code of the tariff.	
Name	String	Name of the tariff.	
Description	String	Description of the tariff.	

Credential

Name	Type	Description	Remarks
UserName	String	User name of the user.	Has to be unique.
Password	String	Password of the user.	

PersonGroup

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the group.	
ShortName	String	Code of the tariff.	
Description	String	Description of the group	

ClubInfo

Name	Type	Description	Remarks
Info1	String	Info field 1 for club information	
Info2	String	Info field 2 for club information	
Info3	String	Info field 3 for club information	
Info4	String	Info field 4 for club information	
Info5	String	Info field 5 for club information	
Info6	String	Info field 6 for club information	
Info7	String	Info field 7 for club information	
Info8	String	Info field 8 for club information	
Info9	String	Info field 9 for club information	
Info10	String	Info field 10 for club information	
Info11	String	Info field 11 for club information	
Info12	String	Info field 12 for club information	
Info13	String	Info field 13 for club information	
Info14	String	Info field 14 for club information	
Info15	String	Info field 15 for club information	
ClubLocations	List<ClubLocation>	List of locations for the club	

ClubLocation

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the club location.	
Location	AddressLocation	The location details	See 10.4.2 for the AddressLocation object.
ContractType	ClubLocationContractType	The contract type	
StartDateContract	DateTime?	Start date of the contract	
EndDateContract	DateTime?	End date of the contract	
CommentContract	String	Comment for the contract	

The **SavePersonOptions** object looks as follows:

Name	Type	Description	Remarks
CreateZipcodes	Bool	Do unknown countries and postal codes have to be created in ReCreateX? This does not happen by default.	Postal codes are only created if both **ZipCode, Town** and **Country** have been filled in.
IgnoreCredentials	Bool	Use this parameter to avoid providing the **credentials** (web username and password) when saving a person.	Fields **UserName** and **Password** are mandatory by default

ChildCareExtraInfoParent

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the tariff.	mandatory
AddressId	Guid	The unique identifier of the person.	mandatory
Benefit	ChildCareBenefit	Type of benefit.	mandatory
FatherStatute	ChildCareParentStatute	Type of statute of the father.	mandatory
MotherStatute	ChildCareParentStatute	Type of statute of the mother.	mandatory
StartDate	Datetime	Begindate of the family situation	Mandatory
EndDate	DateTime	Begindate of the family situation	
NumberOfChildren	Int	Number of children in the family	
Category	String		
ProofChildBenefit	bool	Is there proof that of the granted benefit?	
ProofFamilyComposition	Bool	Is there proof of the family composition?	

ChildCareBenefit

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the benefit.	
Code	String	Code of the benefit.	
Description	String	Name of the benefit.	

Response

The response will return a SavePersonResult object. This object contains the Person object and possibly ValidationResults that indicate what went wrong.

SavePersonResult

Name	Type	Description	Remarks
Person	Person	The manipulated Person object.	
ValidationResults	List<ValidationResult>		

ValidationResult

Name	Type	Description	Remarks
BrokenRuleName	String	The error code.	
Message	String	The standard error.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	

SavePersonContacts

Request

The request consists of a service context and a Person object. It is important that the Contacts within this Person object are filled in correctly. Optionally, it is possible to modify or update the existing contacts.

- In case of modification the rule is applied: The contacts that are present both in the Person object as in the back office are modified. (If not, the existing contacts are not modified.)
- In case of an update, the rule is applied: Contacts that are present in ReCreateX but not in the Person object, will be deleted in ReCreateX. (If not, the existing contacts that are not present in the Person object are left unchanged.)
- Combinations of both settings are possible.

Name	Type	Description	Remarks
Person	Person	The input: Person object with a list of the contacts to be modified (New and existing).	
UpdateExisting Contacts	Bool	Existing contacts are modified.	
CleanupExisting Contacts	Bool	Missing existing contacts are deleted.	

Person

See 8.8.1.1: Person.

Response

The response will contain a SavePersonResult object (although only the contacts will be updated).

Name	Type	Description	Remarks
Person	Person	The ouput: Person object with a list of the modified contacts.	
ValidationResults	List<ValidationResult>	List of possible validation errors.	

Person

See 8.8.1.1: Person.

ValidationResult

See 8.8.2.2: ValidationResult

SavePersonDeliveryAddress

This method is used to save a new delivery address or to modify an existing delivery address.

Request

The request consists of the service context, a DeliveryAddress object and a CreateInvalidZipcodes object (bool).

Name	Type	Description	Remarks
DeliveryAddress	DeliveryAddress	Details of the address data that have to be saved.	See 8.17.2 for more details concerning this object.
CreateInvalidZipcodes	Bool	Indicates if invalid postal codes (postal codes that do not exist in ReCreateX) may be created.	

Response

The response will contain a SavePersonDeliveryAddressResult object. This object contains the DeliveryAddress object and possible ValidationResults that indicate what went wrong.

SavePersonDeliveryAddressResult

Name	Type	Description	Remarks
DeliveryAddress	DeliveryAddress	The manipulated or new DeliveryAddress object.	See 8.17.2 for more details concerning this object.
ValidationResults	List<ValidationResult>		

ValidationResult

Name	Type	Description	Remarks
BrokenRuleName	String	The error code.	
Message	String	The standard error.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	

SavePersonPriceGroups

Request

The request consists of a service context and a Person object. It is important that the PriceGroups within the PersonSettings are filled in correctly.

Response

The response will contain a SavePersonResult object. This object contains the Person object and possible ValidationResults that indicate what went wrong.
The following tests are active for the SavePersonPriceGroups:

- PersonPriceGroup has to contain a valid PriceGroup (Id).
- The price group type has to be valid within the type of the PersonPriceGroup.
- The combination of the type, start date, end date, start time, end time and day of the week has to be unique.

SavePersonRelation

Request

The Request consists of the service context, 2 different person Id's, the relation type and extra remarks.

Name	Type	Description	Remarks
PersonId1	GUID	ID of the principal person.	
PersonId2	GUID	ID of the related person.	
Comment	String	Extra remark concerning the relation between both persons.	
RelationType	RelationType?	Indicates which relation type is concerned: • Unknown • Household (=family) • Family (=relative) • Business (=business relation) • Contact	

Response

The response contains a **ValidationResult** object. This object looks as follows:

Name	Type	Description	Remarks
BrokenRuleName	String	Name of the validation rule.	
Message	String	Error message.	
IsValid	Bool	Rule has been successfully processed.	If false, the other properties will contain more details.

SavePersonsubcategories

Request

The request consists of a service context and a Person object. It is important that the Subcategories within this Person object are filled in correctly.

Name	Type	Description	Remarks
Person	Person	The input: Person object with a list of the contacts to be modified (New and existing).	

Person

See 8.8.1.1: Person.

Response

The response will contain a SavePersonResult object (although only the subcategories will be updated).

Name	Type	Description	Remarks
Person	Person	The ouput: Person object with a list of the modified subcategories.	
ValidationResults	List<ValidationResult>	List of possible validation errors.	

Person

See 8.8.1.1: Person.

ValidationResult

See 8.8.2.2: ValidationResult

ValidateUserName

This operation will verify if a user name is already used or not.

Request

The request consists of the service context and a text object for the user name.

Name	Type	Description	Remarks
UserName	String	The user name that has to be validated.	

Response

The response consists of an AuthenticationResult that indicates whether the user name already exists or not.

Name	Type	Description	Remarks
HasSucceeded	Boolean	Indicates if the user name is already used or not. • True = not yet used • False = already used	
Error	AuthenticationError	An enumeration that returns the error of the AuthenticationProcess. Possible values are: • None • InvalidParameters • UserNameExists	For security purposes no correct information is given here.
PersonId	GUID	In this case always GUID.Empty.	Is not used for this method.
LastLoginDate	DateTime?	Indicates when the user authenticated himself the last time.	If empty: first time.

9 Lessons module (as of RCX 5.0.5)

CatchUpLessonGroupDay

This service method registers a user for a catch-up lesson group day.

Request

The requests consists of a CatchUpLessonGroupDayManipulationCriteria object.

CatchUpLessonGroupDayManipulationCriteria

Name	**Type**	**Description**	**Remarks**
LessonGroupId	Guid	Which lessongroup the users needs to unsubscribe from	Mandatory

StudentId	Guid	Which user needs to be unsubscribed	Mandatory
Date	DateTime	Which date and time the user wants to unsubscribe	Mandatory
Comment	String	An extra comment field	Optional

Response

The response will be a ValidationResult object.

ValidationResult

Name	Type	Description	Remarks
BrokenRuleName	String	Contains the name of the rule that could not be met. Empty if validation was successful.	
IsValid	Bool	TRUE if validation and registration was successful, FALSE if validation was not successful.	
Message	String	Contains a translated error message why the catch-up registration failed.	

DeleteWaitingListStudent

This service deletes a WaitingListStudent (student on waiting list).

Request

The request consists of the service context and a GUID, the ID of the WaitingListStudent: WaitingListStudentId.

Response

The response is a simple boolean.

FindCatchUpLessonGroupDayManipulations

This service visualises a list of CatchUpLessonGroupDayManipulations

Request

The request consists of the **service context** and an **CatchUpLessonGroupDayManipulationsSearchCriteria** object.

The **CatchUpLessonGroupDayManipulationsSearchCriteria** inherits from **LessonGroupDayManipulationSearchCriteria** (see: LessonGroupDayManipulations).

CatchUpLessonGroupDayManipulationsSearchCriteria

Name	Type	Description	Remarks
SubscriptionId	Guid?	Filter on SubscriptionId	
Includes	CatchUpLessonGroupDayManipulationIncludes	Include additional info if necessary	

CatchUpLessonGroupDayManipulationIncludes

The **CatchUpLessonGroupDayManipulationIncludes** inherits from **LessonGroupDayManipulationIncludes** (see: LessonGroupDayManipulations)

Name	Type	Description	Remarks
Subscriptions	Bool	Include the subscriptions	

Response

The response will contain a list of **CatchUpLessonGroupDayManipulation** (IList<CatchUpLessonGroupDayManipulation>).

CatchUpLessonGroupDayManipulation inherits from **LessonGroupDayManipulationBase** (see**:** LessonGroupDayManipulations)

Name	Type	Description	Remarks

Guid?	SubscriptionId	The SubscriptionId of the manipulation	Always filled in when field exists
Subscription	Subscription	The Subscription of the manipulation	Only filled in when the property Subscription is set true in the include.

FindLessonGroupComments

This service method returns a list of comments related to the lesson group.

Request

The request consists of the service context and a LessonGroupCommentSearchCriteria object.

LessonGroupCommentSearchCriteria

Name	Type	Description	Remarks
Date	DateTime	The date for which you want to get the comments	
LessonGroupId	Guid	The lesson group for which you want to get the comments	
Includes	LessonGroupCommentIncludes	Defines which comments you want to receive in the result	When not passed, all will be returned

LessonGroupCommentIncludes

Name	Type	Description	Remarks
Current	bool	Get the current lesson group comment	
Previous	bool	Get the previous lesson group comment	

Response

The response will contain a list of lesson group comments (List<LessonGroupComment>).

Name	Type	Description	Remarks
Id	Guid	ID of the lesson group comment	
LessonGroupId	Guid	ID of the lesson group	
Date	DateTime	Date of the lesson group comment	
Comment	string	Actual comment text	

FindLessonGroupOptions

This service finds all options that are linked to a certain lesson group.

Request

The request consists of the service context and a **LessonGroupOptionSearchCriteria** object.

LessonGroupOptionSearchCriteria

Name	Type	Description	Remarks
LessonGroupId	GUID	The unique identifier of the lesson group	Mandatory
PersonId	GUID	The unique identifier of the person (this id is used for price calculation when working with different price groups)	Optional
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response will contain a list of lesson group options (List<LessonGroupOption>).

LessonGroupOption

Name	Type	Description	Remarks

Id	GUID	The unique identifier of the lesson group option.	
LessonGroupId	GUID	The unique identifier of the lesson group to which the option is linked.	
ArticleId	GUID	The unique identifier of the article.	
Description	String	Description of the article.	
Price	Decimal	The unit price for the lesson group option.	Taking into account the price group of the person (if provided)
MaxOptions	Int	The maximum number of options that can be reserved.	

FindLessonGroupsForCatchUp

This service returns all lesson groups which a certain student can use for catch-up lesson registration.

Request

The request consists of the service context and a LessonGroupForCatchUpSearchCriteria object.

LessonGroupSearchCriteria

Name	Type	Description	Remarks
PersonId	Guid	The student for which you want to receive all lesson groups for catch-up.	
Date	DateTime	The day for which you want to register for catch-up.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response will contain a list of lesson groups (List<LessonGroup>).
See ListLessonGroups for more info about the LessonGroup object.

FindPersonLlvCurrentLessonGroups

This service method returns the lesson groups of a person.

Request

The request consists of the service context and a PersonLlvCurrentLessonGroupSearchCriteria.

PersonLlvCurrentLessonGroupSearchCriteria

Name	Type	Description	Remarks
PersonId	Guid	ID of the person of which the lesson group transfer history has to be called up.	Mandatory
Includes	PersonLlvCurrentLessonGroupIncludes	Additional includes to determine the response	Optional

PersonLlvCurrentLessonGroupIncludes

Name	Type	Description	Remarks
Amount	bool	Include the total amount of persons which are registerd for the lesson group?	
HeadInstructor	bool	Include the unique identifier of the head instructur of the lesson group?	
Norms	bool	Include the linked norms for the lesson group?	

Response

The response will contain a collection of LessonGroup objects.
See ListLessonGroups for more information about the LessonGroup object.

FindPersonLlvTransferHistories

This service method returns the lesson group transfer history of a person.

Request

The request consists of the service context and a PersonLlvTransferHistorySearchCriteria.

PersonLlvTransferHistorySearchCriteria

Name	Type	Description	Remarks
PersonId	Guid	ID of the person of which the lesson group transfer history has to be called up.	Mandatory

Response

The response will contain a collection of PersonLLVTransferHistory objects.

PersonLLVTransferHistory

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the transfer.	
Date	DateTime	The date and time the transfer was performed.	
FromGroup	string	The description of the lesson group the person was transferred from.	
FromGroupCode	string	The code of the lesson group the person was transferred from.	
ToGroup	string	The description of the lesson group the person was transferred to.	
ToGroupCode	string	The code of the lesson group the person was transferred to.	

FindUnsubscribeLessonGroupDayManipulations

This service visualises a list of UnsubscribeLessonGroupDayManipulations

Request

The request consists of the **service context** and an **UnsubscribeLessonGroupDayManipulationsSearchCriteria** object.

The **UnsubscribeLessonGroupDayManipulationsSearchCriteria** inherits from **LessonGroupDayManipulationSearchCriteria** (see: LessonGroupDayManipulations).

UnsubscribeLessonGroupDayManipulationsSearchCriteria

Name	Type	Description	Remarks
ReasonId	Guid?	Filter on ReasonId	
Includes	UnsubscribeLessonGroupDayManipulationIncludes	Include additional info if necessary	

UnsubscribeLessonGroupDayManipulationIncludes

The **UnsubscribeLessonGroupDayManipulationIncludes** inherits from **LessonGroupDayManipulationIncludes** (see: LessonGroupDayManipulations)

Name	Type	Description	Remarks
Reasons	Bool	Include the reasons	

Response

The response will contain a list of **UnsubscribeLessonGroupDayManipulation** (IList<UnsubscribeLessonGroupDayManipulation>).

UnsubscribeLessonGroupDayManipulation inherits from **LessonGroupDayManipulationBase** (see**:** LessonGroupDayManipulations)

Name	Type	Description	Remarks
Guid?	ReasonId	The ReasonID of the manipulation	Always filled in when field exists
Reason	Reason	The Reason of the manipulation	Only filled in when the property Reasons is set true in the include.

FindWaitinglistStudentOverview

This service generates a list of WaitingListStudent objects, these are the waiting lists on which a specific person is registered.

Request

The request consists of the service context and a WaitingListStudentSearchCriteria object.

WaitingListStudentSearchCriteria

Name	Type	Description	Remarks
PersonId	GUID	Calls up all waiting lists for which this person has already registered with registration date. WaitingListStudent object	Obligatory.
Includes	WaitingListStudentIncludes	Specifies what extra information needs to be included	

WaitingListStudentIncludes

Name	Type	Description
PreferredLessonGroups	Bool	When true, will return the list of preferred LessonGroups

Response

The response will contain a list of waiting list students (List<WaitingListStudent>).

WaitingListStudent

Name	Type	Description	Remarks
Id	GUID	ID of the waiting list student.	
WaitingListId	GUID	ID of the waiting list.	
EntryDate	DateTime	Date of registration on the waiting list.	
WaitingListCode	String	Code of the waiting list.	
WaitingListDescription	String	Description of the waiting list.	
Price	Decimal	Price VAT incl. of the price article, linked to a waiting list.	
PreferredLessonGroups	List<WaitingListStudentsLessonGroup>	A list of preferred lesson groups	

WaitingListStudentsLessonGroup

Name	Type	Description
Id	GUID	ID of the WaitingListStudentsLessonGroup.
Priority	int	The priority of the WaitingListStudentsLessonGroup
LessonGroup	LessonGroup	Information about the preferred LessonGroup

GetLlvCreditInfo

This services shows the total credits for a person.

Request

The request consists of the service context and a LlvCreditInfoSearchCriteriaobject.

LlvCreditInfoSearchCriteria

Name	**Type**	**Description**	**Remarks**

PersonId	GUID?	ID of the person of which the information has to be called up.	Optional

Response

The response will contain a collection of LlvCreditInfo objects.

LlvCreditInfo

Name	Type	Description	Remarks
PersonId	GUID	The unique identifier of the person.	
TotalCredits	decimal	The total credits for the person.	

GetLLVInfo

This services shows specific swimming lesson information of a person.

Request

The request consists of the service context and a PersonLLVInfoSearchCriteria object.

PersonLLVInfoSearchCriteria

Name	Type	Description	Remarks
PersonId	GUID?	ID of the person of which the information has to be called up.	Required

Response

The response will contain a LLVInfo object.

Name	Type	Description	Remarks
Id	GUID	Unique ID	
PersonID	GUID	Unique ID for the person	
FirstPresenceDate	DateTime	The date of the first presence in a lessongroup	
FirstPresenceLessonGroupId	Guid	The ID of the lessongroup where the first presence of a person was in	
LastPresenceDate	DateTime	The date of the last presence in a lessongroup	
LastPresenceLessonGroupId	Guid	The ID of the lessongroup where the last presence of a person was in	
TotalPresent	int	The amount of times a person has been present	
TotalNotPresent	int	The amount of times a person has been absent	
TotalCredits	decimal	The total credits for the person.	

GetPersonLLVInfoByLessonGroup

This services shows specific swimming lesson information of a person within a lesson group.

Request

The request consists of the service context and a PersonLLVInfoSearchCriteria object.

PersonLLVInfoSearchCriteria

Name	Type	Description	Remarks
LessonGroupId	GUID	ID of the lesson group of which the information has to be called up.	
PersonId	GUID?	ID of the person of which the information has to be called up.	If empty, the info of all students within the lesson group is called up.

Response

The response will contain a PersonLLVInfo object.
See 8.8.1.6: PersonLLVInfo.

LessonGroupDayManipulations

In ReCreateX it is possible to manipulate a LessonGroupDay.

Currently there are two possibilities:

- Unsubscribe (see: FindUnsubscribeLessonGroupDayManipulations)
- CatchUp (see: FindCatchUpLessonGroupDayManipulations)

For all possibilities we have provided a base class.

Request

The request consists of the service context and an object that inherits from LessonGroupDayManipulationSearchCriteria.

LessonGroupDayManipulationSearchCriteria

Name	Type	Description	Remarks
Paging	PagingCriteria	See 4.2: Paging Criteria	
Id	GUID?	Filter on Id	
PersonId	GUID?	Filter on PersonId	
LessonGroupId	GUID?	Filter on LessonGroupId	
LessonDateFrom	DateTime?	Filter on LessonDateFrom	
LessonDateTo	DateTime?	Filter on LessonDateTo	

It is possible that an object that inherits from LessonGroupDayManipulationSearchCriteria will contain an include object.
The include object (if available) will inherit from LessonGroupDayManipulationIncludes.

LessonGroupDayManipulationIncludes

Name	Type	Description	Remarks
Persons	Bool	Include the persons	
LessonGroups	Bool	Include the lessongroups	

Response

LessonGroupDayManipulationBase

The response will contain a list of objects that inherit from LessonGroupDayManipulationBase.

Name	Type	Description	Remarks
Id	Guid	The id of the manipulation	
PersonId	Guid	The personId coupled with the manipulation	
Person	Person	The person coupled with the manipulation	Only filled in when the property Persons is set true in the LessonGroupDayManipulationIncludes object.
LessonGroupId	Guid	The lessongroupId coupled with the manipulation	
LessonGroup	LessonGroup	The lessongroup coupled with the manipulation	Only filled in when the property LessonGroups is set true in the LessonGroupDayManipulationIncludes object.
LessonDate	DateTime	The lesson date when the manipulation occurs.	
Credit	Decimal	Credits gained or lost with this manipulation	

Comment	String	Additional comment	
CreatedOn Web	Bool	Indicates if the manipulation is created on web.	

Lessons – Brief overview

As of now, it is possible to use the course management (lessons) tool on the Webshop.
The system consists of 3 main parts, that is:

- Calling up course management information (provided for Website)
  - General
    - FindPersons with Include PersonLLVCurrentLessonGroups (finding the current lesson groups of a person, see FindPersonsCriteria)
    - ListLessonGroups (finding the (preferential) lesson groups)
  - Presences
    - ListAbsentReasons (list of reasons for absence)
    - List StudentPresences (overview of presences)
  - Scores
    - ListNorms (list of the standards for which scores are obtained)
    - ListStudentFollowScores (overview scores)
- Register for waiting lists
  - ListWaitingLists (finding the waiting lists)
  - ListSwimmingLevels (finding the allowed swimming levels)

The system functions as follows:

1. a. i. 1. a. i.
    1. Person chooses a waiting list from a list
    2. Person chooses his preferential lesson groups + priority
    3. Person logs in and pays

Detailed management of waiting lists and lesson groups is done in ReCreateX.

- Entering and saving course management information (for our own tablet application)
  - SaveLLVInfoToDB (saving the modified presences/scores)
  - SaveStudentTransfer (transfering students from one lesson group to another)
  - LLVMessageTranslations (specific translations for Tablet application)

ListAbsentReasons

This operation shows all reasons for absence as defined within ReCreateX.

Request

The request consists of the division's ID.

Name	Type	Description	Remarks
DivisionId	GUID	The unique ID of the division.	

AbsentReasonSearchCriteria

For the moment not yet implemented.

Response

The response is a list of **AbsentReason** objects. The **AbsentReason** object can be considered as a simple version of the **Reason** object within ReCreateX.

Name	Type	Description	Remarks
Id	GUID	The unique ID of the reason.	
ShortName	String	The unique code of the reason.	
Description	String	The description of the reason.	

ListLessonGroups

This service visualises a list of lesson groups.

Request

The request consists of the service context and a LessonGroupSearchCriteria object.
This criterion contains an object to take care of paging in case of long lists and if required a waiting list ID to only call up the preferential lesson groups of a specific waiting list.

LessonGroupSearchCriteria

Name	Type	Description	Remarks
Id	Guid?	Retrieves the lesson group with this Id	
Paging	PagingCriteria	See 4.2: Paging Criteria	
WaitingListId	GUID?	If filled in: Only call up lesson groups that are defined as possible preferential lesson group of the waiting list.	Optional
IncludeNorms	Boolean?	Show list of linked standards.	
IncludeInstructors	Bool?	Add list of instructors.	
PersonId	GUID?	Only call up the lesson groups for which this person is/was registered. <StillInLessonGroup> indicates if the person is still in the lesson group.	.
EmployeeId	GUID?	Only call up the lesson groups of which this employee is instructor.	
RightsEmployeeId	GUID?	Only call up the lesson groups from divisions that this employee has acces to.	From RCX 5.4.0
OnlyThisDivision	bool?	Only call up lessongroups that belong to the current division.	From RCX 5.4.0
DayOfWeek	int?	Only call up the lessongroups for the given day in the week and the lessongroups that are started for the current time	
IgnoreCurrentTime	Bool	If set to true, the check for the current time used in the DayOfWeek parameter is ignored	From RCX 5.12.0

Response

The response will contain a list of lesson groups (List<LessonGroup>).

Name	Type	Description	Remarks
Id	GUID	ID of the lesson group.	
ShortName	String	Code of the lesson group.	
Description	String	Description of the lesson group.	
Comment	String	Remarks concerning the lesson group.	
FullName	String	Full name consisting of code and description.	
Priority	Integer	Priority of the preferential lesson group.	Initially always 0 Allowed value 0 till 9 incl.
DivisionId	GUID	ID of the division to which the lesson group belongs.	
DefaultNumberOfLessons	Int	Assumed number of lessons in the lesson group, 0 = not specified.	
FromDate	DateTime	Start date period.	Not obligatory.
ToDate	DateTime	End date period.	Not obligatory.
LessonGroupNorms	List<Norms>	List of the linked standards.	
EntryDate	DateTime	Registration date for the lesson group. Only applicable if asked using FindPerson with Include PersonLLVCurrentLessonGroups.	
StillInLessonGroup	Bool	Registration date for the lesson group. Only applicable if asked using FindPerson with Include PersonLLVCurrentLessonGroups.	
LessonGroupInstructors	List<Employee>	Show list of instructors per lesson group.	
HeadInstructorID	Guid	ID of the headinstructor	

MaxOverBooking	int	Indicates how many students are allowed when the lessongroup is fully booked	
MaxAllowedStudents	int	Indicates what the maximum allowed students are in the lessongroup (excl. MaxOverBooking)	
MinimumAge	int	Indicates the minimum age for participants to the lesson group	

ListNorms

This operation calls up the different standards per lesson group. These are the criteria for which a child has to pass in a lesson group.

Request

The request consists of the service context and a NormSearchCriteria object.

NormSearchCriteria

Name	Type	Description	Remarks
LessonGroupId	GUID	The unique ID of a lesson group.	

Response

The response is a list of **Norm** objects. The **Norm** object can be considered as a simple version of the **SwimmingLevelNorm** object within ReCreateX.

Name	Type	Description	Remarks
Id	GUID	The unique ID of the standard.	
ShortName	String	The unique code of the standard.	
Description	String	The description of the standard.	
Comment	String	Remarks concerning the standard.	

ListStudentFollowScores

This operation calls up the scores obtained by the students for the different standards within a lesson group, per student (history) or per day (all students).

Request

The request consists of the service context and a StudentFollowScoreSearchCriteria object.

StudentFollowScoreSearchCriteria

Name	Type	Description	Remarks
LessonGroupId	Guid?	Unique identifier of the lesson group for which the scores have to be called up.	Obligatory.
AddressId	Guid?	Unique identifier of the person of whom the scores have to be called up. => overview per day	If filled in, SelectedDateTime is ignored.
SelectedDateTime	DateTime?	Date of the scores. If AddressId is not filled in, an overview is shown of all students of this lesson group on this day.	
PreviousScore	Boolean?	If true, you will see the last scores ever obtained per standard and per student until today instead of the scores of today.	Only functions if no AddressId has been specified.

Response

The response is a list of **StudentFollowScore** objects. The **StudentFollowScore** object can be considered as a simple version of the **StudentFollow** object within ReCreateX.

StudentFollowScore

Name	Type	Description	Remarks
Id	Guid	A unique identifier.	
AddressId	Guid	Unique identifier of the student.	

LessonGroupId	Guid	Unique identifier of the lesson group.	Refers to the **LessonGroup** object.
Name	String	The name of the student.	
FirstName	String	The first name of the student.	
MiddleName	String	The middle name of the student.	
Date	DateTime	Registration date.	Here, this date will always be the same.
NormScores	IList<NormScore>	List of scores for the different norms.	The number of standards can differ per lesson group, which is why 20 standards have been provided by default. On the basis of ***ListNorms*** the correct number and name can be called up. In both methods the standards are called up alphabetically according to ShortName. 0 = nothing 1 = insufficient 2 = almost sufficient 3 = sufficient

NormScore

Name	Type	Description	Remarks
NormId	Guid	Unique identifier of the norm to which the score applies to.	
Score	int	The score for the norm.	0 = nothing 1 = insufficient 2 = almost sufficient 3 = sufficient

ListStudentFollowScoresAll

This operation calls up the current and previous scores obtained by the students for the different standards within a lesson group per day (all students).

Request

The request consists of the service context and a StudentFollowScoreAllSearchCriteria object.

StudentFollowScoreAllSearchCriteria

Name	Type	Description	Remarks
LessonGroupId	Guid	Unique identifier of the lesson group for which the scores have to be called up.	Obligatory
SelectedDateTime	DateTime	Date for which the scores have to be called up.	Obligatory

Response

The response contains a **StudentFollowScoreAllScores** object.

StudentFollowScoreAllScores

Name	Type	Description	Remarks
Scores	IList<StudentFollowScore>	List of all current scores.	
PreviousScores	IList<StudentFollowScore>	List of all previous scores.	

ListStudentPresences

This operation calls up the information concerning the presences/absences of the students, per student (history) or per day (all students).

Request

The request consists of the service context and a StudentPresenceSearchCriteria object.

StudentPresenceSearchCriteria

Name	Type	Description	Remarks
LessonGroupId	GUID	ID of the lesson group of which the presences have to be called up.	Obligatory.
AddressId	GUID?	ID of the person of whom the presences have to be called up. => history	If filled in, SelectedDateTime is ignored.
SelectedDateTime	DateTime?	Date of the presences. If AddressId is not filled in, an overview is shown of all students of this lesson group on this day.	

Response

The response is a list of **StudentPresence** objects. The **StudentPresence** object can be considered as a simple version of the **StudentFollowPresent** object within ReCreateX.

Name	Type	Description	Remarks
Id	GUID	A unique ID.	
AddressId	GUID	The unique ID of the student.	Refers to the **Person** object.
Name	String	The name of the student.	
FirstName	String	The first name of the student.	
MiddleName	String	The middle name of the student.	
PresentAccessControl	Boolean	Indicates if the student entered through the access control that day.	
HasSwimmingComment	Boolean	Indicates if an extra remark was added for this student.	
Present	Boolean	Presence registered on this day.	
ReasonId	GUID	ID of the reason for which the student is absent.	Refers to the **AbsentReason** object.
LessonGroupId	GUID	ID of the lesson group.	Refers to the **LessonGroup** object.
Date	DateTime	Registration date.	

ListSwimmingDiplomas

This operation shows a list of swimming diplomas.

Request

The request consists of the service context and a SwimmingDiplomaSearchCriteria object. It is possible to call up an entire list of swimming diplomas or only the diplomas obtained by a certain person.

SwimmingDiplomaSearchCriteria

Name	Type	Description	Remarks
PersonId	GUID	The unique ID of a person.	If the PersonId is not filled in, an entire list is visualised, if not only the diplomas obtained are visualised.

Response

The response is a list of **SwimmingDiploma** objects. The **SwimmingDiploma** object can be considered as a simple version of the **SwimmingDiploma** object within ReCreateX.

Name	Type	Description	Remarks
Id	GUID	The unique ID of the swimming diploma.	
ShortName	String	The unique code of the swimming diploma.	
Description	String	The description of the swimming diploma.	
Comment	String	Remarks concerning the swimming diploma.	

ListSwimmingLevels

This service visualises a list of swimming levels.

Request

The request consists of the service context and a SwimmingLevelSearchCriteria object.

SwimmingLevelSearchCriteria

Name	Type	Description	Remarks
Paging	PagingCriteria	See 4.2: Paging Criteria	
WaitingListId	GUID?	If filled in: Only call up the swimming levels that are defined as allowed swimming levels for the waiting list.	

Response

The response will contain a list of lesson groups (List<LessonGroup>).

Name	Type	Description	Remarks
Id	GUID	ID of the swimming level.	
ShortName	String	Code of the swimming level.	
Description	String	Description of the swimming level.	
Comment	String	Remarks concerning the swimming level.	

Listwaitinglists

This service generates a list of waiting lists to which a valid 'Web price article' has been linked.

Request

The request consists of the service context and a WaitingListSearchCriteria object.

WaitingListSearchCriteria

Name	Type	Description	Remarks
Paging	PagingCriteria	See 4.2: Paging Criteria	
PersonId	GUID?	Calls up all waiting lists for which this person has already registered.	Not obligatory.
WaitingListId	GUID?	The waiting list you want to find.	
DivisionIDs	GUID?[ ]	Adds the waiting list(s) for the Divisions which ID's you fill in	

Response

The response will contain a list of waiting lists (List<WaitingList>).

Name	Type	Description	Remarks
Id	GUID	ID of the waiting list.	
ShortName	String	Code of the waiting list.	
Description	String	Description of the waiting list.	
Comment	String	Remarks concerning the waiting list.	
WebPriceItemId	GUID	ID of the price article.	Refers to the **Article** object.
Price	Decimal	Price VAT inclusive of the price article.	
WaitingListLessonGroups	List <LessonGroup>	List of the possible preferential lesson groups.	
WaitingListSwimmingLevels	List <SwimmingLevel>	List of the allowed swimming levels.	If empty => all levels are allowed.
MaxStudent	Integer	Maximal number of students allowed on the waiting list.	If 0 => unlimited.
NumberOfStudents	integer	Number of students on the waiting list.	
MaxDuration	Integer	Maximal waiting time on the waiting list.	
SwimmingLesson	SwimmingLesson	The swimming lesson linked to the waitinglist	

SwimmingLesson

Name	Type	Description	Remarks

Id	Guid	Id of the swimmingLesson	
Description	String	Description of the swimming lesson	Not obligatory.
ShortName	String	Shortname of the swimming lesson	
Comment	String	Remarks concerning the swimming lesson	
MinimumAge	Int?	Minimum age	
MaximumAge	Int?	Maximum age	

LLVMessageTranslations

This operation calls up all specific translations for our tablet application (TabletLLV).

Request

The request consists of the division's ID.

Name	Type	Description	Remarks
DivisionId	GUID	The unique ID of the division.	

Response

The response is a list of **LLVMessageTranslation** objects.

Name	Type	Description	Remarks
Id	GUID	The unique ID of the translation.	
Message	String	The name of the message to be translated.	
LanguageId	String	Unique code of the language.	
Body	String	The translated message.	

SaveLessonGroup

This service saves changes to a lessongroup.

Request

The request consists of the service context and a LessonGroup object.

LessonGroup

Only some values can be altered with this service.
When these values are not passed or passed as NULL within the request, the value will not be changed.

Name	Type	Description	Remarks
Comment	String	Commentary on the lessongroup	Required
Description	String	The description of the lessongroup	Required
ShortName	String	The ShortName of the LessonGroup	Required

Response

The response will be a SaveResult, containing a ValidationResult

SaveLLVInfoToDB

This operation is used to save entered scores and presences in the ReCreateX database => to be executed per lesson group.

Request

The request consists of a service context, a StudentPresence object, a StudentFollowScore object, a Norm object and a Person object.
In the tablet application the Save is executed when changing a lesson group, clicking on the save button and closing the program.

Name	Type	Description	Remarks
StudentPresences	List <StudentPresence>	List of the presence registrations to be saved.	

StudentFollowScores	List <StudentFollowScore>	List of the scores to be saved.	
Norms	List <Norms>	List of the standards of the lesson group that are saved.	
Students	List <Person>	List of persons of which the swimming lesson remark field has been modified. (SwimmingComment).	

Response

The response is a simple boolean.

SaveStudentTransfer

This operation is used to transfer students from one lesson group to another.

Request

The request consists of a service context, a Person object, two LessonGroup objects and an integer.

Name	Type	Description	Remarks
Person	Person	Person object that has to be transferred.	
FromLessonGroup	LessonGroup	Source lesson group.	
ToLessonGroup	LessonGroup	Destination lesson group.	
OverruleWarnings	Int	Indicates which warnings can be overruled. 0 = no overrule 1 = overrule number of places 2 = overrule mentioned above + overrule allowed swimming levels 3 = overrule mentioned above + overrule minimal age	
InstructorId			
MakeCopy	bool	0 = transfer to another lessongroup 1 = copy to another lessongroup (and stay in current)	

Response

The response consists of a StudentFollowTransferFeedback.
This is an enum: 0 = Ok
1 = Error
2 = GroupFull group full, transfer failed
3 = AskOverRuleGroupFull group full but overrule possible
4 = AskSwimmingLevelInvalid wrong level overrule possible
5 = AskMinAge wrong age overrule possible

UnsubscribeLessonGroupDay

This service method unsubscribes a user from a lesson group day.

Request

The requests consists of a UnsubscribeLessonGroupDayManipulationCriteria object.

UnsubscribeLessonGroupDayManipulationCriteria

Name	Type	Description	Remarks
LessonGroupId	Guid	Which lessongroup the users needs to unsubscribe from	Mandatory
StudentId	Guid	Which user needs to be unsubscribed	Mandatory
Date	DateTime	Which date and time the user wants to unsubscribe	Mandatory
Comment	String	An extra comment field	Optional
ReasonId	Guid?	The reason why the user wants to unsubscribe	Optional

Response

The response will be a ValidationResult object.

ValidationResult

Name	Type	Description	Remarks
BrokenRuleName	String	Contains the name of the rule that could not be met. Empty if validation was successful.	
IsValid	Bool	TRUE if validation was successful, FALSE if validation was not successful.	
Message	String	Contains a translated error message why the unsubscribe failed.	

ValidateWaitingListCall

This service validates a waitinglist call and returns information related to it

Request

The request consists of a ValidateWaitingListCriteria object.

WaitingListSearchCriteria

Name	Type	Description	Remarks
LessonGroupId	GUID	The LessonGroup the student is waiting for to subscribe	Obligatory
WaitingListStudentId	GUID	The waitinglist student you want to find.	Obligatory

Response

The response will contain a list of waiting lists (List<WaitingList>).

Name	Type	Description	Remarks
StudentName	string	name of the called student	
IsExpired	boolean	is the call expired?	
Lesson description	string	name of the lesson the student is called for	

ConfirmWaitingListCall

This service confirms (or not) a waitinglist call and returns a validation result

Request

The request consists of a ConfirmWaitingListCallCriteria object.

WaitingListSearchCriteria

Name	Type	Description	Remarks
WaitingListStudentId	GUID	The waitinglist student that has been called	Obligatory
Accepted	BOOL	Wether or not the user accepts the call	Obligatory

Response

The response will contain a list of waiting lists (List<WaitingList>).

Name	Type	Description	Remarks
ValidationResult	Object	ValidationResult Object containing: • bool IsValid • string Message (null if IsValid is true) • string BrokenRuleName (What went wrong)	

SavePreferredLessonGroups

This endpoint will attempt to save preferred LessonGroups for a given WaitingListStudent

Request

The request consists of the service context and a SavePreferredLessonGroupsCriteria object.

Name	Type	Description	Remarks
WaitingListStudentId	ID	The ID of the WaitingListStudent	
PreferredLessonGroups	List<WaitingListStudentsLessonGroup>	The List of LessonGroups that should be saved as preferred	

Response

Name	Type	Description
ValidationResult	ValidationResult	Indicates whether the Save was succesful

10 Fitness module (as of RCX 5.0.5)

CancelFitnessActivityReservation

This method is used to cancel a fitness reservation.

Request

The request consists of the ID (ID of the FitnessActivityReservation) that has to be cancelled.

Name	Type	Description	Remarks
FitnessActivityDayReservationId	GUID	ID of the fitness reservation that has to be cancelled.	

Response

The response contains a **FitnessActivityReservationValidationResult** object. This object contains validation data as well as the original **FitnessActivityReservation** object.
The FitnessActivityReservationValidationResult looks as follows:

Name	Type	Description	Remarks
BrokenRuleName	String	The code of the error message.	
Message	String	The error message.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	
ValidatedFitnessActivityReservation	FitnessActivityReservation	FitnessActivityReservation object that was called up on the basis of the given Id.	

ConfirmFitnessActivityReservation

This method is used for confirming a fitness reservation.

Request

The request consists of the ID (ID of the FitnessActivityReservation) that has to be confirmed.

Name	Type	Description	Remarks
FitnessActivityDayReservationId	GUID	ID of the fitness reservation that has to be confirmed.	

Response

The response contains a **FitnessActivityReservationValidationResult** object. This object contains validation data as well as the original **FitnessActivityReservation** object.
The FitnessActivityReservationValidationResult looks as follows:

Name	Type	Description	Remarks
BrokenRuleName	String	The error code.	

Message	String	The error message.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	
ValidatedFitnessActivityReservation	FitnessActivityReservation	FitnessActivityReservation object that was called up on the basis of the given ID.	

FindFitnessActivities

This method calls up all fitness activities on the basis of certain search criteria.

Request

The request consists of the service context and a **FitnessActivitySearchCriteria** object.

Name	Type	Description	Remarks
FitnessActivityId	GUID?	ID of a specific fitness activity (if you mean to call up only 1 entity).	Nullable (if not applicable).
FitnessActivityIds	List<GUID>	IDs of the specific fitness activities	
DescriptionPattern	String	Name or part of the description that can be searched for. For partial matches, the % sign can be used. The % sign can be used before or after the word, as well as before and after the word.	Nullable (if not applicable).
From	DateTime	If filled in: Only fitness activities that start after the given date will be shown.	
Until	DateTime	If filled in: Only fitness activities that start before the given date will be shown.	
CardId	GUID	Customer card that is used.	Nullable (if non-member).
SubscriptionId	GUID	Membership that is used.	Nullable (if non-member).
Includes	FitnessActivityIncludes	Determines which data are included in the response and/or which extra records will be present in the returnset.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
FitnessActivityCategoryId	GUID?	ID of a specific fitness activity category for which you want to get the activities	Nullable (if not applicable).
FitnessActivityCategoryIds	List<GUID>	IDs of fitness activity categories for which you want to get the activities	

The **FitnessActivityIncludes** object can be built up using the following criteria:

Name	Type	Description	Remarks
Price	Bool	Include price of the fitness activity days in the response	
Days	Bool	Include possible registration days in the response	Reservations are made on activity days.
Slots	Bool	Include possible registration slots in the response	If days include slots, one or several slots have to be reserved.
Locations	Bool	Include the locations	
Image	Bool	Find the image or not	
ImageUrl	Bool	Find the imageUrl or not	
Categories	Bool	Include the fitness activity categories	

Response

The response will contain a list of **FitnessActivity** objects that meet the criteria.
The FitnessActivity object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the fitness activity.	
Code	String	Code of the fitness activity.	
Description	String	Description of the fitness activity.	
Image	Picture	The picture of the fitness activity.	
ImageUrl	String	The link on which a picture of the fitness activity can be found.	
Location	AddressLocation	The location where this fitness activity takes place.	
Days	List<FitnessActivityDay>	List of days on which you can register for this activity.	
Color	Color	Colour that is used to visualise this fitness activity.	
WaitingListDisabled	Bool	Is the waiting list disabled for this fitness activity?	

The **AddressLocation** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the location.	
Code	String	Code of the location.	
Name	AddressName	Name of the location.	
Address	Address	Address of the location.	See 8.8.1.3 for details concerning the Address object.
Coordinates	Coordinates	The GPS coordinates for the location	

The **AddressName** object looks as follows:

Name	Type	Description	Remarks
First	String	First name of the location.	
Middle	String	Middle name of the location.	
Last	String	Surname of the location.	

The **Coordinates** object looks as follows:

Name	Type	Description	Remarks
X	String	X-coordinate	
Y	String	Y-coordinate	

The **FitnessActivityDay** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the fitness activity day.	
From	DateTime	Start date and time of the fitness activity day.	
Until	DateTime	End date and time of the fitness activity day.	
Price	Decimal	Price of the fitness activity day.	Can differ for members and non-members.
Slots	List<FitnessActivityDaySlot>	List of slots for which you can register.	If there are slots, at least one slot has to be chosen.
MaxReservations	Int	Indicates the maximal number of reservations for the activity day.	
MaxReservationsNonMember	Int	Indicates the maximal number of non-member reservations for the activity day.	
CurrentReservations	Int	Indicates the current number of reservations for the activity day.	1. Open reservations = MaxReservations - CurrentReservations
CurrentReservationsMember	Int	Indicates the current number of reservations of members for the activity day.	
CurrentReservationsNonMember	Int	Indicates the current number of reservations of non-members for the activity day.	

The **FitnessActivityDaySlot** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the slot of the fitness activity day.	
Description	String	Description of the slot.	
Reserved	Bool	Is this slot already reserved?	

The **FitnessActivityCategory** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the fitness activity category.	
Code	String	Code of the fitness activity category.	
Description	String	Description of the fitness activity category.	

FindFitnessActivityCategories

This method calls up all fitness activity categories on the basis of certain search criteria.

Request

The request consists of the service context and a **FitnessActivityCategorySearchCriteria** object.

Name	Type	Description	Remarks
FitnessActivityCategoryId	GUID?	ID of a specific fitness activity category (if you mean to call up only 1 entity).	Nullable (if not applicable).
FitnessActivityCategoryIds	List<GUID>	IDs of the specific fitness activity categories	
Keyword	String	Name or part of the description that can be searched for.	Nullable (if not applicable).

Response

The response will contain a list of **FitnessActivityCategory** objects that meet the criteria.

FindFitnessActivityReservations

This method calls up all fitness activities on the basis of certain search criteria.

Request

The request consists of the service context and a **FitnessActivitySearchCriteria** object.

Name	Type	Description	Remarks
FitnessActivityDayReservationId	GUID	ID of a specific fitness reservation (if you mean to call up only 1 entity).	Nullable (if not applicable).
PersonId	GUID	ID of the person of whom you want to list the fitness reservations.	Nullable (if not applicable).
FitnessActivityDayId	GUID	ID of the fitness activity day of wich you want to list the fitness reservations.	Nullable (if not applicable).
FitnessActivityId	GUID	ID of the fitness activity of wich you want to list the fitness reservations.	Nullable (if not applicable).
From	DateTime	If filled in: Only fitness activities that were booked after the given date will be shown.	
Until	DateTime	If filled in: Only fitness activities that were booked before the given date will be shown.	
Includes	FitnessActivityReservationIncludes	Determines which data are included in the response and/or which extra records will be present in the returnset.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

The **FitnessActivityReservationIncludes** object can be built up using the following criteria:

Name	Type	Description	Remarks
Confirmed	Bool	Include confirmed fitness reservations in the response.	
Canceled	Bool	Include cancelled fitness reservations in the response.	
PersonDetails	Bool	Find ReCreateX Person details • YES = the entire Person object is shown • NO = only the ID of the Person is shown	See 8.8.1.1 for details concerning the Person object.
Slots	Bool	Find reserved slots. • YES = all reserved slots of the reservation in question will be added to the FitnessActivityDay object. • NO = FitnessActivityDay object is shown without details concerning the reserved slots in question.	See 8.8.1.1 for details concerning the Person object.
Locations	Bool	Include linked locations in the response.	See 0 for details concerning the AddressLocation object.
Image	Bool	Find the image or not.	
ImageUrl	Bool	Find the imageUrl or not.	
WaitingList	Bool	Find fitness reservations that are on the waiting list.	

Response

The response will contain a list of **FitnessActivityReservation** objects that meet the criteria.
The FitnessActivityReservation object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the fitness reservation.	
ReservationNumber	Int64	Sequence number of the fitness reservation.	
ReservationDate	DateTime	Date of the fitness reservation.	
ConfirmationDate	DateTime	Confirmation date of the fitness reservation.	
CancelationDate	DateTime	Cancellation date of the fitness booking.	
Status	FitnessActivityReservationStatus	This is an enumeration for the status of the fitness reservation: • Reserved = 0 • Confirmed = 1 • Cancelled = 2	
NonMemberResevation	Bool	Anonymous (non-member) fitness reservation or not.	
Warranty	Decimal	Warranty that was deducted for the fitness reservation.	
Credits	Int	Sessions that were deducted for the fitness reservation.	
LineAmount	Decimal	Net amount that was paid for the fitness reservation.	
VatAmount	Decimal	VAT amount that was paid for the fitness reservation.	
Amount	Decimal	Total (gross) amount that was paid for the fitness reservation.	
FitnessActivityDay	FitnessActivityDay	Fitness activity day for which the fitness reservation was added.	See 10.4.2 for details concerning the FitnessActivityDay object.
FitnessActivity	FitnessActivity	Fitness activity for which the fitness reservation was added.	See 10.4.2 for details concerning the FitnessActivity object.
PersonId	GUID	ID of the person that made the fitness reservation.	
Person	Person	Person object that made the fitness reservation.	See 8.8.1.1 for details concerning the Person object.
WaitingList	Bool	Indicates if the reservation is placed on the waiting list or not.	
Posted	Bool	Indicates if the reservation is linked to sales (posted)	

FindPersonCards

This method calls up customer cards on the basis of certain search criteria.

Request

The request consists of the service context and a FindPersonCardsCriteria object.

Name	Type	Description	Remarks
Id	GUID	ID of a specific customer card (if you mean to call up only 1 entity).	Nullable (if not applicable).
Number	String	Card number of a specific customer card (if you mean to call up only 1 entity).	Nullable (if not applicable).
PersonId	GUID	ID of a ReCreateX Person for which you want to find customer cards.	Nullable (if not applicable).
Includes	PersonCardIncludes	Determines which data are included in the response and/or which extra records will be present in the returnset.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
CardString	String	The complete output of readed card	

The **PersonCardIncludes** object can be built up using the following criteria:

Name	Type	Description	Remarks
PersonDetails	Bool	Find ReCreateX Person details • YES = the entire Person object is shown	See 8.8.1.1 for details concerning the Person object.

		• NO = only the ID of the Person is shown	
PersonImage	Bool	Include the image of the person within the person details	
IncludeGuestDetails	Bool		
ExcludePeripheralStrings	Bool		
IncludeBlocked	Bool?	Determines if blocked cards should be returned	Default true

Response

The response will contain a list of **PersonCard** objects that meet the criteria.
The PersonCard object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the customer card.	
Description	String	Description of the customer card.	This is in fact the description of the linked article.
Card	Card	The card	Empty in GetCard
PersonId	GUID	ID of the person that made the fitness reservation.	
Person	Person	Person object that made the fitness reservation.	See 8.8.1.1 for details concerning the Person object.

The **Card** object contains the following data:

Name	Type	Description	Remarks
Id	GUID	ID of the card.	
PersonCards	Array of PersonCard	Personcards belonging to this Card	Empty in FindPersonCards
CardNumber	String	Entire card number.	
Number	String	Short card number.	
PeripheralString	PeripheralString	Type of card.	

The **PeripheralString** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the customer card type.	
Code	String	Code of the customer card type.	
Description	String	Description of the customer card type.	e.g. Mifare – Barcode ...)

Fitness – Brief overview

It is now possible to register for fitness/wellness activities on the Webshop.
The system functions as follows:

- Person logs in as a member (using the known user name and password) or as non-member (anonymously).
- Person chooses his customer card and membership he wants to use.
- Person chooses his fitness activity and period for which he wants to register.
- Person pays.

Warning: **In the Fitness module it is necessary to log in first because it is also possible to register anonymously.**
Detailed management of fitness activities is done in ReCreateX.
The Webservice methods that have been created for this purpose are the following:

- FindPersonCards (find the customer cards)
- FindSubscriptions (find membership)
- GetFitnessNonMemberPerson (find non-member person)
- FindFitnessActivities (find fitness activities and days)
- FindFitnessActivityReservations (find fitness reservations)
- ConfirmFitnessActivityReservation (confirm one fitness reservation)
- CancelFitnessActivityReservation (cancel one fitness reservation)

The following extra Webservice methods have been added.

- GetFitnessActivity (find one fitness activity)
- GetFitnessActivityReservation (find one fitness reservation)

GetCard

This method calls up a divisioncard on the basis of certain search criteria.

Request

The request consists of the service context and a FindPersonCardsCriteria object.

Name	Type	Description	Remarks
Id	GUID	ID of a specific card	Nullable (if not applicable).
Includes	CardIncludes	Determines which data are included in the response and/or which extra records will be present in the returnset.	
CardString	String	The complete output of read card read	

The **CardIncludes** object can be built up using the following criteria:

Name	Type	Description	Remarks
PersonDetails	Bool	Find ReCreateX Person details • YES = the entire Person object is shown • NO = only the ID of the Person is shown	See 8.8.1.1 for details concerning the Person object.

Response

The response will contain a **Card** object that meets the criteria.
The Card object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the division card.	
CardNumber	String	The full number of the division card	
Number	String	Number of the card	
Peripheralstring	Peripheralstring	Type of card	
PersonCards	Array of PersonCard	PersonCard objects that belong to this DivisionCard	
Guests	Array of Anonymous	The guest data linked to the card	
Blocked	Bool	Indicates whether the card is blocked or not	

GetFitnessActivity

This method calls up the details of one specific FitnessActivity.

Request

The request consists of the ID of the fitness activity.

Name	Type	Description	Remarks
FitnessActivityId	GUID	ID of the fitness activity.	

Response

The response will contain a **FitnessActivity** object.

Name	Type	Description	Remarks

FitnessActivity	FitnessActivity	FitnessActivity object.	See 0 for details concerning the FitnessActivity object.

This response can also be generated by specifying the ID of the fitness activity and activating all includes using the ***FindFitnessActivities*** *method.*

GetFitnessActivityReservation

This method calls up the details of one specific FitnessActivityReservation.

Request

The request consists of the ID of the fitness activity.

Name	Type	Description	Remarks
FitnessActivityReservationId	GUID	ID of the fitness reservation.	

Response

The response will contain a **FitnessActivityReservation** object.

Name	Type	Description	Remarks
FitnessActivityReservation	FitnessActivityReservation	FitnessActivityReservation object.	See 10.6.2 for details concerning the FitnessActivityReservation object.

This response can also be generated by specifying the ID of the fitness reservation and activating all includes using the ***FindFitnessActivityReservations*** *method.*

GetFitnessNonMemberPerson

This method calls up the **Person** object that is used for non-member or anonymous fitness reservations. By logging in behind the scenes with this person, non-member reservations or anonymous reservations can be made. In case of non-member reservations, no customer card or membership can be specified.

Request

This request contains no specific parameters.

Response

The response contains a **Person** object. This object is used to log in for non-member reservations.
See 8.8.1.1 for details concerning the Person object.

11 Expositions module

CancelOrganisedVisit

This method is used for cancelling an organised visit. Organised visits could have been already paid or not. If an organised visit is already paid, a (fictive) cash or invoice payment method id should be added to the request in order to correctly booking the cancellation.

Request

The CancelOrganisedVisit request consists of the service context and an CancelOrganisedVisitCriteria object. This object looks as follows:

CancelOrganisedVisitCriteria

Name	Type	Description	Remarks
OrganisedVisitId	GUID	The unique ID of the exposition visit.	
ReasonId	GUID	A mandatory reason for the cancellation. Must be of type 'Unknown' or 'Exposition'	
PaymentMethodId	GUID?	If filled in, cancellation will be recorded on this payment method. This is mandatory when the organised visit already has been recorded (posted)	

Response

The response is an object of type CancelOrganisedVisitResponse. An CancelOrganisedVisitResponse contains a property 'CancelOrganisedVisitValidationResult', which is a derived class from ValidationResult

CancelOrganisedVisitValidationResult

Name	Type	Description	Remarks
IsValid	Bool	Bool returning if cancellation succeeded or not	
BrokenRuleName	String	If cancellation did not succeed, or validation has failed, the BrokenRuleName will be filled in with string 'OrganisedVisits_CANCEL_FAILED'	
ReturnAmount	Decimal	The value of the total amount of the organised visit that will be cancelled. Is information only.	

ChangeExpositionPeriodCapacity

This method is used for calling up exposition visits.

Request

The ChangeExpositionPeriodCapacity request consists of the service context and a ChangeExpositionPeriodCapacityCriteria object. This object looks as follows:

ChangeExpositionPeriodCapacityCriteria

Name	Type	Description	Remarks
ExpositionPeriodId	GUID	The unique ID of the exposition period.	
Capacity	int	The capacity will be changed to this value	

Example

```
{
  "Context": {
        "ShopId": "00000000-0000-0000-0000-000000000641",
        "Password": "myVerySecretPassword",
        "Language": "nl"
  },
  "Criteria":{
        "ExpositionPeriodId": "a0906afd-b0ee-e611-8160-78acc098f184",
        "Capacity": 100
  }
}
```

Response

The response contains a boolean **HasSucceeded** which indicates if the change has succeeded.

Example

```
{
    "HasSucceeded": true
}
```

ChangeExpositionPeriodState

This method is used for calling up exposition visits.

Request

The ChangeExpositionPeriodState request consists of the service context and a ChangeExpositionPeriodStateCriteria object. This object looks as follows:

ChangeExpositionPeriodStateCriteria

Name	Type	Description	Remarks
ExpositionPeriodId	GUID	The unique ID of the exposition period.	
State	int	The state in which the exposition period must be changed	0 = Start, 1 = Stop

Example

```
{
  "Context": {
        "ShopId": "00000000-0000-0000-0000-000000000641",
        "Password": "myVerySecretPassword"
  },
  "Criteria":{
        "ExpositionPeriodId": "02936AFD-B0EE-E611-8160-78ACC098F184",
        "State": 1
  }
}
```

Response

The response contains a boolean **HasSucceeded** which indicates if the change has succeeded.

Example

```
{
    "HasSucceeded": true
}
```

CheckOrganisedVisitAlreadyRegistered

This API call is used to check if a certain user is already registered on a timeslot of a certain exhibition.

Note: Will always return false if setting in ReCreateX is disabled

Request

Name	Type	Description
Context	ServiceContext	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.

Criteria	CheckOrganisedVisitAlreadyRegisteredCriteria	The criteria used for CheckOrganisedVisitAlreadyRegistered

CheckOrganisedVisitAlreadyRegisteredCriteria

Name	Type	Description
ExpositionId	GUID	The unique identifier of the exposition
ExpositionPeriodId	GUID	The unique identifier of the exposition period (timeslot)
CustomerId	GUID	The unique identifier of the user

Response

The response is of type CheckOrganisedVisitAlreadyRegisteredResponse

Name	Type
Result	CheckOrganisedVisitAlreadyRegisteredResult

CheckOrganisedVisitAlreadyRegisteredResult

Name	Type	Description
AlreadyRegistered	bool	Indicates whether the user is already registered for the timeslot on the exhibition

ExpositionS – Brief overview

The Webservice provides the possibility to order exposition reservations through your web site.
The system functions as follows:

- Person selects the exposition he/she wants to book tickets for.
- Person selects the date and time for the exposition he/she wants to book.
- Person selects the desired tickets (ex: 1 adult / 2 children).
- Optionally the person selects extra optional articles or tickets.

Detailed management of all exposition reservations is done in ReCreateX.
Following methods are available within this module:

- ListExpositionTypes
- FindExpositions
- ListExpositionPeriods
- ListAudiences
- FindOrganisedVisits

FindExpositionOverviewByDay

Calling up this Webservice operation will visualize the desired exposition periods on the basis of the criteria.

Request

The request consists of the service context and an ExpositionOverviewByDayCriteria object.

ExpositionOverviewByDayCriteria

Name	Type	Description	Remarks
ExpositionId	Guid	The unique identifier of the exhibition	Mandatory
StartTime	DateTime	The date and time "from" you want to receive exhibition periods for	Mandatory
EndTime	DateTime	The date and time "until" you want to receive exhibition periods for	Mandatory
Includes	ExpositionOverviewByDayIncludes	Determines which additional data is included in the response Determines which extra records will be present in the result set	/

ExpositionOverviewByDayIncludes

Name	Type	Description	Remarks
Pricing	Boolean	The result will also include the pricing details	/

Response

A list of exposition overview items. An exposition overview item consists of the following parts:

ExpositionOverviewByDayIncludes

Name	Type	Description	Remarks
ExpositionId	Guid	The unique identifier of the exhibition	/
ExpositionCode	String	The code of the exhibition	/
ExpositionName	String	The name of the exhibition	/
Periods	IList<ExpositionOverviewItemPeriod>	The list of periods within the exhibition	/
Prices	IList<ExpositionOverviewItemPrice>	The list of prices for the exhibition	/

ExpositionOverviewItemPeriod

Name	Type	Description	Remarks
From	DateTime	The start date and time for the exhibition period	/
Until	DateTime	The end date and time for the exhibition period	/
Occupancy	Occupancy	The occupancy and availability details for the exhibition period	/

ExpositionOverviewItemPrice

Name	Type	Description	Remarks
PriceGroupId	Guid	The unique identifier of the linked price group or article	/
PriceGroupName	String	The name of the linked price group or article	/
Price	Decimal	The default price for the price group or article	/
MinPresaleDiscount	Decimal	The minimum discount available for the price group or article	/
MaxPresaleDiscount	Decimal	The maximum discount available for the price group or article	/

⚠ *NOTE: The ExpositionOverviewItemPrice object will only contain the default pricing and will not contain any detailed pricing information, such as additional costs or graduated prices.*

FindExpositions

Calling up this Webservice operation will visualise the desired expositions on the basis of the linked parameters.

Request

The request consists of the service context and an ExpositionSearchCriteria object.

ExpositionSearchCriteria

Name	Type	Description	Remarks
ExpositionId	GUID	If filled in: The ID of the specific exposition one wants to find.	
NamePattern	String	If filled in: Only expositions of which the name contains the name pattern will be shown.	
From	DateTime	If filled in: Only expositions that start after the given date will be shown.	
Until	DateTime	If filled in: Only expositions that start before the given date will be shown.	
ExpositionTypeId	GUID	If filled in: Only expositions that belong to the ExpositionType with the given ID will be shown.	

AudienceId	GUID	If filled in: Only expositions for the given target group will be shown.	
Includes	ExpositionIncludes	This object will determine which characteristics of the expositions will be called up(see 11.3.1.2: ExpositionIncludes).	
Paging	PagingCriteria	See 4.2: Paging Criteria	
TranslationRequired	Boolean	If true, only the expositions that have a translation will be included.	
LocationAddressId	GUID	If filled in: Only expositions that are organised in the given location.	
AudienceIds	List<GUID>	If filled in: Only expositions for the given target group will be shown.	
SubCategoryIds	List<GUID>	If filled in: Only expositions for the given subcategories will be shown.	
ExpositionTypeIds	List<GUID>	If filled in: Only expositions that belong to the given ExpositionTypes will be included.	
ExcludeDonation	Bool?	If filled in: Use this persons pricegroup for price calculation	
SelectionType	SelectionType?	The type of selection you wish • Normal • Vouchers • Both	
TraderId	Guid?	If filled in: Only load expositions which are linked to the trader (company)	

ExpositionIncludes

Name	Type	Description	Remarks
ImageUrl	Boolean	Include Image URL of the Exposition.	
Image	Boolean	Include Image of the Exposition.	
Audiences	Boolean	Include all linked audiences.	
Pricing	Boolean	Include the pricing details	
Subcategories	Boolean	Include the linked subcategories	
Location	Boolean	Include the location	
Allpricing	Boolean	Include all pricing details (depending on minimum quantity)	
Articles	Boolean	Include the extra articles	
AllArticles	Boolean	Include all pricing details (depending on minimum quantity) of the articles	
ActivePeriods	Boolean	Include the active periods for the articles	
AskNamesConfiguration	Boolean	Include the configuration for asking names	
IncludePricesOnlyAllowedInCombiExpositions	Boolean	Include prices that are only allowed in CombiExpositions	

Response

A list of expositions. An exposition consists of the following parts:

Exposition

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the exposition.	
Code	String	The code of the exposition.	
Name	String	The name of the exposition.	
Description	String	The general description of the exposition.	
Image	Picture	The image of the exposition.	
From	DateTime	The start date of the exposition as configured on the exposition periods	Equals the first date of all underlying periods
Until	DateTime	The end date of the exposition as configured on the exposition periods	Equals the last date of all underlying periods

Location	String	The location where the exposition takes place.	
ExpositionType	ExpositionType	The type to which the exposition belongs.	
Periods	List<ExpositionPeriod>	A list of linked ExpositionPeriods.	Not in use. In order to call up the periods of a specific exposition, the appropriate Webservice method has to be used.
Prices	List<ExpositionPrices>	A list of possible prices for the exposition.	
Vat	VAT	The VAT for the exposition.	
Audiences	List<Audiences>	A list of linked target groups.	
DivisionId	GUID	The division to which the exposition belongs.	
SubCategories	Collection<ExpositionCategory>	All subcategories the exposition is linked to.	
AddressLocation	AddressLocation	The location the exposition will take place.	
ImageUri	String	The Imaging service URL to the image.	
MaxVisitorsPerGroup	Int	Maximum number of visitors allowed per group.	
Info1	String	Extra information field 1	
Info2	String	Extra information field 2	
Info3	String	Extra information field 3	
Info4	String	Extra information field 4	
Info5	String	Extra information field 5	
GiftAidEnabled	Bool	Is the gift aid functionality enabled for this exposition?	
GenLoyaltyActive	Bool	Is the general loyalty activated for this exposition?	
ControlType	Int	0 = visitors per period 1 = groups per period	
MaxVisitorsPerPeriodWeb	Int	The maximum number of visitors per period for web	
Articles	Collection<ExpositionArticle>	All extra articles which are automatically sold with the exposition	
OptionalArticles	Collection<ExpositionArticleOptional>	All optional articles which can be sold with the exposition	
AskNames	Bool	Is it possible to manage the names of the visitors?	
MaxTicketsPerWebSale	int	Maximum tickets in one web sale 0 = infinite	
AllowOverbooking	Bool	Is overbooking allowed on the exposition?	
AskNamesConfiguration	AskNamesConfiguration	The configuration for asking names (which fields are visible/mandatory)	Will only be filled when the include is set to TRUE in the request and when the AskNames property is true for the exposition
StartDate	DateTime	The start date of the exposition	
EndDate	DateTime	The end date of the exposition	
ResellerCapacity	int	The reseller capacity is a capacity that is managed by an external party.	
ExpositionTimePeriodCategories	ExpositionTimePeriodCategory	The different timepriod categories set up for that exposition	
FinalRegistrationPerCountryGroups	Collection<FinalRegistrationPerCountryGroup>	All info for final registration	
MinimumAge	int	The minimum age to buy a ticket for this	

		exposition	
ItemGroupsSelected	Bool	Are item groups selected for this exposition	

ExpositionPeriod

An exposition period is the time on which an exposition takes place. An exposition can obviously take place on several times. Hence, an exposition contains a list of ExpositionPeriods.
An ExpositionPeriod has the following properties:

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the ExpositionPeriod.	
From	Date Time	Start time of the ExpositionPeriod.	
Until	Date Time	End time of the ExpositionPeriod.	
FinalSubscriptionDate	Date Time	Final registration date for the exposition.	
Occupancy	Occupancy	The occupancy of this ExpositionPeriod.	Warning, the remaining number can be less than the maximal number of visitors minus the visitors registered!
ExpositionId	GUID	ID of the exposition to which the exposition period belongs.	
IsExcluded	bool	Determines if the period is excluded or not.	The excluded periods will only be returned when the **IncludeExcludedDays** parameter is set to TRUE in the **ExpositionPeriodSearchCriteria** object.
FinalSubscriptionDateBo	Date Time?	Final registration date for the exposition in back office.	
Comment	string	Information / Comment about the period	
ResellerCapacity	int	The reseller capacity is a capacity that is managed by an external party.	

Occupancy

The occupancy object has the following properties:

Name	Type	Description	Remarks
Current	Int	The current number of reservations already made.	
Maximum	Int	The maximum number of reservations that can be made.	
Remaining	Int	The number of remaining reservations that can be made.	
ControlType	Int	0 = visitors per period 1 = groups per period	
MaxVisitorsPerGroup	Int	The maximum number of visitors per group	Only useful when ControlType = 1

ExpositionPrice

The exposition period can contain several prices. Each of these prices has a specific target group. Such a price looks as follows:ExpositionArticle

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the price specified in ReCreateX.	
Amount	Decimal	The actual price.	
Group	PriceGroup	The price group to which the price applies.	
CalculationType	CalculationType	Enumeration of calculation method. The possible values are: • PriceTimesHours = 0 • FixedPrice = 1 • PriceTimesQuantity = 2	The price is either fixed, either dependent on the number of persons, the number of hours or the number of hours and persons. • Price x Hour • Fixed • Price x Number

		• PriceTimesQuantityTimesHours = 3	• Price x Number x Hour
MinimumQuantity	Int	The minimum quantity the price is available from	Ex: • € 5,00 for 0 to 9 • € 4,00 for 10+
ExpositionAmount	Decimal	The price for the exposition ticket	Amount = ExpositionAmount + ArticleAmount
ArticleAmount	Decimal	The added price for additional article sales	
ActivePeriods	List<Period>	The active periods in which the exposition ticket can be sold/bought	
MinSaleQuantity	Int	The minimum amount of tickets which need to be sold/bought for the exposition ticket	
MaxSaleQuantity	Int	The maximum amount of tickets which need to be sold/bought for the exposition ticket	
SelectionType	SelectionType	The type of selection • Normal • Vouchers • Both	
HasAllowedSubscription	Bool	Does this item have a required membership?	
ArticleInfo1	string	Info1 field of the article the price is based upon	old prices based on pricegroups containes empty string
ArticleInfo2	string	Info2 field of the article the price is based upon	old prices based on pricegroups containes empty string
ArticleInfo3	string	Info3 field of the article the price is based upon	old prices based on pricegroups containes empty string
ArticleInfo4	string	Info4 field of the article the price is based upon	old prices based on pricegroups containes empty string
ArticleInfo5	string	Info5 field of the article the price is based upon	old prices based on pricegroups containes empty string

PriceGroup

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the price group specified in ReCreateX.	
Code	String	The code of the price group.	
Name	String	The description of the price group.	
Type	PriceGroupType	Enumeration of the price group type. • General = 0 • PlaceReservations = 1 • Ticketing = 2 • Entries = 3 • Sales = 4 • Rental = 5 • Wellness = 6	• General (= valid everywhere) • Facility bookings • Ticketing bookings • Registrations • Sales (& expositions) • Rental • Fitness/wellness

ExpositionArticle

The exposition period can contain several extra articles which are automatically added to the exposition visit.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the exposition article specified in ReCreateX.	
ArticleId	GUID	The unique identifier of the article	
Description	String	Article description	
Amount	Decimal	Article price	
CalculationType	CalculationType	Enumeration of calculation method. The possible values are: • PriceTimesHours = 0 • FixedPrice = 1	The price is either fixed, either dependent on the number of persons, the number of hours or the number of hours and persons. • Price x Hour • Fixed

		- PriceTimesQuantity = 2 - PriceTimesQuantityTimesHours = 3	- Price x Number - Price x Number x Hour
Minimum Quantity	Int	The minimum quantity the price is available from	Ex: - € 5,00 for 0 to 9 - € 4,00 for 10+
UseLinkedPrices	Bool	Defines whether custom prices are used per price group/article	
LinkedPrices	Colleciton <GUID>	Lists all ExpositionPrice identifiers to which the article is applicable as defined in RCX	
ActivePeriods	List<Period>	The active periods for the article	
Maximum Quantity	int	The maximum quantity of which you can buy per booking	

ExpositionArticleOptional

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the exposition article specified in ReCreateX.	
ArticleId	GUID	The unique identifier of the article	
Description	String	Article description	
ShortName	String	Article shortname	
MinPrice	Decimal	The minimum price for this optional article	
MinQuantity	Int	The minimum quantity the price is available for	Ex: - € 5,00 for 0 to 9 - € 4,00 for 10+
MaxQuantity	int	The maximum quantity the price is available for	
MaxQuantityPerTimeSlot	int	The maximum quantity per time slot	
MaxQuantityPerTicket	int	The maximum quantity per ticket	
ActivePeriods	List<Period>	The active periods for the article	
UnallowedItemSales	IList<Guid>	The optional item may not be sold if all the access ticket ids can be found within this array	
AdditionalInformation	String	The extra information in a chosen language	

AskNamesConfiguration

The AskNamesConfiguration contains details about which fields are visible and mandatory when the AskNames property is active for the exposition.

Name	Type	Description	Remarks
AskFirstName	Bool	Defines whether the first name field is visible	
MandatoryFirstName	Bool	Defines whether the first name field is mandatory	
AskName	Bool	Defines whether the last name field is visible	
MandatoryName	Bool	Defines whether the last name field is mandatory	

AskBirthDate	Bool	Defines whether the birth date field is visible	
MandatoryBirthDate	Bool	Defines whether the birth date field is mandatory	

ExpositionTimePeriodCategory

Name	Type	Description	Remarks
Description	string	The name of the ExpositionTimePeriodCategory	
From	DateTime	The start time of that ExpositionTimePeriodCategory	
Until	DateTime	The end time of that ExpositionTimePeriodCategory	
Sequence	Int	The number of that ExpositionTimePeriodCategory. For sorting/ordering purposes.	

FinalRegistrationPerCountryGroup

Name	Type	Description	Remark
ID	Guid	The unique identifier of the FinalRegistrationPerCountryGroup	
ShortName	string	The short name	
Name	string	The description	
ExpositionId	Guid	The unique identifier of the exposition	
CountryGroupId	Guid	The unique identifier of the country group	
RegistrationClosurePeriodInMinutes	int	The amount of minutes you need between the exposition Start DateTime and moment of reservation	
Countries	List	The list of countries in the country group	

FindExpositionPrices

This call can be used to retrieve ExpositionPrices specifically, without all other possible unwanted information that is usually retrieved in FindExpositions.

Request

The request consists of the service context and an FindExpositionPricesSearchCriteria object.

FindExpositionPricesSearchCriteria

Name	Type	Description	Remarks
ExpositionId	GUID	If filled in: The ID of the specific exposition one wants to find.	
ExcludeDonation	Bool?	Excludes donations for price calculation if set to true	
PersonId	Guid?	Will calculate the price for this specific Person if filled in	
ExpositionPeriodDate	DateTime?	Will be used for price calculation if filled in. Also used for Active Periods	
SelectionType	SelectionType?	The type of selection you wish • Normal • Vouchers • Both	
IsRebooking	Bool	Set to true if you want to retrieve exposition prices for a rebooking	
IncludeActivePeriods	Bool	Will return the active periods if set to true	
IncludeOnlyAllowedInCombiExpositions	Bool	Will return prices that are only allowed in CombiExpositions	

Response

A list of expositions. An exposition consists of the following parts:

ExpositionPrice

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the Exposition Price	
Amount	Decimal	How much should be paid for the Exposition Price	
Group	PriceGroup	The pricegroup of the Exposition Price	
CalculationType	CalculationType	How the Exposition Price is calculated	• PriceTimesHours = 0 • FixedPrice = 1 • PriceTimesQuantity = 2 • PriceTimesQuantityTimesHours = 3
MinimumQuantity	Int	The minimum allowed quantity of the Exposition Price	
ExpositionAmount	Decimal		
ArticleAmount	Decimal		
ActivePeriods	List<Period>		
MinSaleQuantity	Int		
MaxSaleQuantity	Int		
AdditionalInformation	String		
ReservationCost	Decimal		
SelectionType	SelectionType		• Normal = 0 • Vouchers = 1 • Both = 2
AskNames	Bool		
MandatoryBuyMinQtyWeb	Bool		
SortOrder	Int		
UitpasChanceTariffMandatory	Bool		
HasAllowedSubscription	Bool	has the item a required membership?	
ArticleInfo1	string	Info1 field of the article the price is based upon	old prices based on pricegroups containes empty string
ArticleInfo2	string	Info2 field of the article the price is based upon	old prices based on pricegroups containes empty string
ArticleInfo3	string	Info3 field of the article the price is based upon	old prices based on pricegroups containes empty string
ArticleInfo4	string	Info4 field of the article the price is based upon	old prices based on pricegroups containes empty string
ArticleInfo5	string	Info5 field of the article the price is based upon	old prices based on pricegroups containes empty string

PriceGroup

Name	Type	Description	Remarks
Id	Guid		
Code	String		
Name	String		
Type	PriceGroupType		• General • PlaceReservations • Ticketing • Entries • Sales • Rental • Wellness

GenLoyaltyProfileCheck	Bool		
GenLoyaltyProfile	String		
GenLoyaltyActive			
ActivateSubsidyRegulation	Bool		

Period

Name	Type	Description	Remarks
From	DateTime		
Until	DateTime		

FindOrganisedVisits

This method is used for calling up exposition visits.

Request

The FindOrganisedVisits request consists of the service context and an OrganisedVisitSearchCriteria object. This object looks as follows:

FindOrganisedVisitsRequest

Name	Type	Description	Remarks
OrganisedVisitId	GUID?	The unique ID of the exposition visit.	
From	DateTime?	If filled in, only the exposition visits that start after this date will be shown.	
Until	DateTime?	If filled in, only the exposition visits that end before this date will be shown.	
PersonId	GUID?	ID of the person of whom you want to find the exposition visits.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	OrganisedVisitIncludes	This object will determine which characteristics of the exposition visits will be called up.	
`ModifiedSince`	DateTime	Filters on organised visits that were modified or that have the organisedVisitPriceGroup or the organisedVisitSales changed after this datetime.	
OrderNumber	String	If filled in, only the exposition visits with this Order Number will be shown	

OrganisedVisitIncludes

Name	Type	Description	Remarks
Cancelled	Bool	Also call up cancelled visits?	
PersonDetails	Bool	Also call up detailed data of the person?	
PeriodReservations	Bool	Also call up periodical detailed data of the exposition visit?	
Articles	Bool	Also call up linked articles of the exposition visit?	
ContactDetails	Bool	Also call up detailed data of the person's contact?	
CombiOrganisedVisits	Bool	Also call up CombiOrganisedVisits	

Response

The response is a list of OrganisedVisit objects. An OrganisedVisit object looks as follows:

OrganisedVisit

Name	Type	Description	Remarks

Id	GUID	The unique identifier of the exposition visit.	
No	Long	Number of the exposition visit.	
StartDate	DateTime	Start date and time of the exposition visit.	
EndDate	DateTime	End date and time of the exposition visit.	
Comment	String	Extra remarks concerning the exposition visit.	
PeriodReservationAmount	Decimal	Total amount of all underlying period reservations.	
ArticlesAmount	Decimal	Total amount of all linked articles.	
CancelCostAmount	Decimal	Cancellation cost in case the visit was cancelled.	
CancelReturnAmount	Decimal	Refund in case the visit was cancelled.	
TotalAmount	Decimal	Total amount of the exposition visit.	
Cancelled	Bool	Has the exposition visit been cancelled?	
Closed	Bool	Has the exposition visit been closed?	
PersonId	GUID	Unique ID of the linked person.	
Person	Person	Detailed data of the linked person.	See 8.8.1.1: Person
PeriodReservations	List<OrganisedVisitPeriodReservations>	Overview of all linked period bookings. 2 implementations possible: • OrganisedVisitPriceGroupPeriodReservation • OrganisedVisitArticlePeriodReservation	
Articles	List<OrganisedVisitArticle>	Overview of all linked articles.	
GenLoyaltyCardDiscount	Decimal	The general loyalty discount of the exposition visit.	(From RCX 5.4.0)
TotalAmountWithoutDiscount	Decimal	The total amount without the given discount amount.	
ReservationCost	Decimal	The amount of reservation costs	
PostedAdvancement	Decimal	The amount of advancement that is booked	
CouponDiscount	Decimal	The discount amount for the used coupon(s)	
PostedAmount	Decimal	The total amount that is booked	
Balance	Decimal	The remaining open amount (that still needs to be booked)	
Posted	Bool	Is the visit booked?	
Contact	Contact	The customer's contact	
OrderNumber	String	Ordernumber	
PersonalizedMessage	String	Personalized message	
PeriodReservationLineAmount	Decimal	The line amount ot the period reservation	
PeriodReservationVatAmount	Decimal	The vat amount of the period reservation	
ArticlesLineAmount	Decimal	The sum of all article line amounts	
ArticlesVatAmount	Decimal	The sum of all article VAT amounts	
SalesInfos	List<SaleInfo>	The list of sales information	
Coupon	String	The used coupon code	
LanguageID	Guid?	The Language of the OrganisedVisit	
HasEtickets	Bool	Indicates whether the OrganisedVisit has any ETickets	
NumberOfEtickets	Int	Indicates how many ETickets the OrganisedVisit has	
MaxCancelWebDays	Int	Indicates how many days before the OrganisedVisit's date it can be cancelled	
MaxCancelWebHours	Int	Indicates how many hours before the OrganisedVisit's date it can be cancelled	

OrganisedVisitPriceGroupPeriodReservation

Name	Type	Description	Remarks
Id	GUID	The unique identifier.	
ExpositionId	GUID	The unique identifier of the linked exposition.	
ExpositionPeriodCode	GUID	The code of the linked exposition period.	
ExpositionPeriodName	GUID	The name of the linked exposition period.	
ExpositionPeriodDescription	GUID	The description of the linked exposition period.	
ExpositionPeriodId	GUID	The unique identifier of the linked exposition period.	
ExpositionPeriodFrom	DateTime	Start date of the linked exposition period.	
ExpositionPeriodUntil	DateTime	End date of the linked exposition period.	
Quantity	Int	Quantity (in this case the number of persons).	
UnitPrice	Decimal	The unit price.	
Amount	Decimal	Total amount (= Quantity x UnitPrice)	
Visitors	List<OrganisedVisitor>	The details of the visitors	
AskNames	Bool	Is it possible to manage the names of the visitors?	
PriceGroup	PriceGroup	Price group	See 11.3.2.5: PriceGroup
LineAmount	Decimal	The line amount of the period reservation	
VatAmount	Decimal	The VAT amount of the period reservation	
VatPercentage	Decimal	The VAT percentage of the period reservation	

OrganisedVisitArticlePeriodReservation

Name	Type	Description	Remarks
Id	GUID	The unique identifier.	
ExpositionId	GUID	The unique identifier of the linked exposition.	
ExpositionPeriodCode	GUID	The code of the linked exposition period.	
ExpositionPeriodName	GUID	The name of the linked exposition period.	
ExpositionPeriodDescription	GUID	The description of the linked exposition period.	
ExpositionPeriodId	GUID	The unique identifier of the linked exposition period.	
ExpositionPeriodFrom	DateTime	Start date of the linked exposition period.	
ExpositionPeriodUntil	DateTime	End date of the linked exposition period.	
Quantity	Int	Quantity (in this case the number of persons).	
UnitPrice	Decimal	Unit price.	
Amount	Decimal	Total amount.	
ArticleId	GUID	The unique identifier of the linked article.	
ArticleCode	String	Code of the linked article.	
ArticleName	String	Name of the linked article.	
ArticleDescription	String	Description of the linked article.	
OrganisedVisitArticleName	Type	Description.	Remarks.
Id	GUID	The unique identifier.	
ArticleId	GUID	The unique identifier of the linked article.	
ArticleCode	String	Code of the linked article.	
ArticleName	String	Name of the linked article.	
ArticleDescription	String	Description of the linked article.	
PriceGroup	PriceGroup	Price group.	See 10.3.2.4: PriceGroup

Quantity	Int	Quantity (in this case the number of articles).	
UnitPrice	Decimal	The unit price per article.	
Amount	Decimal	Total amount (= Quantity x UnitPrice).	
Visitors	List<OrganisedVisitor>	The details of the visitors	
AskNames	Bool	Is it possible to manage the names of the visitors?	
LineAmount	Decimal	The line amount of the period reservation	
VatAmount	Decimal	The VAT amount of the period reservation	
VatPercentage	Decimal	The VAT percentage of the period reservation	
UnitPriceExc	Decimal	The unit price without VAT per article	

OrganisedVisitor

Name	Type	Description	Remarks
Id	GUID	The unique identifier.	
Name	String	The name of the visitor	
FirstName	String	The first name of the visitor	
BirthDate	DateTime	The birth date of the visitor	
Email	String	The e-mail of the visitor	
Phone	String	The phone of the visitor	
Street	String	The street of the visitor	
HouseNumber	String	The house number of the visitor	
Zipcode	String	The zipcode of the visitor	
City	String	The city of the visitor	
Country	String	The country of the visitor	

OrganisedVisitArticle

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the linked article	
ArticleCode	String	Code of the linked article	
ArticleName	String	Name of the linked article	
ArticleDescription	String	Description of the linked article	
PriceGroup	PriceGroup	Price group	See 11.3.2.5: PriceGroup
Quantity	Int	Quantity (in this case the number of articles)	
UnitPrice	Decimal	The unit price of the linked article	
Amount	Decimal	The total amount (=Quantity X UnitPrice)	
LineAmount	Decimal	The line amount of the article	
VatAmount	Decimal	The VAT amount of the article	
VatPercentage	Decimal	The VAT percentage of the article	
UnitPriceExc	Decimal	The unit price without VAT of the linked article	

SaleInfo

Name	Type	Description
Id	GUID	The unique identifier of the sale
SalesNo	long?	The sales number of the sale

SalesDate	DateTime	The date of the sale
InvoiceNumber	long	The invoice number of the sale
InvoiceDate	DateTime	The invoice date of the sale
Guest	Anonymous	The information about the anonymous customer

Anonymous

Name	Type	Description
Name	string	The last name of the anonymous customer
FirstName	string	The first name of the anonymous customer
Street1	string	The first street of the anonymous customer
Street2	string	The second street of the anonymous customer
Number	string	The house number of the anonymous customer
Box	string	The house box of the anonymous customer
Home	string	The city of the anonymous customer
Country	string	The country of the anonymous customer
Email	string	The e-mail of the anonymous customer
Newsletter	bool	Indicates if the anonymous customer wants to subscribe to the newsletter
ZipCode	string	The zipcode of the anonymous customer
Telephone	string	The telephone of the anonymous customer

OrganisedVisitSkill

Name	Type	Description
Id	Guid	The id of the SkillSubCategory
OrganisedVisitId	Guid	The id of the organised visit
SkillDescription	Guid	The description of the SkillSubCategory

GetExpositionOptionalArticlesCapacity

Calling up this Webservice operation will calculate the remaining capacity for an optional article.

Request

The request consists of the service context and an GetExpositionOptionalArticlesCapacityCriteriaobject.

GetExpositionOptionalArticlesCapacityCriteria

Name	Type	Description	Remarks
ExpositionPeriodId	Guid	The ID of the specific exposition period (timeslot) you want to calculate the capacity for	
OptionalArticleIds	IList<Guid>	The IDS of the specific optional articles you want to calculate the capacity for	

Sample

```
{
  "Criteria": {
    "ExpositionPeriodId": "f41f765a-d5c2-e711-937d-78acc098f184",
    "OptionalArticleIds": [
       "ac0345b2-d5c2-e711-937d-78acc098f184"
    ]
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000640",
    "Password": "d3m0"
  }
}
```

Response

A list of ExpositionOptionalArticleCapacity. An ExpositionOptionalArticleCapacity consists of the following parts:

Exposition

Name	Type	Description	Remarks
OptionalArticleId	Guid	The ID of the optional article	
RemainingCapacity	int	The remaining capacity for this optional article and the timeslot in the request	

Sample

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
GetExpositionOptionalArticlesCapacityResponse, ReCreateX.WebShop.
WebServices.Contracts",
    "result": [
        {
            "$type": "ReCreateX.WebShop.Model.
ExpositionOptionalArticleCapacity, ReCreateX.WebShop.Model",
            "optionalArticleId": "ac0345b2-d5c2-e711-937d-78acc098f184",
            "remainingCapacity": 15
        }
    ]
}
```

GetExpositionSales

Calling up this Webservice operation will visualise sale details about a specific exposition.

Request

The request consists of the service context and an GetExpositionSalesCriteria object.

GetExpositionSalesCriteria

Name	Type	Description	Remarks
ExpositionId	GUID	The ID of the specific exposition	

Example

```
{
  "Criteria": {
          "ExpositionId": "40C19DDB-B0EE-E611-8160-78ACC098F184"
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000641",
    "Password": "anc"
  }
}
```

Response

Object of type **ExpositionSalesSummary** is returned in the response

ExpositionSalesSummary

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the exposition.	
TotalSold	int	The total tickets sold for this exposition	
Details	Array of ExpositionSalesDetail	Sale details per price	
ExpositionPeriodSalesSummaries	Array of ExpositionPeriodSalesSummary	Sale summary per exposition	

ExpositionPeriodSalesSummary

Name	Type	Description	Remarks
ExpositionPeriodId	Guid	The id of the exposition period	
Details	Array of ExpositionSalesDetail	Sale details per price	
TotalSold	int	The total tickets sold for this exposition period	

ExpositionSalesDetail

Name	Type	Description	Remarks
PriceId	Guid	The id of the price(group)	
TicketSold	int	Number of sold tickets	

Example

```
{
    "Summary": {
        "Id": "40c19ddb-b0ee-e611-8160-78acc098f184",
        "Details": [
            {
                "PriceId": "4255a368-4f22-4322-a29c-011abbd26839",
                "TicketsSold": 2
            },
            {
                "PriceId": "7ebf1681-ed42-4002-ba6e-49f276261ac7",
                "TicketsSold": 5
            }
        ],
        "TotalSold": 7,
        "ExpositionPeriodSalesSummaries": [
            {
                "ExpositionPeriodId": "a0906afd-b0ee-e611-8160-
78acc098f184",
                "Details": [
                    {
                        "PriceId": "4255a368-4f22-4322-a29c-
011abbd26839",
                        "TicketsSold": 2
                    }
                ],
                "TotalSold": 2
            },
            {
                "ExpositionPeriodId": "a4906afd-b0ee-e611-8160-
78acc098f184",
                "Details": [
                    {
                        "PriceId": "7ebf1681-ed42-4002-ba6e-
49f276261ac7",
                        "TicketsSold": 5
                    }
                ],
                                "TotalSold": 5
                        }
                ]
        }
}
```

ListAudiences

This method will generate a list of the target audiences as defined in ReCreateX.

Request

The request only consists of the service context.

Response

The response is a list of Audiences. The Audience object looks as follows:

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the audience.	
Code	String	The code of the audience.	

Name	String	The name of the audience.	
Description	String	The description of the audience.	

ListExpositionPeriods

Each exposition has a number of exposition periods. Since this number can be rather big, we have split up the calling up of exposition periods into a separate Webservice operation. What an ExpositionPeriod looks like, has been discussed in 11.3.2.2: ExpositionPeriod.

Request

The request consists of the service context and an ExpositionPeriodSearchCriteria object.

ExpositionPeriodSearchCriteria

As mentioned above, an exposition can contain a large number of exposition periods. Therefor, 2 parameters are provided on an ExpositionPeriodSearchCriteria.

Name	Type	Description	Remarks
ExpositionId	GUID	The unique identifier of the exposition of which the periods are called up.	
From	DateTime	If filled in, only the exposition periods that start after this date will be shown.	
Until	DateTime	If filled in, only the exposition periods that end before this date will be shown.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
IgnoreAvailability	Bool?	If true, the availability (Occupancy) will not be filled.	Usually used for performance reasons
LastUpdateFrom	DateTime?	The exposition period must be modified after or equal the given date	
LastUpdateUntil	DateTime?	The exposition period must be modified before or equal the given date	
ExpositionPeriodId	GUID	The unique identifier of the exposition period.	
IncludeExcludedDays	bool?	If filled in, the excluded days will be included in the response.	
IncludeBlocked	bool?	If filled in, the blocked periods will be included in the response.	

Response

This returns a list of the ExpositionPeriods (see 11.3.2.2: ExpositionPeriod) that meet the request above.

ListExpositionTypes

Calling up this Webservice will visualise the desired ExpositionTypes on the basis of the linked parameters.

Request

This only contains a service context.

Response

A list of ExpositionTypes. An ExpositionType consists of the following fields.

ExpositionType

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the exposition type.	
Code	String	The code of the exposition type.	
Description	String	The description of the exposition type.	
Name	String	The name of the exposition type.	

CalculateCombiExpositionReservation

This endpoint will calculate the prices of your combi exposition.

Request

The request consists of the service context and a CalculateCombiExpositionReservationCriteria object.

CalculateCombiExpositionReservationCriteria

Name	Type	Description
CombiExpositionId	Guid	The ID of a specific CombiExposition
PersonId	Guid	Id of the person buying the tickets
SelectedCombiExpositionTickets	List<SelectedCombiExpositionTicket>	The tickets u selected to buy
ExpositionPeriodIds	List<Guid>	The periods u selected for your expositions

SelectedCombiExpositionTicket

Name	Type	Description
Id	Guid	The id of the combi exposition ticket
Quantity	int	The quantity of the selected ticket
Price	decimal	The price of the ticket. in the request this is set at 0
Description	string	The description of the selected ticket
ExtraArticles	CombiExpositionTicketExtraItem	The list of extra items attached to this ticket

CombiExpositionTicketExtraItem

Name	Type	Description
Id	Guid	The id of the combi exposition ticket extra article
ItemId	Guid	The id of the item the extra article is based on
Price	decimal	The price of the article. in the request this is set at 0
Description	string	The description of the selected ticket

Response

The response is very similar to the request. But with the prices filled in and a total price added to the result.

Name	Type	Description
SelectedCombiExpositionTickets	List<SelectedCombiExpositionTicket>	A list of all the selected combi tickets with the prices filled in.
TotalPrice	decimal	The total price of all the selected tickets and the extra articles.

FindCombiExpositions

This endpoint will return a list of CombiExpositions based on the passed search criteria.

Request

The request consists of the service context and an FindCombiExpositionsCriteria object.

FindCombiExpositionsCriteria

Name	Type	Description
Id	Guid?	The ID of a specific CombiExposition
NamePattern	string	A part of the description (or translation) of a CombiExposition
From	DateTime?	Retrieves CombiExpositions that take place after this

Until	DateTime?	Retrieves CombiExpositions that take place before this
Includes	FindCombiExpositionsIncludes	To be able to include extra information in the response
Paging	PagingCriteria	To be able to return specific page results

ExpositionIncludes

Name	Type	Description
Pricing	bool	Include the tickets and pricing to CombiExpositionTickets
Expositions	bool	Includes the Expositions linked to the CombiExposition
TimePeriodCategories	bool	Includes the TimePeriodCategories linked to the expositions

Response

The response consists of a Result, which in turn contains a list of CombiExpositions.

Exposition

Name	Type	Description
Id	Guid	The unique identifier of the Combi Exposition.
ShortName	String	The ShortName of the Combi Exposition.
Name	String	The Name of the Combi Exposition
Description	String	The Description of the Combi Exposition.
AllowedOverMultipleDays	Bool	Incidates that the Combi Exposition allows period selection over multiple days
FixedChronologicalOrder	Bool	Indicates that periods for the Combi Exposition have to be selected in chronological order
CustomerPriceGroupId	Guid?	The overriding customer price group id that should be used in price calculation if specified
Expositions	List<CombiExpositionExposition>	List of Expositions linked to the Combi Exposition
CombiExpositionTickets	List<CombiExpositionTicket>	List of CombiExpositionTickets that can be bought
CombiExpositionSteps	List<CombiExpositionStep>	List of the CombiExpositionSteps that are linked to the Combi Exposition

CombiExpositionExposition

This is an Exposition linked to a CombiExposition

Name	Type	Description
Id	Guid	The unique identifier of the CombiExpositionExposition.
ExpositionId	Guid	The unique identifier of the Exposition
Sequence	Int	The order in which the periods for each Exposition should be selected
MinTimeBefore	DateTime?	The minimum time that has to be available between 2 chosen periods
ShortName	String	The ShortName of the Exposition
Description	String	The Name of the Exposition
Comment	String	The comment/long description of the Exposition
ExpositionTimePeriodCategories	List<ExpositionTimePeriodCategorie>	The list of TimePeriodCategories linked to the Exposition

CombiExpositionTicket

This is a ticket for a CombiExposition

Name	Type	Description
Id	Guid	The unique identifier of the CombiExpositionTicket
Shortname	String	The ShortName of the CombiExpositionTicket
Description	String	The Description of the CombiExpositionTicket
CombineTickets	Bool	Indicates whether the ETickets of the CombiExpositionTicket should be combined into 1 ETicket.
TicketLayout	Int	Indicates which TicketLayout should be used for the ETickets
Tickets	List<CombiExpositionTicketPrice>	The list of individual tickets/prices part of the CombiExpositionTicket
Price	decimal	The sum of the prices of the list of Tickets
ExtraArticles	List<CombiExpositionTicketExtraItem>	The list of the "automatic articles" of a combiExpositionTicket
OptionalArticles	List<CombiExpositionsTicketOptionalArticle>	The list of the optional articles of a combiExpositionTicket

CombiExpositionTicketPrice

This is the price of the individual tickets being part of the CombiExpositionTicket

Name	Type	Description
Id	Guid	The unique identifier of the CombiExpositionTicketPrice
ExpositionItemId	Guid	The unique identifier of the Exposition ticket
ShortName	string	The ShortName of the Exposition ticket
Description	string	The description of the Exposition ticket
Price	decimal	The price of the Exposition ticket

CombiExpositionStep

This is a step for a combiExposition

Name	Type	Description
Id	Guid	The unique identifier of the Combi Exposition step
ShortName	string	The shortname of the Combi Exposition
Description	string	The description of the Combi Exposition
Sequence	Int	The order in which the steps should be shown
Optional	Bool	Indicate whether the step is mandatory
Comment	string	The comment of the combiExpositionStep
CombiExpositionId	Guid	The unique identifier of the Combi Exposition

CombiExpositionTicketExtraItem

This is a ExtraItem for a combiExpositionTicket

Name	Type	Description
Id	Guid	The unique identifier of the Combi Exposition ticket ExtraItem

ShortName	string	The shortname of the Combi Exposition ticket ExtraItem
Description	string	The description of the Combi Exposition ticket ExtraItem
ItemId	Guid	The ID of the item
Quantity	Int	The quantity of this ExtraItem
UnitPrice	decimal	The price of one ExtraItem
Price	decimal	The price of one ExtraItem * Quantity

CombiExpositionsTicketOptionalArticle

This is a Optional article for a combiExpositionTicket

Name	Type	Description
Id	Guid	The unique identifier of the Combi Exposition ticket OptionalArticles
ShortName	string	The shortname of the Combi Exposition ticket OptionalArticles
Description	string	The description of the Combi Exposition ticket OptionalArticles
ItemId	Guid	The ID of the item
Quantity	Int	The quantity of this OptionalArticles
UnitPrice	decimal	The price of one ExtraItem
Price	decimal	The price of one ExtraItem * Quantity
Optional	bool	Indicates whether the OptionalArticle is optional
MinQuantity	Int	Minimum you have to buy for the visit
MaxQuantity	Int	Maximum you can buy for the visit
MaxPerTimeSlot	Int	Maximum you can buy per timeslot
MaxPerTicket	Int	maximum you can buy per ticket
PriceCalcForSalesItem	Int	The calculationmethod for salesItem
CombiExpositionTicketId	Guid	The unique identifier of the CombiExpositionTicket

FindExpositionArticlePrices

OptionalAndAutomaticExpositionArticlesRequest

The request consist of the service context and an FindExpositionArticlePricesCriteria object

FindExpositionArticlePricesCriteria

Name	Type	Description
ExpositionId	Guid	The id of the exposition u want the prices from
ExpositionPeriodId	Guid	The id of the period that u want prices for.
PersonId	Guid	The person id that will be used to base the price group on.

Response

Returns the updated prices of exhibition articles based on the date and price and if we take the price of booking date or the price of visit date

OptionalAndAutomaticExpositionArticles

Name	Type	Description

AutomaticArticles	IList<ExpositionArticle>	List of automatic article
OptionalArticles	IList<ExpositionArticleOptonal>	List of optional articles

ExpositionArticleOptional

Name	Type	Description	Remarks
Id	GUID	Id of the optional article	
ArticleId	GUID	Id of the article	
Description	String	Description of the article	
ShortName	String	ShortName of the article	
Amount	Decimal	The actual price.	
Group	PriceGroup	The price group to which the price applies.	
CalculationType	CalculationType	Enumeration of calculation method. The possible values are: • PriceTimesHours = 0 • FixedPrice = 1 • PriceTimesQuantity = 2 • PriceTimesQuantityTimesHours = 3	The price is either fixed, either dependent on the number of persons, the number of hours or the number of hours and persons. • Price x Hour • Fixed • Price x Number • Price x Number x Hour
MinimumQuantity	Int	The minimum quantity the price is available from	Ex: • € 5,00 for 0 to 9 • € 4,00 for 10+
ExpositionAmount	Decimal	The price for the exposition ticket	Amount = ExpositionAmount + ArticleAmount
ArticleAmount	Decimal	The added price for additional article sales	
ActivePeriods	List<Period>	The active periods in which the exposition ticket can be sold/bought	
MinSaleQuantity	Int	The minimum amount of tickets which need to be sold /bought for the exposition ticket	
MaxSaleQuantity	Int	The maximum amount of tickets which need to be sold /bought for the exposition ticket	
SelectionType	SelectionType	The type of selection • Normal • Vouchers • Both	
AdditionalInformation	String	The extra information in a chosen language	

FindCombiOrganisedVisits

Calling up this Webservice operation will visualize the desired combi organised visit(s) on the basis of the linked parameters.

Request

The request consists of the service context and an CombiOrganisedVisitSearchCriteria object.

ExpositionSearchCriteria

Name	Type	Description	Remarks
ID	GUID	If filled in: The ID of the specific CombiOrganisedVisit one wants to find.	
PersonID	String	If filled in: retrieves CombiOrganisedVisits for that person	

Response

A list of expositions. An exposition consists of the following parts:

CombiOrganisedVisit

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the CombiExposition.	
CustomerId	GUID	Id of the person	
PurchaseDate	DateTime	Purchase date of the combi organised visit	
Amount	Decimal	Total price of the combi organised visit	
Quantity	Int	Quantity of combi organised visits bought	
CombiOrganisedVisitReservations	IList<CombiOrganisedVisitPeriodReservation>	List of the period reservations linked to the combi organised visit	
VisitNo	Long	The organised visit number	

CombiOrganisedVisitPeriodReservation

Name	Type	Description	Remarks	
Id	GUID	The unique identifier of the CombiOrganisedVisitPeriodReservation.		
CombiOrganisedVisitId	GUID	Id of the CombiOrganisedVisit		
ExpositionId	GUID	Id of the exhibition		
ExpositionPeriodId	GUID	Id of the exhibition period		
UnitPrice	Decimal	unit price of the exhibition		
TicketTemplateId	GUID	Id of the ticket template		
PaidQuantity	Int	Quantity of tickets that are paid		
Quantity	int	Quantity of tickets ordered		
CanBeRebooked	bool	Boolean if a combi organised visit can be rebooked		
ArticleId	GUID	Id of the combi exhibition ticket		
ArticleCode	string	Shortname of the combi exhibition ticket		
ArticleName	string	Description of the combi exhibition ticket		

GetCombiExpositionAllowedPeriods

Request

The request consists of the service context and a GetCombiExpositionAllowedPeriodsCriteria object.

GetCombiExpositionAllowedPeriodsCriteria

Name	Type	Description
CombiExpositionId	Guid	The ID of a specific CombiExposition
ExpositionId	Guid	the ID of the exposition u want the allowed periods from
FixedPeriodIds	List<Guid>	the periods previously selected for the previous expositions.

Response

Name	Type	Description
Periods	List<Period>	the list of allowed periods that can be used to find the expositionperiods with

Periods

Name	Type	Description
From	DateTime	the start time
Until	DateTime	he end time

12 Ticketing module

CreateCultureEvent

This function creates a new cultureevent, based on the cultureevent given in the parameters.
This function will take the following properties into acount:

- Code
- Description
- From
- Until
- BreakFrom
- BreakUntil
- CultureEventCategory (tries to link either by id, code or name, and creates it if it doesn't exist)
- CultureActivity (tries to link either by id, code or name, and creates it if it doesn't exist)
- Hall (tries to link either by id, code or name, and returns an error if non-existing). The Hall must have a ReservationOrdering named BAS to complete the process.
- Prices (groups will be linked using code and will not be created if the group doesn't exist)
- Status (opened or closed)

Request

The request consists of the service context and a **CultureEvent** object.

Response

The response object consists of a message (ok if all is good or an error message if something went wrong) and the created CultureEvent.

FindCultureEvents

Request

The request consists of a service context and a CultureEventSearchCriteria object.
The properties of a CultureEventSearchCriteria look as follows:

CultureEventSearchCriteria

This object determines the search criteria that are applied.

Name	Type	Description	Remarks
CultureActivityId	GUID?	The ID of the linked CultureActivityId.	Nullable, not obligatory.
CultureEventId	GUID?	The unique ID of the CultureEvent?	Nullable, not obligatory.
From	DateTime?	Start date of the culture event.	Nullable, not obligatory.
Includes	CultureEventIncludes	Determines which parts of the culture event will be included.	
Name	String	Part of the name of the culture event / description.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
Until	DateTime?	End date of the culture event.	Nullable, not obligatory.
CultureEvent-CategoryId	GUID?	The ID of the linked CultureCategory. A list can be called up using ListCultureEventCategories.	Nullable, not obligatory.
CultureActivityIdList	IList<GUID>	A list of CultureActivityID's on which can be filtered.	Nullable, not obligatory.

CultureEventCategory IdList	IList<GUID>	A list of CultureCategoryID's on which can be filtered.	Nullable, not obligatory.
PersonId	Guid?	An unique identifier used for price calculation on CultureEventOptions	Nullable, not obligatory.
TraderId	Guid?	Search the culture events linked with this trader	Nullable, not obligatory.
HallId	Guid?	Search the culture events linked to a certain hall	Nullable, not obligatory.
CultureSubscriptionId	Guid?	Search the extra culture events for a certain culture subscription	Nullable, not obligatory. Only applicable for culture subscriptions (=ticketing memberships).

CultureEventIncludes

This object allows to exclude certain parts of a culture event.

Name	Type	Description	Remarks
Image	Bool	Include the image of the event or not.	
ImageUrl	Bool	Include the image URI of the event or not.	This URI can be used in the imaging service
Options	Bool	Find the options of the current culture event or not.	
HallImage	Bool	Include the image of the hall or not	
HallImageUri	Bool	Include the image URI of the hall or not	This URI can be used in the imaging service

Response

The result of the search function is a list of CultureEvents.

CultureEvent

CultureEvent is the object that contains all information on ticketing (or culture) events.

Name	Type	Description	Remarks
AdministrativeCost	AdministrativeCost		
AvailableSeats	Int32	The number of available seats that can still be booked using the Webservices.	
Code	String	The code of the CultureEvent.	
CultureActivity	CultureActivity	The culture activity to which the CultureEvent belongs.	
Description	String	The description of the CultureEvent.	
From	DateTime?	The start date of the CultureEvent.	Nullable
Hall	Hall		
Id	GUID	The unique identifier of the CultureEvent.	
Image	Picture	The picture of the CultureEvent.	
ImageUrl	String	The Url on which the image can be found.	
IncassoCost	IncassoCost	The additional costs in case of direct debit payment. See 6.12.2: Response.	
MaxTicketsPerAddress	Int32	The maximal number of tickets that can be bought by the same person for this event.	
MaxVisitors	Int32	The maximal number of participants to this event.	
MaxVisitorsWeb	Int32	The maximal number of participants to this event that can register on the Website.	
Name	String	The name of the CultureEvent.	
Prices	Collection<CultureEventPrice>	A list of prices of the CultureEvent.	
ReservationCost	ReservationCost	The additional reservation costs for this event.	
SalesFrom	DateTime?	Date as of which tickets can be sold.	Nullable
SalesUntil	DateTime?	Date until which tickets can be sold.	Nullable
SeatSequenceId	GUID?	Is not used by Webservices.	Nullable

Status	CultureEventStatus	Possible values are: • Open • Closed • NotForSale	
Until	DateTime?	The end date of the CultureEvent.	Nullable
VatId	GUID	The identifier of the VAT tariff that applies to this event.	
AllowBasReservation	Bool	Indication for the Web designer that the seats of this event can be reserved using the BAS system.	
AllowExplicitSeatReservation	Bool	Indication for the Web designer that the seats of this event can be reserved by manual selection.	
ShippingCost	ShippingCost	Provides an indication of the cost for shipping tickets.	
Options	IList<CultureEventOption>	A collection of CultureEventOptions linked with the CultureEvent	
CultureEventCategory	CultureEventCategory	The CultureEventCategory linked with the CultureEvent	
UseReservationCosts	Bool	Indication for the web designer that the CultureEvent uses reservation costs	
AvailableSitePlaces	Int	The available site places	
AvailableTotal	int	Available total places	
BreakFrom	DateTime?	Optional start date of the break	
BreakUntil	DateTime?	Optional end date of the break	
AllowWaitingListReservation	Bool	Indication that the CultureEvent may use the waiting list	
MaxTicketsPerAddress	Int	Maximum tickets per user	
SoldOut	Bool	Indication if the CultureEvent is sold out	
AllowOverbooking	Bool	Indication if the CultureEvent may overbook	
GenLoyaltyActive	Bool?	Indication if Gen Loyalty is active	
ShowHallForBas	Bool	Indication if the hall overview must be shwon after a reservation (webshop purpose)	
HideAmountofFreePlaces	Bool	Indication if the free places must not be shown	
ShippingCost	ShippingCost	The shipping cost for this CultureEvent	
DonationEnabled	Bool	Indication if donation is allowed on this event	
AllowChangeSeats	Bool	Indication if it is allowed to change seats after BAS selected the seats	
ExtraDescription	String	The extra description of the CultureEvent	

CultureEventPrice

A CultureEvent can contain several prices. Each of these prices has a specific price group.
Such a price looks as follows:

Name	Type	Description	Remarks
ID	GUID	The unique identifier of the price specified in ReCreateX.	
Amount	Decimal	The actual price.	
Group	PriceGroup	The price group to which the price applies.	
CustomerID	Guid?	The customer to whom the price applies.	
SeatBlock	SeatBlock	The collection of rows (block) to which the price applies.	
SeatRow	SeatRow	The row to which this price applies.	
Seat	Seat	The seat to which this price applies.	
SeatRange	SeatRange	The rank to which this price applies.	
CultureSite	Site	The site t which the price applies	
StartDate	DateTime?	The date as of which the price is valid. If this date is filled in, this price only applies for bookings made after the given date.	
MinimumQty	Int?	If filled in, this price only applies for sales as of x number of tickets.	

ShippingCost

A CultureEvent can have a price for shipping tickets using a postal service.
To let users choose between shipping or not shipping tickets, it can be good to provide an indication of how much it will cost to send these tickets.

Name	Type	Description	Remarks
Name	string	The name for the cost	
Cost	Decimal	The actual price.	
Type	Int	How this cost will be used.	0 = cost per ticket 1 = cost per reservation 2 = cost per event
MaximumCost	Decimal	The maximum cost	

FindCultureReservations

This function generates an overview of the tickets booked for a certain person.

Request

The request consists of the service context and a **CultureReservationSearchCriteria** object.

Name	Type	Description	Remarks
PersonId	GUID	The unique ID of the person for whom you want to call up the reservation history.	To be filled in obligatory!
CultureEventId	GUID	The unique ID of the event of which you want to call up the reservation history.	
CultureEventReservationId	GUID	The unique ID of the reservation you want to call up.	
From	DateTime?	Start date of the reservation history you want to call up. Reservations with a reservation date before the given date will not be visualised.	
Until	DateTime?	End date of the reservation history you want to call up. Reservations with a reservation date after the given date will not be visualised.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
CultureReservationIncludes	Includes	Indicate if the response must include additional data.	

CultureReservationIncludes

A CultureReservationIncludes object contains the following properties

Name	Type	Description	Remarks
Customers	Bool	Include the customers	
IncludeLocked	Bool	Include locked CultureReservations	

Response

The response object consists of a list of **CultureReservation** objects.

CultureReservation

Name	Type	Description	Remarks
Id	GUID	The unique ID of the culture reservation.	
ReservationNumber	Int64	The unique reservation number.	
ReservationDate	DateTime	Reservation date.	

ReservationSeats	ICollection<CultureReservationSeat>	List of the seats that have been reserved.	
ReservationSites	ICollection<CultureReservationSite>	List of the sites that have been reserved.	
PaymentMethodName	String	Name of the payment method by means of which this reservation has been paid.	
PaymentMethodId	GUID	Unique ID of the payment method by means of which this reservation has been paid.	
PaymentMethod	BasketPaymentMethod	Payment method by means of which this reservation has been paid.	See 6.12.2: Response
CultureEventId	GUID	Unique ID of the event.	
CultureEvent	CultureEvent	Details concerning the event.	See 12.3.2.1: CultureEvent
CustomerId	GUID	Unique ID of the person to who this reservation applies.	
Customer	Person	Details concerning the personal data.	See 8.8.1.1: Person
Payed	Bool	Has this reservation already been paid?	
SalesLineId	GUID	Sales line for the reservation.	
Options	ICollection<CultureReservationOption>	Options linked to reservation	
UnitPrice	decimal	Total price without VAT	
DiscountAmount	decimal	Discount amount	
LineAmount	decimal	Total price minus discount amount	
VatAmount	decimal	VAT amount	
OptionAmount	decimal	Total option price	
CancelReturnAmount	decimal	Returned cancellation amount	
CancelCostAmount	decimal	Cancellation cost	
ReservationAmount	decimal	Reservation cost	
InvoiceCost	decimal	Administrative cost	
ShippingCost	decimal	Shipping cost	
CouponDiscount	decimal	Coupon discount	
GenLoyaltyCardDiscount	decimal	Loyalty discount	
Amount	decimal	Total price with VAT	
PostedAmount	decimal	Paid amount	
Balance	decimal	Outstanding amount	
CultureSeriesId	Guid?	The linked culture series id	

CultureReservationSeat

Name	Type	Description	Remarks
Id	GUID	The unique ID.	
CultureEventId	GUID	Unique ID of the event.	
CultureEvent	CultureEvent	Details concerning the event.	
SeatId	GUID	The unique ID of the reserved seat.	
Seat	Seat	Details concerning the seat.	
PriceGroupName	String	Name of the price group.	
PriceGroup	Price	Details concerning the price group.	
Price	Decimal	The price of the reserved seat.	

CultureReservationSite

Name	Type	Description	Remarks
Id	Guid	The unique ID.	
ReservationId	Guid	The reservation ID.	
CultureEventID	Guid	Unqiue ID of the event.	
CultureEvent	CultureEvent	Details concerning the event	
NumberOfPlaces	Int	Number of places reserved on the site	
CultureSite	CultureSite	Details concerning the CultureSite	
ReservationSitePlaces	ICollection<CultureReservationSitePlace>	Details concerning the CultureReservationSitePlaces	

CultureReservationSitePlace

Name	Type	Description	Remarks
Id	Guid	The unique ID.	
CustomerID	Guid	Unique ID of the customer.	
PriceGroup	PriceGroup	Details concerning the PriceGroup	
Price	Decimal	The price of the reservation	

CultureReservationOption

Name	Type	Description	Remarks
Id	Guid	The unique ID	
CultureEventId	Guid	Unique ID of the event	
CultureEventOptionId	Guid	Unique ID of the option	
Quantity	int	Quantity	
Price	decimal	The price of the option	

FindCultureSeries

Request

The request consists of a service context and a CultureSeriesSearchCriteria object.
The properties of a CultureSeriesSearchCriteria look as follows:

CultureSeriesSearchCriteria

This object determines the search criteria that are applied.

Name	Type	Description	Remarks
CultureSeriesId	Guid?	The unique ID of the CultureSerie	Nullable, not obligatory.
NamePattern	string	Part of the name of the cultureseries / description	
CultureActivityId	Guid?	The ID of the linked CultureActivityId.	Nullable, not obligatory.
HallId	Guid?	Receive all culture series which makes use of this hall	Nullable, not obligatory.
Includes	CultureSeriesIncludes	Determines which parts of the culture series will be included.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

CultureSeriesIncludes

This object allows to exclude certain parts of a culture serie.

Name	Type	Description	Remarks
Image	Bool	Include the image of the series or not.	
ImageUrl	Bool	Include the image URI of the series or not.	This URI can be used in the imaging service
Prices	Bool	Find the prices of the culture series or not.	
Hall	Bool	Find the hall of the culture series	
CultureEvents	Bool	Find the culture events of the culture series	
HallImage	Bool	Include the image of the hall or not	
HallImageUri	Bool	Include the image URI of the hall or not	This URI can be used in the imaging service

Response

The result of the search function is a list of CultureSeries.

CultureSeries

CultureSeries is the object that contains all information on a CultureSeries.

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the culture series	
Code	string	The shortname of the culture series	
Name	string	The name of the culture series	
Description	string	The description of the culture series	
MaxVisitors	int	The max visitors of the culture series	
CultureActivity	CultureActivity	The culture activity of the culture series	
HallId	Guid	The hallid of the culture series	
Hall	Hall	The hall of the culture series	
CultureEvents	IList<CultureEvent>	The culture events linked with the culture series	
Prices	IList<CultureSeriesPrice>	The prices linked with the culture series	
AllowExplicitSeatReservation	bool	Indicates if explicit seat reservations are allowed.	
AllowBasReservation	bool	Indicates if bas reservations are allowed.	
ShowHallForBas	bool	Indicates if showing the hall for bas is allowed.	
AllowSingleFreeSeats	bool	Indicates if single free seats are allowed.	
Image	string	The image of the culture series.	
ImageUri	string	The image uri of the culture series	
ShippingCost	ShippingCost	The shipping cost details	
AllowChangeSeats	bool	Indication if it is allowed to change seats after BAS selected the seats	

CultureSeriesPrice

A CultureSeries can contain several prices. Each of these prices has a specific price group. Such a price looks as follows:

Name	Type	Description	Remarks
ID	GUID	The unique identifier of the price specified in ReCreateX.	
Amount	Decimal	The actual price.	
Group	PriceGroup	The price group to which the price applies.	
SeatRange	SeatRange	The rank to which this price applies.	

FindCultureWaitingListReservations

This function generates an overview of the tickets booked for a certain person.

Request

The request consists of the service context and a **CultureWaitingListReservationSearchCriteria** object.

CultureWaitingListReservationSearchCriteria

Name	Type	Description	Remarks
CultureWaitingListReservationId	GUID	The unique ID of the waiting list reservation you want to call up.	
PersonId	GUID	The unique ID of the person for whom you want to call up the waiting list history.	To be filled in obligatory!
CultureEventId	GUID	The unique ID of the event of which you want to call up the waiting list history.	
From	DateTime?	Start date of the reservation history you want to call up. Reservations on the waiting list with a reservation date before the given date will not be visualised.	
Until	DateTime?	End date of the reservation history you want to call up. Reservations on the waiting list with a reservation date after the given date will not be visualised.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	CultureWaitingListReservationIncludes	Determines which data you want to call up.	

CultureWaitingListReservationIncludes

This object allows to exclude certain parts of the data.

Name	Type	Description	Remarks
PersonDetails	Bool	Calls up the data of the person or not.	

Response

The response object consists of a list of **CultureWaitingListReservation** objects.

CultureWaitingListReservation

Name	Type	Description	Remarks
Id	GUID	The unique ID of the waiting list reservation.	
CultureEventId	GUID	Unique ID of the event.	
CultureEvent	CultureEvent	Details concerning the event.	See 12.3.2.1: CultureEvent
CreationDate	DateTime	Creation date of the waiting list reservation.	
CustomerId	GUID	Unique ID of the person to who this reservation applies.	
Customer	Person	Details concerning the personal data.	See 8.8.1.1: Person
NumberOfSeats	Int	The desired number of seats.	
NumberOfSitePlaces	Int	The desired number of standing places.	
Comment	String	Remarks concerning the waiting list reservation.	

GetCultureEventSeating

Returns a hall with all available info on the seats, rows, blocks, ranks and locations for a specific culture event. It is possible that the ranks diverge from the standard hall plan (that can be called up using the GetHallSeating method) per event.
This method replaces in fact the GetHallSeating method. It is advised to use this method instead of the GetHallSeating method. If no ranks have been defined on event level, the ranks of the hall will be taken.

Request

The request consists of the service context and the ID of the event.

Response

The response consists of one (or no) hall object.

Hall

The hall object contains following properties:

Name	Type	Description	Remarks
Address	Address	The location address linked to the hall	
Blocks	IList<SeatBlock>	List of all blocks within the hall	
Code	String	Code of the hall	
Description	String	Description of the hall	
id	Guid	Unique identifier of the hall	
Name	String	Name of the hall	
Rows	IList<SeatRow>	List of all rows within the hall	
CultureSites	IList<CultureSite>	List of all sites within the hall	Site = block for standing places
HallElements	IList<HallElement>	List of all additional hall elements	
Image	Picture	Image of the hall	
ImageUri	String	Image URI of the hall	Use the Imaging service to use the ImageUri

GetCultureEventSeatingWithIncludes

Returns a hall with all available info on the seats, rows, blocks, ranks, hallelements and locations for a specific culture event. It is possible that the ranks diverge from the standard hall plan (that can be called up using the GetHallSeatingWithIncludes method) per event.
This method replaces in fact the GetHallSeatingWithIncludes method. It is advised to use this method instead of the GetHallSeatingWithIncludes method. If no ranks have been defined on event level, the ranks of the hall will be taken.

Request

The request consists of the service context and the ID of the event.

Name	Type	Description	Remarks
CultureEventId	GUID	The unique identifier of the CultureEvent.	
SeatingIncludes	SeatingIncludes	Defines if you want to return HallElements	

SeatingIncludes

Name	Type	Description	Remarks
ExtraElements	Bool	Determines if HallElements should be returned (can contain images)	
HallImage	Bool	Include the image of the Hall?	
HallImageUri	Bool	Include the image URI of the Hall?	This URI can be used in the imaging service

Response

The response consists of one (or no) hall object.

GetCultureSeriesSeating

Returns a hall with all available info on the seats, rows, blocks, ranks and locations for a specific culture series. It is possible that the ranks diverge from the standard hall plan (that can be called up using the GetHallSeating method) per series.
This method replaces in fact the GetHallSeating method. It is advised to use this method instead of the GetHallSeating method. If no ranks have been defined on event level, the ranks of the hall will be taken.

Request

The request consists of the service context and the ID of the series.

Response

The response consists of one (or no) hall object.

GetCultureSeriesSeatingWithIncludes

Returns a hall with all available info on the seats, rows, blocks, ranks, hallelements and locations for a specific culture series. It is possible that the ranks diverge from the standard hall plan (that can be called up using the GetHallSeatingWithIncludes method) per series.
This method replaces in fact the GetHallSeatingWithIncludes method. It is advised to use this method instead of the GetHallSeatingWithIncludes method. If no ranks have been defined on series level, the ranks of the hall will be taken.

Request

The request consists of the service context and the ID of the series.

Name	Type	Description	Remarks
CultureSeriesId	GUID	The unique identifier of the CultureSeries.	
SeatingIncludes	SeatingIncludes	Defines if you want to return HallElements	

SeatingIncludes

Name	Type	Description	Remarks
ExtraElements	Bool	Determines if HallElements should be returned (can contain images)	
HallImage	Bool	Include the image of the Hall?	
HallImageUri	Bool	Include the image URI of the Hall?	This URI can be used in the imaging service

Response

The response consists of one (or no) hall object.

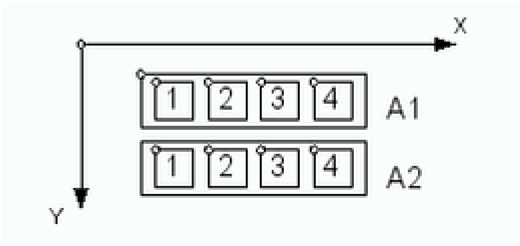
GetHallSeating

Returns a venue with all available info on the seats, rows, blocks, ranks and locations.
This method is not required as of WSDL 2.1 and is preferably replaced by the GetCultureEventSeating method.
A **venue** *(Hall)* consists of:

- The seats in the venue itself (optional)
- The blocks into which a venue can be devided (optional)Blocks allow to manage large venues more easily.(both at the same time is possible as well)
- CultureSites (Standing places)(optional)
- HallElements (if you want to have hallelements, please use GetHallSeatingWithIncludes)

A **block** *(SeatBlock)* consists of rows. ('**zone**' is a synonym for block)
A **row** *(SeatRow)* consists of **seats** *(Seats)*.
Blocks, rows and seats contain location information (coordinates and size) allowing to make a simple visual representation of the layout of a venue (or block).

- The unit of the location information is not determined. It is only the relative location of the seats towards each other that is important. (A diagram can be drawn in ReCreateX, the units used are millimetres.)
- All coordinates [Location.X, Location.Y] are relative towards the container in which the seating elements are. This means:
  - seats are always part of rows
  - rows are part of a block or directly part of a hall
  - blocks are part of a hall
- The coordinates are those of the upper left-hand corner (smallest X and Y coordinates).
- The size of an item [Location.Dx, Location.Dy] allows you to visualise a symbol (rectangle, circle, image, ...) that is not bigger than the given size.
- It is advised not to show the seats of different blocks on one diagram.(This could lead to the visualisation of too much information).It is better to first show the blocks of the venue (or the seats in the venue itself).After the selection of a block, it is possible to show the seats in that block.



For example: On the example above you can see 2 rows (A1 and A2), each containing 4 seats (1, 2, 3 and 4). Each little ball depicts the reference point of the element (always the upper left-hand corner). The coordinates of seat 1 (in row A1) are relative compared to row A1 (e.g. [2, 1]). The coordinates of row A1 are relative compared to the origin (e.g. [10, 5]). Thus, if you would only draw the seat, you would have to draw a rectangle from [12, 6] to [20, 14]. (supposing that DX and DY of that seat are 8. [10 + 2, 5 + 1] to [10 + 2 + 8, 5 + 1 + 8])
Remark: Since the layout of a venue does not change frequently and since the number of seats in a venue can be quite high, it is advised to cach this information.

Request

The request consists of the service context and the ID of the desired hall.

Response

The response consists of one (or no) hall object.

GetHallSeatingWithIncludes

Returns a venue with all available info on the seats, rows, blocks, ranks and locations.
This method is for compatibility only and is preferably replaced by the GetCultureEventSeatingWithIncludes method.
A **venue** *(Hall)* consists of:

- The seats in the venue itself (optional)
- The blocks into which a venue can be devided (optional)Blocks allow to manage large venues more easily.(both at the same time is possible as well)
- CultureSites (Standing places)(optional)
- HallElements (optional)

A **block** *(SeatBlock)* consists of rows. ('**zone**' is a synonym for block)
A **row** *(SeatRow)* consists of **seats** *(Seats)*.
Blocks, rows and seats contain location information (coordinates and size) allowing to make a simple visual representation of the layout of a venue (or block).

- The unit of the location information is not determined. It is only the relative location of the seats towards each other that is important. (A diagram can be drawn in ReCreateX, the units used are millimetres.)
- All coordinates [Location.X, Location.Y] are relative towards the container in which the seating elements are. This means:
  - seats are always part of rows
  - rows are part of a block or directly part of a hall
  - blocks are part of a hall
- The coordinates are those of the upper left-hand corner (smallest X and Y coordinates).
- The size of an item [Location.Dx, Location.Dy] allows you to visualise a symbol (rectangle, circle, image, ...) that is not bigger than the given size.
- It is advised not to show the seats of different blocks on one diagram.(This could lead to the visualisation of too much information).It is better to first show the blocks of the venue (or the seats in the venue itself).After the selection of a block, it is possible to show the seats in that block.

HallElements contain a JSON String that provides exported information for displaying simple elements in a Hall such as rectangles, images, and text. Information for complex objects is not exported (for example 3D objects or paths do not contain their full information).
For example: On the example above you can see 2 rows (A1 and A2), each containing 4 seats (1, 2, 3 and 4). Each little ball depicts the reference point of the element (always the upper left-hand corner). The coordinates of seat 1 (in row A1) are relative compared to row A1 (e.g. [2, 1]). The coordinates of row A1 are relative compared to the origin (e.g. [10, 5]). Thus, if you would only draw the seat, you would have to draw a rectangle from [12, 6] to [20, 14]. (supposing that DX and DY of that seat are 8. [10 + 2, 5 + 1] to [10 + 2 + 8, 5 + 1 + 8])
Remark: Since the layout of a venue does not change frequently and since the number of seats in a venue can be quite high, it is advised to cach this information.

Request

The request consists of the service context and the ID of the desired hall.
A Includes objects has to be passed so you can

Response

The response consists of one (or no) hall object.

GetSeatAllocations

This operation allows you to call up the availability of seats for a show (CultureEvent). The availability of a seat is described by a *Seat Allocation* object. A Seat Allocation object contains a reference to the seat (seat ID) and its Allocation Status.Possible allocation statuses are:

- Available [*Available*]
- Reserved [*Reserved*]
- Blocked [*Blocked*]
- Temporarily reserved *[Locked]*
- In option [*Option*]

For the end user it is usually only important to know the difference between available and unavailable seats.
In addition to calling up the availability of individual seats, it is also possible to ask for the summaries [*Summaries*] of the availability of seats per block and / or per hall. (It is for example possible to indicate per block how many seats are still available.)
Calling up the individual availability and the grouped summaries of availability is done using one and the same Webservice operation.
Together with the data of 12.6 it is possible to visualise the allocation of seats.

Request

The request consists of the service context and the following fields.

Name	Type	Description	Remarks
EventId	GUID?	The unique identifier of the CultureEvent.	Obligatory when CultureSeriesID is not filled in.
SeatBlockCode	String	The unique code of a block. 'The 'SelectedArea' is then the block instead of the hall).	Optional.
AllocationIncludes	Enum	Which Seat Allocations do you want to show: • None: you only want to call up summaries • All: allocations in all blocks and in the hall itself • SelectedArea: only the seats in the hall itself (if SeatBlockCode is empty) or only the seats in the given block (if SeatBlockCode is filled in).	Obligatory.
CultureSeriesID	GUID?	The unique identifier of the culture series.	Obligatory when EventID is not filled in

Response

A SeatAllocationOverview object.

- Overview.Allocations: the desired seat allocation objects
- Overview.Summary: a summary of the availability of seats grouped per block or per hall.
  - Summary.TotalCount: the counting of all seats
  - Summary.Count: the counting of the seats of the hall itself
  - Summary.BlockSummaries[n].Count: the counting of the seats in a specific block
- Each 'counting' object (SeatAllocationCount) contains the following statistics:
  - [Available]: int
  - [Blocked]: int
  - [Locked]: int
  - [Reserved]: int
  - [Option]: int
  - [ResellerOnly]: int
  - [CustomerOnly]: int
- You can calculate the rest yourself:
  - [Unavailable] = [Blocked] + [Locked] + [Reserved] + [Option] + [ResellerOnly] + [CustomerOnly]
  - [Total] = [Available] + [Unavailable]

ListCultureActivities

This method calls up all CultureActivities for the active Division.

Request

No parameters required.

Response

The response is a list of CultureActivity objects.

CultureActivity

Name	Type	Description	Remarks
Id	GUID	The unique ID of the CultureActivity.	
Code	String	Code of the CultureActivity.	
Name	String	Name of the CultureActivity.	

ListCultureEventCategories

This function allows to call up a list of CultureEventCategories for the current division so that, if required, the FindCultureEvents can use a filter to call up only a part of the shows.

Request

No parameters required.

Response

The response is a list of CultureEventCategories.

CultureEventCategory

Name	Type	Description	Remarks
Id	GUID	The unique ID of the CultureEventCategory.	
Code	String	Code of the CultureEventCategory.	
Name	String	Name of the CultureEventCategory.	

ListCultureEventOptions

Request

The request consists of the service context and a ListCultureEventOptionCriteria.
A CultureEventOptionCriteria consists of the following fields.

Name	Type	Description	Remarks
CultureEventId	GUID	The unique identifier of the CultureEvent.	

Response

The response consists of a list of CultureEventOptions. A CultureEventOption object consists of the following fields.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the option.	
Code	String	The code of the option.	
Name	String	The name of the option.	
Price	String	The price of the option.	
ArticleId	GUID	The unique identifier of the article to which the option is linked.	
MaxQuantity	Integer	The maximal number of options.	
Available	Integer	The number of options present.	

ListCultureEventTickets

This function generates an overview of the tickets booked for a certain reservation.

Request

The request consists of the service context and a **CultureEventTicketsSearchCriteria** object.

CultureEventTicketsSearchCriteria

Name	Type	Description	Remarks
CultureEventReservationId	GUID	The unique ID of the reservation of which you want the tickets	To be filled in obligatory!
CultureEventId	GUID	The unique ID of a culture event inside the reservation	

Response

The response object consists of a list of **CultureEventTicket** objects.

CultureEventTicket

Name	Type	Description	Remarks
SeatRow	String	The unique ID of the reservation of which you want the tickets	
SeatRowDescription	String	The unique ID of a culture event inside the reservation	
SeatNumber	String	Number of the seat	
SeatBlock	String	Name of the block where the seat is located	
SeatBlockDescription	String	Description of the block	
SiteId	GUID	ID of the Site	
SiteName	String	Name of the Site	
SiteDescription	String	Description of the Site	
CardNumber	String	CardNumber	Used to generate the barcode
Price	Decimal	The Price of the ticket	
ReservationNumber	String	Number of the reservation	
CultureEventName	String	Name of the culture event	
CultureEventDescription	String	Description of the culture event	
Date	DateTime	Date of the reservation	
StartTime	DateTime	Start date and time of the event	
TemplateId	GUID	ID of the template used in the document service	
HallName	String	Name of the Hall	
HallStreet	String	Street of the Hall	
HallStreetNumber	String	Number of the Hall	
HallBox	String	Box of the Hall	
HallZipcode	String	ZipCode of the Hall	
HallCity	String	City of the Hall	
CustomerFirstName	String	First Name of the customer	
CustomerName	String	Name of the customer	
CustomerMiddleName	String	Middle Name of the customer	
CustomerStreet	String	Street of the customer	
CustomerStreetNumber	String		
CustomerZipcode	String		
CustomerCity	String		
CustomerEmail	String		

CultureEventImage	Byte[]	Image of the culture event	
OptionTemplateId	GUID	Template ID for the options	
CultureEventId	GUID	ID of the Culture Event	
Comment	String	Comment	
ReservationCost	Decimal	Reservation cost	
InvoiceCost	Decimal	Invoice costs	
TotalCost	Decimal	Total cost	
PriceGroupDescription	String	Description of the pricegroup	
PriceGroupShortName	String	Name of the pricegroup	
OGMNumber	String	OGM Number	
ReservationDate	DateTime	The date the reservation was made	
CustomerBox	String	Box of the customer	
CustomerCountry	String	Country of the customer	
HallCountry	String	Country of the hall	
HallTitle	String	Title of the hall	
CultureEventInfo2	String	Extra info (2) of the Culture Event	
CultureEventInfo3	String	Extra info (3) of the Culture Event	
ShippingCost	Decimal	Shipping cost	
CouponDiscount	Decimal	Discount for coupon codes	
GenLoyaltyCardDiscount	Decimal	General loyalty discount	
OptionCost	Decimal	Total cost for all included options	
SeatRangeId	Guid?	ID of the seat range	
Contact	Contact	Details of the contact	
RowImage	byte[]	Image of the seat row	
HallImage	byte[]	Image of the hall	
SeatRange	String	Code of the seat range	
SeatRangeDescription	String	Description of the seat range	

ListCultureSubscriptions

Request

The request consists of a service context and a CultureSubscriptionSearchCriteria object.
The properties of a CultureSubscriptionSearchCriteria look as follows:

CultureSubscriptionSearchCriteria

This object determines the search criteria that are applied.

Name	Type	Description	Remarks
CultureSubscriptionId	GUID?	The ID of the linked CultureSubscription.	Nullable, not obligatory.
Name	String	Part of the name of the culture subscription / description.	
Includes	CultureSubscriptionIncludes	Determines which parts of the culture subscriptions will be included.	

CultureSubscriptionIncludes

This object allows to exclude certain parts of a culture subscription.

Name	Type	Description	Remarks
CultureEventDetails	Bool	Get the culture event details	

Image	Bool	Include the image of the culture subscription or not	
ImageUri	Bool	Include the image URI of the culture subscription or not	This URI can be used in the imaging service
CultureDiscounts	Bool	Get the culture discounts	
HallImage	Bool	Include the image of the hall or not	
HallImageUri	Bool	Include the image URI of the hall or not	This URI can be used in the imaging service

Response

The result of the search function is a list of CultureSubscriptions.

CultureSubscription

CultureSubscription is the object that contains all information on CultureSubscriptions.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the CultureSubscription.	
ShortName	String	The short name of the subscription.	
Description	String	The description of the subscription.	
SalesStartDate	DateTime	The startdate from when the subscription can be sold.	
SalesEndDate	DateTime	The date until when the subscription can be sold.	
ValidStartDate	DateTime	Date from when the discount is still valid on new reservations (see SubscribersRetainDiscount)	
ValidEndDate	DateTime	Date until when the discount is still valid on new reservations (see SubscribersRetainDiscount)	
FixedPriceGroup	Boolean	Is the pricegroup fixed for this subscription?	
FixedPriceGroupID	GUID	The ID of the fixed Pricegroup.	Empty Guid when there is no fixed group
MinChoiceEvents	Int32	The minimum number of free choice reservations that have to be present in the CultureSubscriptionEntry object.	
MaxChoiceEvents	Int32	The maximum number of free choice reservations that have to be present in the CultureSubscriptionEntry object.	
IsFixedPrice	Boolean	Is there a fixed price for this subscription?	
FixedPrice	Int32	The fixed price, if a fixed price is assigned.	
SubscribersRetainDiscount	Boolean	If you are already subscribed to this susbscription, you get the same discount on new reservations (within a given period)	
CultureSubscriptionEvents	List<CultureSubscriptionEvents>	List of all the culture subscription events for which you can make a reservation	
Comment	String	Extra comment concerning this culture subscription	
Image	Picture	Picture of the culture subscription	
ImageUri	String	Imaging service URL to the image of the culture subscription	
CultureSubscriptionDiscounts	List<CultureSubscriptionDiscount>	List of all the culture subscription discounts	
MaxBonusEvents	Int32	The maximum number of bonus reservations that have to be present in the CultureSubscriptionEntry object.	

CultureSubscriptionEvent

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the CultureSubscription.	
CultureSubscriptionId	GUID	Id of the CultureSubscription to which this event belongs to.	
	CultureSubscription	The CultureSubscription object	

CultureSubscription			
CultureEventId	GUID	The Id of the coupled CultureEvent	
CultureEvent	CultureEvent	The CultureEvent object	
Type	CultureSubscriptionEventType	Defines the type of the linked event: • 0 = Choice (This event can be chosen out of the list of free choice events) • 1 = Fixed (This event must be reserved within the membership) • 2 = Bonus (This event can be reserved as a bonus event with possibly different pricing) • 3 = Extra (This event can be reserved as an additional event after buying the membership)	
BonusPrice	Decimal	The price for the CultureEvent when reserving this as a bonus event	
ExtraTickets	bool	Defines if extra tickets (in a separate reservation) can be bought for this event during the ticketing membership flow	

CultureSubscriptionDiscount

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the CultureSubscriptionDiscount.	
MinQuantity	Int	Minimum number of events to include in the reservation before enjoying the discount	
DiscountAmount	Decimal	The discount amount for the culture subscription reservation	
DiscountPercentage	Decimal	The discount percentage for the culture subscription reservation	

Ticketing – Brief overview

The Webservice provides the possibility to order ticketing reservations through your web site.
The system functions as follows:

- Person selects the ticketing event he/she wants to book.
- 2 different reservation methods are available
  - Best Available Seats (BAS)

= The system automatically selects the best available seats.

-
  - Explicit seats

= Person selects the desired tickets.

- Optionally the person selects extra optional articles.

Detailed management of all ticketing reservations is done in ReCreateX.
Following methods are available within this module:

- ListCultureActivities
- FindCultureEvents
- ListCultureSubscriptions
- ListCultureEventOptions
- GetHallSeating
- GetHallSeatingWithIncludes
- GetCultureEventSeating
- GetCultureEventSeatingWithIncludes
- GetSeatAllocations
- ListCultureEventCategories
- FindCultureReservations
- FindCultureWaitingListReservations
- ListCultureEventTickets
- CreateCultureEvent
- UpdateCultureEvent

UpdateCultureEvent

This function updates an existing cultureevent, based on the cultureevent given in the parameters.

This function will take the following properties into account (if filled in, otherwise they are ignored):

- Id (will search for this Id or will search for the Code, if not found, it returns an error message)
- Name (if not empty)
- From (if specified)
- Until (if specified)
- BreakFrom (if specified)
- BreakUntil (if specified)
- CultureEventCategory (tries to link either by id, code or name, and creates it if it doesn't exist), if not specified, it will be ignored
- CultureActivity (tries to link either by id, code or name, and creates it if it doesn't exist) , if not specified, it will be ignored
- Hall (tries to link either by id, code or name, and returns an error if non-existing). The Hall must have a ReservationOrdering named BAS to complete the process. ,If not specified, it will be ignored.
- Prices (groups will be linked using code and will not be created if the group doesn't exist). If no prices are specified, this will be ignored, otherwise you must supply the complete list of prices, because it will overwrite the current prices
- Status (opened or closed)

Request

The request consists of the service context and a **CultureEvent** object.

Response

The response object consists of a message (ok if all is good or an error message if something went wrong) and the updated CultureEvent (if not closed).

13 Facility bookings module

CalculateReservationCancelCost

This method will calculate the cancel cost of a specific facility booking.

Request

The request is built up as follows:

Name	Type	Description	Remarks
ReservationId	GUID	ID of the facility booking.	
ReasonID	GUID	The cancellation reason.	

Example

```
{
  "ReservationID": "7CEF5435-7011-E811-813F-A0481CB9DBF9",
  "ReasonID": "9941b066-6211-e811-813f-a0481cb9dbf9",
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000641",
    "Password": "*******"
  }
}
```

Response

The response contains a **decimal** object which represents the calculation cost.

Example

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
CalculateReservationCancelCostResponse, ReCreateX.WebShop.WebServices.
Contracts",
    "result": 2.25
}
```

CancelReservation

This method will cancel a specific facility booking.
It is only possible to cancel reservations on the Webshop that:

- Have not been cancelled yet
- Have not been posted yet

Request

The request is built up as follows:

Name	Type	Description	Remarks
ReservationId	GUID	ID of the facility booking.	
ReasonID	GUID	The cancellation reason.	
ReasonDescription	String	The description that is given in case of cancellation.	

Response

The response contains a **ValidationResult** object. This object looks as follows:

Name	Type	Description	Remarks
BrokenRuleName	String	Name of the validation rule.	
Message	String	Error message.	

IsValid	Bool	Rule has been successfully processed.	True = success
			False = fout

DeleteReservationAttachment

This method is used to delete a reservation attachment/comment.

Request

The request consists of the service context and a Guid representing the unique identifier of the ReservationAttachment object.

Response

The response consists of a ValidationResult object that indicates whether or not the delete was successful.

Name	Type	Description	Remarks
IsValid	Bool	Indicates if this is a valid ValidationResult.	
Message	String	The standard error.	
BrokenRuleName	String	The code of the error message.	

FindAvailablePlacesForReservationActivities

This service is used to load all available places per reservation activity provided in the criteria.

Request

The request consists of the service context and an **FindAvailablePlacesForReservationActivitiesCriteria** object.

Name	Type	Description	Remarks
ReservationActivityIds	IList<Guid>	Specify for which reservation activities the available places needs to be loaded	May not be null and must have at least one value
PersonId	Guid?	Some places are disabled for certain customer groups. Specify the customer here.	Optional

Response

The response will contain a list of **AvailablePlacesForReservationActivity** objects that meet the criteria.

Name	Type	Description	Remarks
ReservationActivityId	Guid	ID of the reservation activity	
Places	IList<Place>	The places that are available for this reservation activity.	

FindInfrastructureOpenings

This service calls up the opening hours of one or of all infrastructures between a specific start and end date.

Request

The request consists of the service context and an InfrastructureOpeningsSearchCriteria object.

InfrastructureOpeningsSearchCriteria

Name	Type	Description	Remarks
InfrastructureId	GUID?	ID of the infrastructure of which you want to call up the opening hours.	If not filled in, the data of all infrastructures are called up.
From	DateTime	Calls up the opening hours as of this date.	
Until	DateTime	Calls up the opening hours until this date.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response will contain an InfrastructureOpenings object that meets the criteria.

InfrastructureOpenings

Name	Type	Description	Remarks
OpeningHourColor	Color	Color used within ReCreateX to visualise opening hours.	
ClosingHourColor	Color	Color used within ReCreateX to visualise closing hours.	
ClosingDayColor	Color	Color used within ReCreateX to visualise closing days.	
InfrastructureOpeningHours	IList<InfrastructureOpeningHours>	List of opening hours per infrastructure.	

InfrastructureOpeningHours

Name	Type	Description	Remarks
InfrastructureID	GUID	Unique ID of the infrastructure.	
OpenHours	IList<OpeningHour>	List of opening hours of the infrastructure.	

OpeningHour

An OpeningHour object gives a date with **2 possible time spans** for opening hours.

Name	Type	Description	Remarks
Date	DateTime	Date on which the infrastructure is open.	
From1	DateTime?	As of when is the infrastructure open?	*From1 & To1 belong together*
To1	DateTime?	Until when is the infrastructure open?	
From2	DateTime?	As of when is the infrastructure open?	*From2 & To2 belong together*
To2	DateTime?	Until when is the infrastructure open?	

FindPriceGroupReservationActivityByActivity (as of RCX 5.9.0)

This method returns the possible CustomerGroups with the price to pay for a place, date and reservationactivity.

Request

The request is built up as follows:

Name	Type	Description	Remarks
PlaceId	Guid	The place of the reservation	
ReservationActivity	Guid	The activity of the reservation	
Date	DateTime	The date of the reservations	

Response

The response contains a **PriceGroupReservationActivity** oject. This object looks as follows:

Name	Type	Description	Remarks
Id	Guid	Unique identifier	
CustomerPriceGroupID	Guid	The id of the customergroup	
CustomerPriceGroupShortName	String	The shortname of the customergroup	
CustomerPriceGroupDescription	String	The description of the customergroup	
Price	Decimal?	The price	

FindReservationActivities

This service is mainly used to select a ReservationActivity from a list if one is required. An example is a combobox or list control with ReservationActivities.

Request

The request consists of the service context and a ReservationActivityCriteria object.

Name	Type	Description	Remarks
Reservation ActivityId	GUID	ID of a specific ReservationActivity. (If you mean to call up only 1 entity)	Nullable (if not applicable).
NamePattern	String	Name or part of the name that can be searched for. The system will look in the description. For partial matches, the % sign can be used. The % sign can be used before or after the word, as well as before and after the word.	Nullable (if not applicable).
Includes	ReservationActivity Includes	Determines which data are included in the response and/or which extra records will be present in the returnset.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

ReservationActivityIncludes

The ReservationActivityInclude object consists of the following properties:

Name	Type	Description	Remarks

Blocked	Boolean	Do blocked ReservationActivities also have to be visualised.	True means include.
IncludeImage	Boolean	The result will also include the related image	
IncludeImageUrl	Boolean	The result will also include the related image url	

Response

The response will contain a list of ReservationActivities that meet the criteria.

Name	Type	Description	Remarks
Id	GUID	ID of the ReservationActivity.	
Name	String	Name of the activity.	
Description	String	Description of the ReservationActivity.	
MinParticipants	Integer	Minimal number of participants for the activity.	
MaxParticipants	Integer	Maximal number of participants for the activity.	
MaxReservations	Integer	Maximal number of reservations for the activity.	
IsBlocked	Boolean	Is the ReservationActivity blocked?	
IsTechnicalActivity	Boolean	Is the ReservationActivity a technical activity?	
IsCulturalActivity	Boolean	Is the ReservationActivity a cultural activity?	
Color	Color	Colour that is used to visualise this activity within ReCreateX.	
Image	Picture	Image linked to the infrastructure	
ImageUrl	string	Image url to load the infrastructure image	

FindReservationAttachments

This method is used to find one or more reservation attachments or comments.

Request

The request consists of the service context and a ReservationAttachmentSearchCriteria object.

Name	Type	Description	Remarks
Id	Guid?	ID of a specific ReservationAttachment. (If you mean to call up only 1 entity)	Optional
Reservation Id	Guid?	ID of a reservation for which you want to retrieve the attachments/comments	Optional
EmployeeId	Guid?	ID of the employee who created/modified the attachment/comment	Optional
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	ReservationAttachmentIncludes	Determines which data are included in the response and/or which extra records will be present in the return set.	

ReservationAttachmentIncludes

The ReservationAttachmentIncludes object consists of the following properties:

Name	Type	Description	Remarks
Attachment	Boolean	The result will also include the attachments (if any)	
EmployeeName	Boolean	The result will also include the names of the related employees	

Response

The response will contain a list of ReservationAttachment objects that (inherit from the AttachmentComment object and) meet the criteria.

Name	Type	Description	Remarks

ReservationId	Guid	ID of the reservation to which the attachment/comment is linked	Mandatory
Id	Guid	ID of the reservation attachment/comment	
Description	String	Small description of the reservation attachment/comment	
Comment	String	The comment you want to add to the reservation	
Attachment	Attachment	The attachment you want to add to the reservation	
EmployeeId	Guid?	ID of the linked employee	
EmployeeName	PersonName	Name of the linked employee	
CreationDate	DateTime	Creation date of the reservation attachment/comment	
ModificationDate	DateTime	Last modification date of the reservation attachment/comment	
CreatedByEmployeeId	Guid	The employee who created this attachment	Can be an empty Guid

FindReservations

This service is mainly used to select a valid Reservation from a list if one is required. An example is a grid or list control with Reservations.

Request

The request consists of the service context and a **ReservationSearchCriteria** object.

Name	Type	Description	Remarks
ReservationId	GUID	ID of a specific reservation. (If you mean to call up only 1 entity).	Nullable (if not applicable).
CustomerId	GUID	ID of a customer of whom you want to find reservations.	Nullable (if not applicable).
PlaceId	GUID	ID of a place for which you want to find reservations.	Nullable (if not applicable).
InfrastructureId	GUID	ID of an infrastructure for which you want to find reservations.	Nullable (if not applicable).
Reservation ActivityId	GUID	ID of a ReCreateX ReservationActivity for which you want to find places.	Nullable (if not applicable).
NamePattern	String	Name or part of the name that can be searched for. The system will look in the description. For partial matches, the % sign can be used. The % sign can be used before or after the word, as well as before and after the word.	Nullable (if not applicable).
FromDateTime	DateTime	Start date and time of the period within which you want to find reservations.	
ToDateTime	DateTime	End date and time of the period within which you want to find reservations.	
Includes	ReservationIncludes	Determines which data are included in the response and/or which extra records will be present in the returnset.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

ReservationIncludes

The ReservationInclude object consists of the following properties:

Name	Type	Description	Remarks
OptionalReservations	Boolean	The result will also include optional reservations.	
SingleReservations	Boolean	The result will also include reservations that are not part of a list or series.	
ReservationsInList	Boolean	The result will also include reservations that are part of a list.	
SerieReservations	Boolean	The result will also include reservations that are part of a series.	
DeletedReservations		The result will also include deleted reservations.	

	Boolean		
CustomerInfo	Boolean	The result will also include related customer info.	
PlaceInfo	Boolean	The result will also include related place info. This only returns the main place on which the reservation took place (not the additional cumulated places – use ReservedPlaces for this)	
InfrastructureInfo	Boolean	The result will also include related infrastructure info.	
ReservationActivityInfo	Boolean	The result will also include related reservation activity info.	
ReservedPlaces	Boolean	The result will also include additional reserved places. This is applicable when using cumulated places.	
InfrastructureImage	Boolean	The result will also include the related infrastructure image	
InfrastructureImageUrl	Boolean	The result will also include the related infrastructure image url	
Contacts	Boolean	The result will also include the related contact	
SubsidyInfo	Boolean	The result will also include the related subsidy info	
ReservationActivityImage	Boolean	The result will also include the related reservation activity image	
ReservationActivityImageUrl	Boolean	The result will also include the related reservation activity image url	
PersonImage	Boolean	The result will also include the related customer/person image	
PersonImageUrl	Boolean	The result will also include the related customer/person image url	
DressingRoomAsMainReservation	Boolean	The result will include dressing room reservations as normal (main) reservations	Default TRUE
DressingRoomAsChildReservation	Boolean	The result will include dressing room reservations as child reservations of the main reservations. If TRUE, the DressingRoomReservations property of the Reservation object will be filled.	Default FALSE
EmployeeTaskIds	Boolean	The result will also include the ID of all related employee tasks.	
ReservationSales	Boolean	The result will also include the additional article sales	
RentalSales	Boolean	The result will also include the additional rental orders	

Response

The response will contain a list of Reservations that meet the criteria.

Name	Type	Description	Remarks
Id	GUID	ID of the Reservation.	
Description	String	Description of the Reservation.	
ReservationNo	Int64	Reservation number.	
SeriesNumber	Int64	Reservation series number.	
SetNumber	Int64	Reservation list number.	
OGM	String	Maximal number of days during which a reservation can be made on the Web.	
StartDateTime	DateTime	Is this place part of a collection of accumulated places?	
EndDateTime	DateTime	The Infrastructure to which this place relates.	Dependent on the Includes.
	Integer	The ReservationActivity that applies to this place.	Dependent on the Includes.

ParticipantsCount			
Amount	Decimal	Amount of the reservation.	
PostedAmount	Decimal	Posted amount of the reservation.	
RentalPrice	Decimal	Amount of rented articles linked to this reservation.	
SalesPrice	Decimal	Sales price.	
Balance	Decimal	Outstanding balance.	
HasBeenInvoiced	Boolean	Has been invoiced.	
ToPay	Boolean	Has to be paid.	
IsOption	Boolean	Is an optional reservation.	
IsDeleted	Boolean	Reservation has been deleted.	
DateLastModfied	DateTime	Date and time of last modification.	
DateDeleted	DateTime	Date and time of deletion.	
ReservationActivity	ReservationActivity	Related ReservationActivity.	Filled in dependent on the Includes.
Place	Place	Related place.	Filled in dependent on the Includes.
Infrastructure	Infrastructure	Related infrastructure.	Filled in dependent on the Includes.
Customer	Person	Related customer.	Filled in dependent on the Includes.
Deletable	Bool	Indicates if the reservation can be cancelled.	
ReservedPlaces	IList<Place>	Related reserved places. This not only contains the main place, but also alle additional reserved places (when using cumulated places).	Filled in dependent on the Includes.
DateCreated	DateTime	The date and time the reservation was created	
CardNumber	String	The card number that is associated to the reservation	
CouponDiscount	Decimal	Discount by coupon codes	
TotalAmount	Decimal	The total amount of the reservation	
Contact	Contact	The related contact	
SubsidyInfo	SubsidyInfo	The related subsidy info	
DressingRoomReservations	IList<Reservation>	Related dressing room reservations	Filled if the DressingRoomAsChildReservation in ReservationIncludes is set to TRUE
EmployeeTaskIds	IList<Guid>	List of IDs for all related employee tasks.	Filled if the EmployeeTaskIDs in ReservationIncludes is set to TRUE
ReservationArticleSales	IList<ReservationSale>	List of the additional article slaes	Filled if the ReservationSales in ReservationIncludes is set to TRUE
ReservationRentalOrders	IList<RentalOrderLine>	List of the additional rental orders	Filled if the RentalSales in ReservationIncludes is set to TRUE
PriceGroupId	Guid	The pricegroup id	Read only
SubsidyAmount	decimal	The subsidy amount	Read only
Pincode	String	The Pincode/Cardnumber	
Approved	Bool	Is an approved reservation	
DressingRoomGender	DressingRoomGenderEnum	The gender that is assigned to the dressing room	• DressingRoomGender • 0 = None • 1 = Male • 2 = Female
DressingroomTeam	DressingRoomTeamEnum	The team that is assigned to the dressing room	• DressingRoomTeam • 0 = None • 1 = Home • 2 = Away
CompetitionOwnName	string	The name of the Home team	

CompetitionCompetitorName	string	The name of the Opponent	

Place

The Place object in the FindReservations response looks as follows (Standard Place object).

Name	Type	Description	Remarks
Id	GUID	ID of the place.	
Name	String	Name of the place.	
Description	String	Description of the place.	
MaxDaysReservation	Integer	Maximal number of days during which a reservation can be made in ReCreateX.	
MaxDaysReservationWeb	Integer	Maximal number of days during which a reservation can be made on the Web.	
IsCumulated	Boolean	Is this place part of a collection of accumulated places?	
Infrastructure	Infrastructure	The Infrastructure to which this place relates.	Not filled in as nested object.
ReservationActivity	ReservationActivity	The ReservationActivity that applies to this place.	Not filled in as nested object.
AllowedReservationActivityIds	IList<GUID>	Overview of all allowed ReservationActivities for this place.	Dependent on the Includes.
AllowedRentalArticles	IList<PlaceRentalArticle>	Overview of all allowed PlaceRentalArticles for this place.	
CreateOnlineReservation	Bool	Is it allowed to create online reservations for this place?	

Infrastructure

The Infrastructure object in the FindReservations response looks as follows (Standard Infrastructure object).

Name	Type	Description	Remarks
Id	GUID	ID of the infrastructure.	
Name	String	Name of the infrastructure.	
Description	String	Description of the infrastructure.	
Location	Address	Location of the infrastructure (physical address).	
Comment	String	Remarks concerning the infrastructure.	
LastChanged	DateTime	Date of last modification.	
IsBlocked	Boolean	Is the infrastructure blocked?	
Places	IList<Place>	Not filled in as nested object.	
Division	Division	Not filled in as nested object.	
CreateOnlineReservation	Bool	Is it allowed to create online reservations for this place?	
CreateOnlineReservationFrom	DateTime	Creating online reservations is allowed from this date.	
CreateOnlineReservationTo	DateTime	Creating online reservations is allowed until this date.	
Image	Picture	Byte array of the image	
ImageUrl	String	Image url to load the infrastructure image	
MinBookingDate	DateTime	Reservation date must be after this date	
MaxBookingDate	DateTime	Reservation date must be before this date	
ShowInfrastructureInfoOnWeb	Bool	Indicates if info can be shown on web	

ReservationActivity

The ReservationActivity object in the FindReservations response looks as follows (Standard ReservationActivity object).

Name	Type	Description	Remarks
Id	GUID	ID of the ReservationActivity.	
Name	String	Name of the activity.	
Description	String	Description of the ReservationActivity.	
MinParticipants	Integer	Minimal number of participants for the activity.	
MaxParticipants	Integer	Maximal number of participants for the activity.	
MaxReservations	Integer	Maximal number of reservations for the activity.	
IsBlocked	Boolean	Is the ReservationActivity blocked?	
IsTechnicalActivity	Boolean	Is the ReservationActivity a technical activity?	
IsCulturalActivity	Boolean	Is the ReservationActivity a cultural activity?	
Image	Picture	Byte array of the image	
ImageUrl	String	Image url to load the reservation acitivity image	

Person

The Customer property in the Reservation object reflects a Person object as described in the SavePerson method.

ReservationSale

The ReservationSale object in the FindReservations response looks as follows.

Name	Type	Description	Remark
Id	Guid	The id of the additional sale	
ReservationId	Guid	The id of the linked reservation	
ArticleId	Guid	The id of the linked article	
VatId	Guid	The id of the VAT	
Quantity	int	Quantity	
Hours	TimeSpan	Hours	
CalculationType	int	The calculation type	

CalculationTypeEnum	CalculationType	The calculation type as an enum	PriceTimesHours = 0,
			FixedPrice = 1,
			PriceTimesQuantity = 2,
			PriceTimesQuantityTimesHours = 3

VatAmount	decimal	VAT	
AmountExVat	decimal	Price excl. VAT	
Amount	decimal	Price incl. VAT	
Comment	string	Additional comment	
Nights	int	Number of nights	
Days	int	Number of days	
Participants	int	How many participants	
FromDateTime	DateTime	Start date	
ToDateTime	DateTime	End date	
Sequence	int	Sequence of the sale	
EntryType	int	The entry type	
Article	Article	The linked article	
UnitPriceInc	int	The unit price incl. VAT	

FindReservationsOverviewsByDay

This service is mainly used to get a compact overview of reservations by day.
When you need more detailed information on the reservations, you will need to use the FindReservations method.

Request

The request consists of the service context and a **FindReservationsOverviewsByDayCriteria** object.

FindReservationsOverviewsByDayCriteria

Name	Type	Description	Remarks
StartTime	DateTime	Start date and time for the period to search for reservations	
EndTime	DateTime	End date and time for the period to search for reservations	
Includes	FindReservationsOverviewsByDayIncludes		

FindReservationsOverviewsByDayIncludes

The FindReservationsOverviewsByDayIncludes object consists of the following properties:

Name	Type	Description	Remarks
Optional	Boolean	The result will also include optional reservations.	

Response

The response will contain a list of **ReservationsOverviewItem** objects that meet the criteria.

ReservationsOverviewItem

Name	Type	Description	Remarks
Id	Guid	ID of the reservation	
StartDateTime	DateTime	Start date and time of the reservation	
EndDateTime	DateTime	End date and time of the reservation	
InfrastructureId	Guid	ID of the linked infrastructure	
InfrastructureName	String	Name of the linked infrastructure	
PlaceId	Guid	ID of the linked place	
PlaceName	String	Name of the linked place	
ActivityId	Guid	ID of the linked activity	
ActivityName	String	Name of the linked activity	
EmployeeTasks	IList<ReservationsOverviewItemTask>	Overview of all linked employee tasks	
HasAttachments	bool	Indiciates if the reservation has attachments linked	
CustomerName	PersonName	The name of the customer linked with the reservation	
ActivityDescription	string	The description of the linked activity	
IsOptionalReservation	bool	Indicates if the reservation is optional	

ReservationsOverviewItemTask

The ReservationsOverviewItemTask object looks as follows

Name	Type	Description	Remarks
Id	GUID	ID of the task	
EmployeeId	Guid	ID of the employee that is linked to the task	
EmployeeFirstName	String	First name of the employee that is linked to the task	
EmployeeLastName	String	Last name of the employee that is linked to the task	

FindSubsidyInfos

This service is used to search for subsidy infos.

Request

The request consists of the service context and a **SubsidyInfoSearchCriteria**.

Name	Type	Description	Remarks
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response will contain a list of **SubsidyInfo** objects that meet the criteria.

Name	Type	Description	Remarks
Id	GUID	ID of the subsidy info.	
Code	String	Code of the subsidy info.	
Description	String	Description of the subsidy info.	

ListInfrastructures

This service is mainly used to select an Infrastructure from a list if one is required. An example is a combobox or list control with Infrastructures.

Request

The request consists of the service context and an InfrastructureSearchCriteria object.

Name	Type	Description	Remarks
InfrastructureId	GUID	ID of a specific Infrastructure. (If you mean to call up only 1 entity)	Nullable (if not applicable).
AddressId	GUID	ID of a physical location in ReCreateX.	Nullable (if not applicable).
DivisionId	GUID	ID of a ReCreateX Division.	Nullable (if not applicable).
NamePattern	String	Name or part of the name that can be searched for. The system will look in the description. For partial matches, the % sign can be used. The % sign can be used before or after the word, as well as before and after the word.	Nullable (if not applicable).
Includes	InfrastructureInclude	Determines which data are included in the response and/or which extra records will be present in the returnset.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

InfrastructureIncludes

The InfrastructureInclude object consists of the following properties:

Name	Type	Description	Remarks
IncludePlaces	Boolean	The result will also include related places.	True means include.
IncludeDivisionInfo	Boolean	The result will also include related divisions.	True means include.
IncludeImage	Boolean	The result will also include the related image.	
IncludeImageUrl	Boolean	The result will also include the related image url.	
IncludeLocations	Boolean	The result will also include the related infrastructure locations.	Default = True
`IncludeAllowedReservationActivities`	Boolean	The result will include all allowed ReservationActivities on all underlying places	Only the ID is included Only applicable when IncludePlaces = True

Response

The response will contain a list of Infrastructures that meet the criteria.

Name	Type	Description	Remarks
Id	GUID	ID of the Infrastructure.	
Name	String	Name of the Infrastructure.	
Description	String	Description of the Infrastructure.	
Location	Address	Location of the infrastructure (physical address).	
Comment	String	Remarks concerning the infrastructure.	
LastChanged	DateTime	Date of last modification.	
IsBlocked	Boolean	Is the Infrastructure blocked?	
Places	IList<Place>	Places linked to the infrastructure.	
Division	Division	Division linked to the infrastructure.	
CreateOnlineReservation	Bool	Is it allowed to create online reservations for this place?	
CreateOnlineReservationFrom	DateTime	Creating online reservations is allowed from this date.	
CreateOnlineReservationTo	DateTime	Creating online reservations is allowed until this date.	
Image	Picture	Image linked to the infrastructure	
ImageUrl	string	Image url to load the infrastructure image	
ShowInfrastructureInfoOnWeb	Bool	Indicates if info can be shown on web	

Address

The Address object in the FindInfrastructures response looks as follows (Standard Address object).

Name	Type	Description	Remarks
Street	String	Street name	
Number	String	Street number	
Box	String	Box	
ZipCode	String	Zip code or postal code	
Town	String	City, municipality	
Country	String	Country/State	

Place

The Place object in the FindInfrastructures response looks as follows (Standard Place object).

Name	Type	Description	Remarks
Id	GUID	ID of the place.	
Name	String	Name of the place.	
Description	String	Description of the place.	
MaxDays Reservation	Integer	Maximal number of days to book a reservation in advance in ReCreateX.	
MaxDays ReservationWeb	Integer	Maximal number of days to book a reservation in advance on the Web.	
IsCumulated	Boolean	Is this place part of a collection of accumulated places?	
Infrastructure	Infrastructure	The Infrastructure to which this place relates. As a nested object this will be empty (NULL), unless the Place is called up on the highest level.	

Reservation Activity	Reservation Activity	The ReservationActivity that applies to this place. As a nested object this will be empty (NULL), unless the Place is called up on the highest level.	
AllowedReservationActivityIds	IList<GUID>	Overview of all allowed ReservationActivities for this place.	Dependent on the Includes.
AllowedRentalArticles	IList<PlaceRentalArticle>	Overview of all allowed PlaceRentalArticles for this place.	
CreateOnlineReservation	Bool	Is it allowed to create online reservations for this place?	
MinDays Reservation	Integer	Minimal number of days to book a reservation in advance in ReCreateX.	
MinDays ReservationWeb	Integer	Minimal number of days to book a reservation in advance on the Web.	
MaxHours Reservation	Integer	Maximal number of hours to book a reservation in advance in ReCreateX.	
MaxHours ReservationWeb	Integer	Maximal number of hours to book a reservation in advance on the Web.	
MinHours Reservation	Integer	Minimal number of hours to book a reservation in advance in ReCreateX.	
MinHours ReservationWeb	Integer	Minimal number of hours to book a reservation in advance on the Web.	

Division

The Division object in the FindInfrastructures response looks as follows (Standard Division object).

Name	Type	Description	Remarks
Id	GUID	ID of the Division.	
Name	String	Name of the Division.	
Address	Address	Address of the Division (see type 'Address').	
Phone	String	Telephone number	
Fax	String	Fax number	
Email	String	E-mail address	
Website	String	Website url	
BankAccount	String	Bank account number	
Comments	String	Remarks concerning the Division	
Credential	Credential	Credentials for the Division.	

Credentials

The Credentials object in the Division object looks as follows (Standard Credentials object).

Name	Type	Description	Remarks
UserName	String	User name	
Password	String	Password	

ListPlaces

This service is mainly used to select a valid Place from a list if one is required. An example is a combobox or list control with Places.

Request

The request consists of the service context and a PlaceSearchCriteriaobject.

Name	Type	Description	Remarks
PlaceId	GUID	ID of a specific Place. (If you mean to call up only 1 entity).	Nullable (if not applicable).

InfrastructureId	GUID	ID of an Infrastructure for which you want to find places.	Nullable (if not applicable).
ReservationActivityId	GUID	ID of a ReCreateX ReservationActivity for which you want to find places.	Nullable (if not applicable).
NamePattern	String	Name or part of the name that can be searched for. The system will look in the description. For partial matches, the % sign can be used. The % sign can be used before or after the word, as well as before and after the word.	Nullable (if not applicable).
Includes	PlaceIncludes	Determines which data are included in the response and/or which extra records will be present in the returnset.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
PersonId	GUID	ID of a specified user, used for security rights.	
PlaceIds	List<GUID>	List of IDs of places.	Nullable (if not applicable).

PlaceIncludes

The PlaceInclude object consists of the following properties:

Name	Type	Description	Remarks
InfrastructureInfo	Boolean	The result will also include related Infrastructures.	True means include.
ReservationActivityInfo	Boolean	The result will also include related ReservationActivities.	True means include.
Cumulated	Boolean	The result will also include accumulated places.	
Uncumulated	Boolean	The result will include non-accumulated places.	
AllowedReservationActivities	Boolean	The result will include all allowed ReservationActivities.	Only the ID is included.
AllowedRentalArticles	Boolean	The result will include all allowed Rental articles and its default number to rent	
IncludeImage	Boolean	The result will also include the related image	
IncludeImageUrl	Boolean	The result will also include the related image url	

Response

The response will contain a list of Places that meet the criteria.

Name	Type	Description	Remarks
Id	GUID	ID of the place.	
Name	String	Name of the place.	
Description	String	Description of the place.	
MaxDaysReservation	Integer	Maximal number of days during which a reservation can be made in ReCreateX.	
MaxDaysReservationWeb	Integer	Maximal number of days during which a reservation can be made on the Web.	
IsCumulated	Boolean	Is this place part of a collection of accumulated places?	
Infrastructure	Infrastructure	The Infrastructure to which this place relates.	Dependent on the Includes.
ReservationActivity	ReservationActivity	The *standard* ReservationActivity that applies to this place.	Dependent on the Includes.
AllowedReservationActivityIds	IList<GUID>	Overview of all allowed ReservationActivities for this place.	Dependent on the Includes.
AllowedRentalArticles	IList<PlaceRentalArticle>	Overview of all allowed PlaceRentalArticles for this place.	
CreateOnlineReservation	Bool	Is it allowed to create online reservations for this place?	
MinDaysReservation	Int	Minimum days before reservation is possible	
MinDaysReservationWeb	Int	Minimum days before reservation is possible on web	

MaxHoursReservation	Int		
MaxHoursReservationWeb	Int		
MinHoursReservation	Int	Minimum hours before reservation is possible	
MinHoursReservationWeb	Int	Minimum hours before reservation is possible on web	
ConsecutiveBookingWeb	Bool	New reservations be adjacent to previous reservations	
HasReservationRightsWeb	Bool	Has the customer rights to make a reservation in this place	
Email	String	Email of the responsible person of this place	
IsDressingRoom	Bool	Indicates if the place is a dressing room	

The **PlaceRentalArticle** object consists of the following properties:

Name	Type	Description	Remarks
RentalArticleId	GUID	ID of the corresponding rental article.	
NumberToRent	Int	Default amount of rental articles to rent when making a reservation on this place.	Can be overriden

Infrastructure

The Infrastructure object in the ListPlaces response looks as follows (Standard Infrastructure object).

Name	Type	Description	Remarks
Id	GUID	ID of the Infrastructure.	
Name	String	Name of the Infrastructure.	
Description	String	Description of the Infrastructure.	
Location	Address	Location of the infrastructure (physical address).	
Comment	String	Remarks concerning the infrastructure.	
LastChanged	DateTime	Date of last modification.	
IsBlocked	Boolean	Is the Infrastructure blocked?	
Places	IList<Place>	Not filled in as nested object.	
Division	Division	Not filled in as nested object.	
CreateOnlineReservation	Bool	Is it allowed to create online reservations for this place?	
CreateOnlineReservationFrom	DateTime	Creating online reservations is allowed from this date.	
CreateOnlineReservationTo	DateTime	Creating online reservations is allowed until this date.	
Image	Picture	Image linked to the infrastructure	
ImageUrl	String	Image url to load the infrastructure image	
MinBookingDate	DateTime	Reservation date must be after this date	
MaxBookingDate	DateTime	Reservation date must be before this date	

ReservationActivity

The ReservationActivity object in the ListPlaces response looks as follows (Standard ReservationActivity object).

Name	Type	Description	Remarks
Id	GUID	ID of the ReservationActivity.	
Name	String	Name of the activity.	
Description	String	Description of the ReservationActivity.	
MinParticipants	Integer	Minimal number of participants for the activity.	
MaxParticipants	Integer	Maximal number of participants for the activity.	
MaxReservations	Integer	Maximal number of reservations for the activity.	
IsBlocked	Boolean	Is the ReservationActivity blocked?	

IsTechnicalActivity	Boolean	Is the ReservationActivity a technical activity?	
IsCulturalActivity	Boolean	Is the ReservationActivity a cultural activity?	
Color	Color	Color used within ReCreateX to mark such an activity.	
Duration	TimeSpan	Standard duration of the activity.	
Image	Picture	Image linked to the infrastructure	
ImageUrl	string	Image url to load the reservation activity image	

Reservations – Brief overview

The Webservice provides the possibility to order facility bookings (reservations module) through your web site.
The system functions as follows:

- Person selects the activity he/she wants to book.
- Person selects the place he/she wants to book.
- Person selects the period he/she wants to book the place and activity for.
- Person optionally selects additional rental or

Detailed management of facility bookings is done in ReCreateX.
Following methods have been created for this purpose:

- FindReservationActivities
- FindInfrastructures
- FindInfrastructureOpenings
- FindPlaces
- FindReservations
- CancelReservation

SaveReservationAttachment

This method is used to save a reservation attachment/comment or both.

Request

The request consists of the service context and a ReservationAttachment object that inherits from the AttachmentComment object.

Response

The response consists of a ReservationAttachmentValidationResult object that inherits from the ValidationResult object.

Name	Type	Description	Remarks
ReservationAttachment	ReservationAttachment	The ReservationAttachment object as it was created	
IsValid	Bool	Indicates if this is a valid ValidationResult.	
Message	String	The standard error.	
BrokenRuleName	String	The code of the error message.	

UpdatePlaceReservation

This service is mainly used to update fields of a place reservation.

Request

The request consists of the service context, a **PlaceReservation**, an EmployeeId and the OptionalReservation choice.

Name	Type	Description	Remarks
PlaceReservation	PlaceReservation	PlaceReservation object	
EmployeeId	Guid	The ID of the employee	
OptionalReservation	bool	Indicates whether the updated PlaceReservation should be "In Option"	

The **PlaceReservation** inherits of **BasketItem**

Name	Type	Description	Remarks
Id	Guid	Id of the reservation	
Quantity	int	The quantity of the reservation	
Name	string		
UnitPrice	decimal	The unit price	
AdvancementPrice	decimal	The advancement price	
ReservationActivityId	Guid	The reservation activity id	
PlaceId	Guid	The place id	
StartDate	DateTime	The start date	
StartTime	TimeSpan	The start time	
EndDate	DateTime	The end date	
EndTime	TimeSpan	The end time	
Comments	String	The comments	
SerieId	Guid	The serie id	
SerieNumber	long	The serie number	
ReservationNumber	long	The reservation number	
LockTicket	LockTicket	The lock ticket	
NumberOfParticipants	int	The number of participants	
ArticleSales	IList<ArticleSales>	The article sales	
RentalReservations	Ilist<RentalReservation>	The rental reservations	
DivisionCardID	Guid?	The divsion card	
CustomerID	Guid	The customer	
ValidateSubscriptionForPlace	bool	Validate subscription for a place	
ValidateSubscriptionForActivity	bool	Validate subscription for an activity	
PriceGroupID	Guid?	The price group	
SubsidyInfoID	Guid?	The subsidy info	
EmployeeID	Guid	The employee that updates the PlaceReservation	
InfrastructureId	Guid	The infrastructure's ID of the PlaceReservation	
InfrastructureDescription	String	The infrastructure's description of the PlaceReservation	
PlaceDescription	String	The Place's description of the PlaceReservation	
ReservationActivityDescription	String	The ReservationActivity's description of the PlaceReservation	

Response

The response will contain a **ValidationResult**

ValidateSeasonPlanning

This API call exists to validate a SeasonPlanning request

Request

The request consists of the service context, and a ValidateSeasonPlanningCriteria.

ValidateSeasonPlanningCriteria

Name	Type	Description	Remarks
SeasonPlanning	SeasonPlanning	The SeasonPlanning that should be validated	

SeasonPlanning

Name	Type	Description	Remarks
ID	SeasonPlanning	The SeasonPlanning that should be validated	
CreationDate	DateTime	The creation date of the SeasonPlanning	
Status	int	The status of the SeasonPlanning	
Comment	String	The comment of the SeasonPlanning	
AddressId	Guid	The user that requested the SeasonPlanning	
RecurringPlaceReservation	RecurringPlaceReservation	The RecurringPlaceReservation that should be validated in this SeasonPlanning	

Response

The response will contain a SeasonPlanning object, just like the request, but with updated values.

CreateSeasonPlanning

This API call exists to create a SeasonPlanning request

Request

The request consists of the service context, and a CreateSeasonPlanningCriteria.

CreateSeasonPlanningCriteria

Name	Type	Description	Remarks
SeasonPlanning	SeasonPlanning	The SeasonPlanning that should be created	

SeasonPlanning

Name	Type	Description	Remarks
ID	SeasonPlanning	The SeasonPlanning that should be created	
CreationDate	DateTime	The creation date of the SeasonPlanning	
Status	int	The status of the SeasonPlanning	
Comment	String	The comment of the SeasonPlanning	
UserId	Guid	The user that requested the SeasonPlanning	
RecurringPlaceReservation	RecurringPlaceReservation	The RecurringPlaceReservation that should be created in this SeasonPlanning	

Response

The response will contain a SeasonPlanning object, just like the request, but with updated values.

ListSeasonPlannings

This API call exists to retrieve SeasonPlannings

Request

The request consists of the service context, and a ListSeasonPlanningsCriteria.

ListSeasonPlanningsCriteria

Name	Type	Description	Remarks

SeasonPlanningID	Guid?	Returns a specific SeasonPlanning	
SeasonPlanningStatus	int?	Retrieves SeasonPlannings with given status	
AddressId	Guid	Retrieves SeasonPlannings for a given user	
From	DateTime?	Retrieves SeasonPlannings that exist after a given date	
Until	DateTime	Retrieves SeasonPlannings that exist before a given date	
InfrastructureId	Guid?	Retrieves SeasonPlannings for a given Infrastructure	
PlaceId	Guid?	Retrieves SeasonPlannings for a given Place	
ActivityId	Guid?	Retrieves SeasonPlannings for a given ReservationActivity	

Response

The response will contain a ListSeasonPlanningsResult object, which contains a list of SeasonPlannings.

GetSeasonPlanningSettings

This API call exists to retrieve SeasonPlanning settings

Request

The request consists of the service context, and a GetSeasonPlanningSettingsCriteria.

There are currently no parameters in the criteria

Response

The response will contain a GetSeasonPlanningSettingsResult object.

GetSeasonPlanningSettingsResult

Name	Type	Description	Remarks
RequestColor	Color	The color for SeasonPlannings with a Request status	
ProposalColor	Color	The color for SeasonPlannings with a Proposal status	
ProposalAcceptedColor	Color	The color for SeasonPlannings with a ProposalAccepted status	
ApprovedColor	Color	The color for SeasonPlannings with a Approved status	
DeclinedColor	Color	The color for SeasonPlannings with a Declined status	
WaitingListColor	Color	The color for SeasonPlannings with a WaitingLing status	
LicenseActivated	Bool	Indicates if the SeasonPlanning license is active	

14 Bus trip module

Bus trips – Brief overview

New bus trips and bus trip series can be requested using the Webshop Webservices.
(= new requests).
Further management and planning is done in ReCreateX.
Modifications via the Webshop are only allowed for bus trips and series of which the status is 'Requested'.
As soon as a planning has been made, no more modifications can be done via the Webshop, only in ReCreateX.
Deteling and cancelling bus trips and series also depends on the trip's status: Planned, approved or outsourced bus trips can for example not be cancelled in the Webshop.
Deleting is only possible if the bus trip or the bus trips of the series have already been cancelled, rejected or still have the status Requested.
A number of Webservices have been created for this purpose:

- SaveBusTripRequest (processes both new and existing bus trips)
- SaveBusTripSerieRequest (processes both new and existing bus trip series)
- CancelBusTripRequest
- CancelBusTripSerieRequest
- DeleteBusTripRequest
- DeleteBusTripSerieRequest
  As support, a number of extra services are available.
- ListCountries (See Ticket 104674)

- ListZipcodes (See Ticket 104674)
- ListZipcodeStreets (See Ticket 104674)
- ListBustripRequests (See Ticket 104674)
- GetBusTripSerieRequest (See Ticket 104674)
  Already existing, but also used:
- FindPerson (See functionality Authentication – Persons)

CancelBusTrip

This service is used to cancel an individual bus trip (Status 'Cancellation requested').
Afterwards, and if required after a control, this status is confirmed by the back office and changed into the status 'Cancelled' and the necessary related actions are taken.
The possibility to cancel is linked to some conditions.

Request

The request consists of the CancelID (ID of the bus trip) that has to be cancelled.

Name	Type	Description	Remarks
CancelId	GUID	ID of the bus trip that has to be cancelled.	

Response

The response will contain a bus trip request (BusTripRequest) that may have been cancelled.
In addition, possible validation errors (all!) will be given.
A successful action can be recognised by:

- no validation errors (BusTripValidationResults.Count = 0)
- Status = CancelRequested

Name	Type	Description	Remarks
CancelBusTrip Result	BusTripComposite ValidationResult	Validation result + possibly cancelled bus trip.	

BusTripCompositeValidationResult

This object is used as a response container to be able to generate all validation responses and the possibly modified (status) result.

Name	Type	Description	Remarks
BusTrip Validation Results	List <BusTripValidation Result>	All failed validation results. If Count = 0 successful validation!	
BusTrip	BusTripRequest	Bus trip that has been cancelled or that does not meet the conditions to be able to cancel.	

BusTripValidationResult (=derived from ValidationResult)

This object visualises the result of a validation. The text explains what went wrong.

Name	Type	Description	Remarks
BrokenRule Name	String	Name of the validation rule.	
Message	String	Error message.	
IsValid	Bool	Rule has been successfully processed.	

BusTripRequest

See 14.6: GetBusTripRequest.

CancelBusTripSerie

This service is used to cancel all bus trips linked to a series (Status 'Cancellation requested').
Afterwards, and if required after a control, this status is confirmed by the back office and changed into the status 'Cancelled' and the necessary related actions are taken.
The possibility to cancel is linked to some conditions.

Request

The request consists of the CancelSerieID (ID of the bus trip series) that has to be cancelled.

Name	Type	Description	Remarks
CancelSerieId	GUID	ID of the bus trip series that has to be cancelled.	

Response

The response will contain a bus trip request (BusTripRequest) that may have been cancelled.
In addition, possible validation errors (all!) will be given.
A successful action can be recognised by:

- no validation errors (BusTripValidationResults.Count = 0)
- Status = CancelRequested

Name	Type	Description	Remarks
CancelBusTrip SerieResult	BusTripSerie Composition ValidationResult	Validation result + possibly cancelled bus trip series.	

BusTripSerieCompositeValidationResult

This object is used as a response container to be able to generate all validation responses and the possibly modified (status) result.

Name	Type	Description	Remarks
BusTrip Validation Results	List <BusTripValidation Result>	All failed validation results. If Count = 0 successful validation!	
BusTripSerie	BusTripSerieRequest	Bus trip series that has been cancelled or that does not meet the conditions to be able to cancel.	

BusTripValidationResult (=derived from ValidationResult)

This object visualises the result of a validation. The text explains what went wrong.

Name	Type	Description	Remarks
BrokenRule Name	String	Name of the validation rule.	
Message	String	Error message.	
IsValid	Bool	Rule has been successfully processed.	

BusTripSerieRequest

See 14.8: GetBusTripSerierequest

DeleteBusTrip

This service is used to delete an individual bus trip.
The possibility to delete is linked to some conditions.

Request

The request consists of the DeleteId (ID of the bus trip) that has to be deleted.

Name	Type	Description	Remarks
DeleteId	GUID	ID of the bus trip that has to be deleted.	

Response

Possible validation errors (all!) are visualised.
A successful action can be recognised by:

- no validation errors (BusTripValidationResults.Count = 0)
- empty Bustrip object (null)

Name	Type	Description	Remarks
DeleteBusTrip Result	BusTripComposite ValidationResult	Validation result + possibly empty bus trip.	

BusTripCompositeValidationResult

This object is used as a response container to be able to generate all validation responses.

Name	Type	Description	Remarks
BusTrip Validation Results	List <BusTripValidation Result>	All failed validation results. If Count = 0 successful validation!	
BusTrip	BusTripRequest	Bus trip that does not meet the conditions to be able to delete it or empty if bus trip has been successfully deleted.	

BusTripValidationResult (=derived from ValidationResult)

This object visualises the result of a validation. The text explains what went wrong.

Name	Type	Description	Remarks
BrokenRule Name	String	Name of the validation rule.	
Message	String	Error message.	
IsValid	Bool	Rule has been successfully processed.	

BusTripRequest

See 14.6: GetBusTripRequest.

DeleteBusTripSerie

This service is used to delete all bus trips linked to a series.
The possibility to delete is linked to conditions.

Request

The request consists of the DeleteSerieId (ID of the bus trip series) that has to be deleted.

Name	Type	Description	Remarks
DeleteSerieId	GUID	ID of the bus trip series that has to be deleted.	

Response

The response will visualise possible validation errors (all!).
A successful action can be recognised by:

- no validation errors (BusTripValidationResults.Count = 0)
- Empty BusTripSerieRequest

Name	Type	Description	Remarks
DeleteBusTrip SerieResult	BusTripSerie Composite ValidationResult	Validation result + possibly empty bus trip series.	

BusTripSerieCompositeValidationResult

This object is used as a response container to be able to generate all validation responses and the possibly modified (status) result.

Name	Type	Description	Remarks
BusTrip Validation Results	List <BusTripValidation Result>	All failed validation results. If Count = 0 successful validation!	
BusTripSerie	BusTripSerieRequest	Bus trip series that has been not been deleted or that does not meet the conditions to be able to delete it.	

BusTripValidationResult (=derived from ValidationResult)

This object visualises the result of a validation. The text explains what went wrong.

Name	Type	Description	Remarks
BrokenRule Name	String	Name of the validation rule.	
Message	String	Error message.	
IsValid	Bool	Rule has been successfully processed.	

BusTripSerieRequest

See 14.8: GetBusTripSerierequest.

FindCounties

This service is mainly used to select a valid County from a list if one is required. An example is a grid or list control with counties.

Request

The request consists of the service context and a **FindCountiesCriteria** object.

Name	Type	Description	Remarks
Id	GUID	ID of a specific county. (If you mean to call up only 1 entity)	Nullable (if not applicable).
CountryId	GUID	ID of the country of which you want to call up the counties. Each county belongs to a country.	Nullable (if not applicable).
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response will contain a list of **County** objects that meet the criteria.
The County object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the county.	
Code	String	Code of the county.	
Description	String	Description of the county.	
Country	Country	Country to which the county belongs.	See 14.2.2 for details concerning the Country object.

GetBusTripRequest

This service is used to call up an individual bus trip in the back office.

Request

The request consists of the service context and a GUID (Id) object.
Paging and search criteria are irrelevant.

Name	Type	Description	Remarks
Id	GUID	ID of the bus trip that has to be called up.	

Response

The response will contain a bus trip request (BusTripRequest) of which the identifier is the desired ID.

Name	Type	Description	Remarks
Id	GUID	ID of the zipcode.	
BusTripNo	Long (Int64)	Sequence number bus trip.	
ShortName	String	Short name.	
Description	String	Description.	
SeriesID	GUID	Series identifier.	
SeriesNo	Long	Sequence number series.	
Requestor	Person	Applicant.	

ContactPerson	Contact	Contact person.	
DepartureAddress	Address	Place of departure.	
DepartureDate Time	DateTime	Departure time.	
ArrivalAddress	Address	Place of arrival.	
ArrivalDateTime	DateTime	Arrival time.	
TotalDistance	Decimal	Total distance (there and back if round trip is checked off).	
AlsoReturn	Bool	The trip contains the trip there as well as the trip back.	
ReturnDateTime	DateTime	Return time.	
ReturnedDate Time	DateTime	Arrival of trip back.	
RequestDate	DateTime	Date and time of request.	
NumPersons	Int32	Number of passengers (without supervisors and driver).	
NumCompanions	Int32	Number of supervisors.	
ExtraLuggage	Bool	Extra luggage has to be included.	
Comments	String	Comment / Remarks.	
Confirmed	Bool	Confirmed.	
Status	BustripStates	Status of the bus trip.	
ModifiedFrom Series	Bool	This Bus trip has been modified as a result of an update of a bus trip series.	
IsTransfer	Bool	The trip is a transfer.	

Person

The Person object has already been described.
See 8.14.2: Response

Contact

The Contact object has already been described.
See 8.9:

Address

The Address object has already been described.
See 8.8.1.3:

BustripStates

Enumeration:
Requested = 0,
Planned = 1,
Rejected = 2,
Approved = 3,
CancelationRequested = 4,
Outsourced = 5,
Cancelled = 6

GetBusTripSerierequest

This service is used to call up an individual bus trip series in the back office.

Request

The request consists of the service context and a GUID (SerieId) object.
Paging and search criteria are irrelevant.

Name	Type	Description	Remarks
Id	GUID	**SerieId** of the bus trip series that has to be called up.	

Response

The response will contain a bus trip series request (BusTripSerieRequest) of which the Series identifier is the desired ID.

Name	Type	Description	Remarks
Id	GUID	ID of the zipcode.	
SerieId	GUID	Series ID of the series.	
SerieNo	Long	Series sequence number of the series.	
StartDate	DateTime	Start date of the serial trips of the series.	
FromHour	DateTime	Departure time of all bus trips of the series.	
ToHour	DateTime	Arrival time of all bus trips of the series.	
EndDate	DateTime	End date of the series of bus trips.	
Iterations	Int32	Number of iterations (bus trips) linked to the series.	
FrequencyType	FrequencyTypes	With which frequency do the bus trips have to be executed.	
Frequency	SerieFrequency	Daily, weekly and monthly frequency parameters.	
WeekPattern	List <WeekDayRange>	Week pattern (pattern for each day).	
MonthPattern	SerieMonthPattern	Month pattern (pattern for monthly bus trips).	
YearPattern	SerieYearPattern	Year pattern (pattern for yearly bus trips).	
Dates	List<DateTime>	List of dates on which the bus trips are executed.	
Bustrips	List<BusTripRequest>	The list of all bus trips that have been linked to the series.	

WeekDayRange

A WeekDayRange is used to set up the parameters in case of weekly frequency.

Name	Type	Description	Remarks
Weekdays	WeekDays	Day of the week parameters.	
Include	Bool	This day of the week has to be included in the series range.	
From	DateTime	Exception to the regular departure and arrival hours (From component).	
To	DateTime	Exception to the regular departure and arrival hours (To component).	

SerieMonthPattern

A SerieMonthPattern is used to set up the month parameters of the Series in case of monthly frequency.

Name	Type	Description	Remarks
DayOfTheWeek	WeekDays	Day of the week parameters.	
DayOfTheMonth	Int32	Day of the month parameters.	

SerieYearPattern

A YearMonthPattern is used to set up the year parameters of the Series in case of yearly frequency.

Name	Type	Description	Remarks
DayOfTheWeek	WeekDays	Day of the week parameters.	
DayOfTheMonth	Int32	Day of the month parameters.	
MonthNumber	Int32	Number of the month.	

SerieFrequency

A SerieFrequency is a combination object that keeps track of the frequency of the different types of patterns for the calculation of the bus trips of the Series.

Name	Type	Description	Remarks
DayFrequency	Int32	Frequency for day patterns.	
WeekFrequency	Int32	Frequency for week patterns.	
Month Frequency	Int32	Frequency for month patterns.	

WeekDays

Enumeration that describes the days of the week, with only relevant parameters, however.
Dummy = 0, (Data contract serialization requirement – not used)
Undefined = -1,(Special meaning: weekday is not relevant)
Monday = 1,
Tuesday = 2,
Wednesday = 3,
Thursday = 4,
Friday = 5,
Saturday = 6,
Sunday = 7

FrequencyTypes

Enumeration that indicates the frequency types.
Daily = 0,
Weekly = 1,
Monthly = 2,
Yearly = 3.

BusTripRequest

See 14.6.2: Response.

ListBusTripRequests

This service is used to call up a list of bus trips in the back office. Not to be confused with the bus trips that are linked to a specific series.

Request

The request consists of the service context and a BusTripRequestSearchCriteria object.

Name	Type	Description	Remarks
Id	GUID?	ID of a specific bus trip.	Nullable!
RequestorId	GUID?	ReCreateX ID of the applicant.	
NamePattern	String	Name pattern for the description of the bus trip.	
BusTripNo	Long (Int64)	Sequence number of the bus trip.	
SeriesID	GUID?	Series ID of a possible series to which bus trips could be linked.	Nullable!
DepartureFrom DateTime	DateTime?	Criterion Departure date/time from.	
DepartureTo DateTime	DateTime?	Criterion Departure date/time till.	
ArrivalFrom DateTime	DateTime?	Criterion Arrival date/time from.	
ArrivalToDateTime	DateTime?	Criterion Arrival date/time till.	
ReturnFrom DateTime	DateTime?	Criterion Return date/time from.	
ReturnToDateTime	DateTime?	Criterion Return date/time till.	
RequestFrom DateTime	DateTime?	Criterion Request date/time from.	
RequestTo DateTime	DateTime?	Criterion Request date/time till.	
MinimumNumberOfPersons	Int32	Minimal number of passengers.	
MaximumNumberOfPersons	Int32	Maximal number of passengers.	
Includes	BusTripIncludes	Include parameters for the request (which extra objects have to be included in the response?).	
Paging	PagingCriteria	See 4.2: Paging Criteria	

BusTripIncludes

This object is used to indicate which optional related objects have to be included in the response.

Name	Type	Description	Remarks
Contacts	Bool	Include contact persons.	
Persons	Bool	Include applicants.	
ConfirmedBustrips	Bool	Only include confirmed bus trips.	Combine both to show all.

UnConfirmed Bustrips	Bool	Only include unconfirmed bus trips.	
ExtraLugage	Bool	Only include bus trips for which extra luggage has been registered.	Combine both to show all.
NoExtraLugage	Bool	Only include bus trips for which no extra luggage has been registered.	

Response

The response will contain a list of bus trip requests (BusTripRequests) that meet the search criteria.

BusTripRequest

See 14.6.2: Response

ListCountries

This service is mainly used to select a valid Country from a list if one is required. An example is a grid or list control with countries.

Request

The request consists of the Service Context and a ListCountriesCriteria object.

Name	Type	Description	Remarks
Paging	PagingCriteria	See 4.2: Paging Criteria	

Includes

Considering the brief object 'Country', this service holds no Includes functionality.

Response

The response will contain a list of countries (List<Country>).

Name	Type	Description	Remarks
Id	GUID	ID of the Reservation.	
Code	String	2 Letter ReCreateX country code.	
Name	String	Full name of the country.	
CountryGroupId	Guid	ID of the CountryGroup	

ListZipcodeCendrisStreets

This service is used to search address details on the basis of the **zipcode and house number.** This is the most commonly used searched method in the Netherlands.

Request

The request consists of the service context and a **ZipcodeCendrisStreetsSearchCriteria** object.
This criteria object allows to search on (partial) zipcode and house number.

Name	Type	Description	Remarks
ZipcodePattern	String	Zipcode with which a match has to be found.	In the Netherlands queries are always on the basis of these 2 criteria.
Number	Int?	House number with which a match has to be found.	
Includes	ZipcodeCendrisStreetIncludes	Include parameters for the request (which extra objects have to be included in the response?).	
Paging	PagingCriteria	See 4.2: Paging Criteria	

The ZipcodeCendrisStreetIncludes object has only one parameter.

Name	Type	Description	Remarks
Zipcodes	Bool	Do the zipcodes have to be filled in completely in the response? • Yes: All info will be available (street / city / zipcode / country).	

		- No: Only street and zipcode ID will be shown.	

Response

The response will contain a list of ZipcodeCendrisStreet that meet the criteria.

Name	Type	Description	Remarks
Id	GUID	Unique ID.	
Street	String	Street name.	
ZipcodeId	GUID	ID of the zipcode.	
ZipCodes	IList<Zipcode>	Detailed information concerning the zipcodes found (city - zipcode - country).	Dependent on the Zipcode Include setting of the demand.

ListZipcodes

This service is used to be able to find a postal code in case an Internal ID (GUID) is required with a link to the ReCreateX database.
A valid postal code can be selected from a list, searching one is possible on the basis of several criteria.
An example is the creation of a combobox or a list box with postal codes in which the list contains matches that correspond to the typed letters. In order to also use partial matches in a query, a '%' sign can be used after (and before) the typed letters. These serve as wildcards.
Without wildcards the system will try to find an exact match.

Request

The request consists of the service context and a ZipcodesSearchCriteria object.
In addition to the paging criteria to be able to take care of paging of long lists, this criterion also contains an ID allowing to call up individual Zipcodes on the basis of their GUID.
Moreover, matches can be found of (partial) zipcode and (partial) name.

Name	Type	Description	Remarks
Id	GUID	ID of the Zipcode.	
ZipcodePattern	String	Zipcode pattern with which a match has to be found (do not forget the % wildcard if partial matches also have to be taken into account.)	
CityNamePattern	String	Name pattern with which a match has to be found (do not forget the % wildcard if partial matches also have to be taken into account.)	
Street	String	Search on street name.	Used for queries for Dutch postal codes.
Number	Int?	Search on house number.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

Includes

Considering the brief object 'Zipcode', this service holds no Includes functionality.

Response

The response will contain a list of Zipcodes (List<Zipcode>) that meet the criteria.

Name	Type	Description	Remarks
Id	GUID	ID of the zipcode.	
Zip	String	Zipcode.	
City	String	Name of the city/municipality.	
Country	Country	Country in which the municipality is located.	

Country

The Country object has already been described.
See 14.2.2: Response

ListZipcodeStreets

This service is used to find streets that are part of a certain postal code.
Streets can be found using the criteria: zipcode and name pattern.
An example is the creation of a combobox or a list box with street names in which the list contains matches that correspond to the typed letters.
In order to also use partial matches in a query, a '%' sign can be used after (and before) the typed letters. These serve as wildcards.

Without wildcards the system will try to find an exact match.

Request

The request consists of the service context and a ZipcodeStreetsSearchCriteria object.
In addition to the paging criteria, this criterion also contains an ID allowing to call up individual streets on the basis of their GUID.
Moreover, matches can be found on Zipcode ID and (partial) name.

Name	Type	Description	Remarks
Id	GUID	ID of the Zipcode.	
ZipcodeId	String	ZipcodeID with which a match has to be found. (Limits the list of streets to those with the given Zipcode).	
StreetNamePattern	String	Zipcode pattern with which a match has to be found (do not forget the % wildcard if partial matches also have to be taken into account.)	
IncludeZipcodes	Bool	Do the different zipcodes to which the street belongs have to be included in the response.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
Zipcode	String	The zipcode (in text format) with which a match has to be found.	

Response

The response will contain a list of ZipcodeStreets (List<ZipcodeStreet>) that meet the criteria.

Name	Type	Description	Remarks
Id	GUID	ID of the zipcode.	
StreetName	String	Street name.	
City	String	Name of the city/municipality.	
ZipCodes	IList<Zipcode>	List of the zipcodes of the municipalities in which the street is located.	Dependent on the IncludeZipcodes setting of the demand.

Zipcode

The Zipcode object has already been described.
See 14.3.2: Response

SaveBusTripRequest

This service is used to create an individual bus trip or to save an existing one in the back office.

Request

The request consists of the BusTripRequest that has to be saved.
This can be a new bus trip (ID and BusTripNo are still default values) or an existing bus trip (values are already known). The service will automatically find out which of the two cases it is. There are, however, conditions to be able to save an existing bus trip...

Name	Type	Description	Remarks
BusTrip	BusTripRequest	Bus trip that has to be saved.	

Response

The response will contain a bus trip request (BusTripRequest), possibly updated with the saved results (in case of a new bus trip -> default values will become effective values).
In addition, possible validation errors (all!) will be given.

Name	Type	Description	Remarks
SaveBusTrip Result	BusTrip Composition ValidationResult	Validation result + possibly updated bus trip.	

BusTripCompositeValidationResult

This object is used as a response container to be able to generate all validation responses and the possibly saved result (and possible modifications such as an existing ID or bus trip sequence number).

Name	Type	Description	Remarks
BusTrip Validation Results	List <BusTripValidation Result>	All failed validation results. If Count = 0 successful validation!	
BusTrip	BusTripRequest	Bus trip that is saved or has errors.	

BusTripValidationResult (=derived from ValidationResult)

In this object the result of a validation is used. The text explains what went wrong.

Name	Type	Description	Remarks
BrokenRule Name	String	Name of the validation rule.	
Message	String	Error message.	
IsValid	Bool	Rule has been successfully processed.	

BusTripRequest

See 14.6.2: Response.

SaveBusTripSERIERequest

This service is used to create an individual bus trip series or to save an existing one in the back office.

Request

The request consists of the BusTripSerieRequest that has to be saved.
This can be a new series (ID, SerieID and SerieNo are still default values) or an existing series (values are already known). The service will automatically find out which of the two cases it is.
There are, however, conditions to be able to save an existing bus trip series...

Name	Type	Description	Remarks
BusTrip	BusTripSerie Request	Bus trip series that has to be saved.	

Response

The response will contain a bus trip series request (BusTripSerieRequest), possibly updated with the saved results (in case of a new bus trip series -> default values will become effective values).
In addition, possible validation errors (all!) will be given.

Name	Type	Description	Remarks
SaveBusTripSerie Result	BusTripSerie Composite ValidationResult	Validation result + possibly updated bus trip series.	

BusTripCompositeValidationResult

This object is used as a response container to be able to generate all validation responses and the possibly saved result (and possible modifications such as an existing ID or bus trip sequence number).

Name	Type	Description	Remarks
BusTrip Validation Results	List <BusTripValidation Result>	All failed validation results. If Count = 0 successful validation!	
BusTripSerie	BusTripSerieRequest	Bus trip series that is saved or has errors.	

BusTripValidationResult (=derived from ValidationResult)

In this object the result of a validation is used. The text explains what went wrong.

Name	Type	Description	Remarks
BrokenRule Name	String	Name of the validation rule.	

Message	String	Error message.	
IsValid	Bool	Rule has been successfully processed.	

BusTripSerieRequest

See 14.8.2: Response.

15 Employees

AuthenticateEmployee

AuthenticateEmployee is used to validate the credentials of an employee if he/she wants to log in.

Request

Within the AuthenticateEmployeeRequest, only a **Credential** object is expected in addition to the service context object.

Name	Type	Description
Credentials	Credential	Contains the user name and password combination one wants to authenticate. See amongst others 8.3.1

Response

AuthenticateEmployeeResponse contains an **AuthenticationResult** that indicates if authentication succeeded or not.

Name	Type	Description	Remarks
Error	AuthenticationError	An enumeration that returns the error of the AuthenticationProcess. Possible values are: • None • InvalidParameters • UnknownUser • BlockedUser • InvalidCredential	For security purposes no correct information is given here.
HasSucceeded	Boolean	Indicates if the user has been successfully authenticated or not.	
PersonId	Guid	The ID of the authenticated user.	Filled in if authentication was successful
LastLoginDate	DateTime?	The date/time of the last login date	Not applicable in this method
PersonCardId	Guid	The ID of the authenticated user card.	Not applicable in this method
Person	Person	The authenticated user.	Not applicable in this method
ContactId	Guid?	The ID of the authenticated contact.	Not applicable in this method
Employee	Employee	The ID of the authenticated employee	Filled in if authentication was successful

DeleteEmployeeTaskAttachment

This method is used to delete an employee task attachment/comment.

Request

The request consists of the service context and a Guid representing the unique identifier of the EmployeeTaskAttachment object.

Response

The response consists of a ValidationResult object that indicates whether or not the delete was successful.

Name	Type	Description	Remarks
IsValid	Bool	Indicates if this is a valid ValidationResult.	
Message	String	The standard error.	
BrokenRuleName	String	The code of the error message.	

Employees – Brief overview

Within the purchase module employees are used to indicate for which objects a user has rights.
The following functions allow you to call up employees and allow employees to authenticate via the Webservices, so that you can enter the ID of the desired employee in the various criteria objects.
FindEmployees: Is used to find employees.
AuthenticateEmployee: Is used to authenticate an employee on the basis of credentials.

FindEmployeeNotifications

This method is used to find the notifications for a certain employee.

Should only be used within the TaskFlow App.

Request

The request consists of the service context and an EmployeeNotificationSearchCriteria object.

EmployeeNotificationSearchCriteria

Name	Type	Description	Remarks
EmployeeId	Guid	The ID of the employee you wish to load notifications for	
From	DateTime	Filter	
Until	DateTime	Filter	
Includes	EmployeeNotificationIncludes	Include object	

EmployeeNotificationIncludes

Name	Type	Description	Remarks
Task	bool	Do you want to load the task notifications?	
TaskAttachment	bool	Do you want to load the task attachment notifications?	
Reservation	bool	Do you want to load the reservation notifications?	
ReservationAttachment	bool	Do you want to load the reservation attachment notifications?	

Response

The response will contain a list of EmployeeNotification objects that meet the criteria.

There are different kinds of notifications and the model EmployeeNotification is an abstract class.

All different kinds of notifications inherits from EmployeeNotification.

EmployeeNotification

Name	Type	Description	Remarks
Id	Guid	The ID of the notification	
NotificationDate	DateTime	The date when the notification has been created	
CreatedBy	TaskFlowEmployee	The employee who created the specific object	

TaskFlowEmployee

Name	Type	Description	Remarks
Id	Guid	ID of the employee	
Name	PersonName	Name object which contains the full name of the employee	
Email	string	Email of the employee	

PictureUri	string	Picture uri of the employee	

EmployeeTaskNotification

These notications are generated when a new employee task has been created, when the employee task status has been modified or when the employee task has been cancelled.

Name	Type	Description	Remarks
TaskId	Guid	Id of the linked task	
TaskCreationDate	DateTime	Creation date of the linked task	
WorkType	WorkType	The work type	
From	DateTime	Start date of the linked task	
Until	DateTime	End date of the linked task	
Status	EmployeeTaskStatus	The status of the linked task	

Type	EmployeeTaskNotificationType	Type notification	NewTask = 0,
			StatusModified = 1,
			Cancelled = 2,
			EmployeeAssigned = 3,
			WorkTypeModified = 4,
			DateTimeModified = 5,
			DescriptionModified= 6

EmployeeTaskAttachmentNotification

These notifications are generated when a new attachment/comment has been created on a certain employee task.

Name	Type	Description	Remarks
TaskAttachmentId	Guid	The id of the linked attachment	
TaskId	Guid	The id of the linked task	
TaskAttachmentCreationDate	DateTime	Creation date of the linked attachment	
TaskWorkType	WorkType	The work type	
ModificationDate	DateTime	Last modification date of the linked attachment	Can be empty when the attachment has never been modified

Type	EmployeeTaskAttachmentNotificationType	Type notification	NewAttachment = 0,
			AttachmentModified = 1

EmployeeReservationNotification

These notifications are generated when a new reservation (linked with an employee task) is created or when the dates of an existing reservation are updated.

Name	Type	Description	Remarks
ReservationId	Guid	The id of the linked reservation	
ReservationName	string	The name/description of the linked reservation	
ReservationCreationDate	DateTime	The creation date of the linked reservation	
From	DateTime	The start date of the linked reservation	
Till	DateTime	The end date of the linked reservation	

EmployeeReservationNotificationType	Type	Type notification	NewBooking = 0,
			TimeChanged = 1,

			ActivityModified = 2,
			PlaceModified = 3,
			Cancelled = 4,
			DescriptionModified = 5

PlaceId	Guid	The id of the linked place	
PlaceName	string	The name of the linked place	
PlaceDescription	string	The description of the linked place	
InfrastructureId	Guid	The id of the linked infrastructure	
InfrastructureName	string	The name of the linked infrastructure	
InfrastructureDescription	string	The description of the linked infrastructure	
CustomerName	PersonName	The name of the customer	
CustomerPictureUri	string	The picture uri of the customer	
CustomerId	Guid	The ID of the customer	
ReservationIsOptional	bool	Is true when the reservation is optional	

EmployeeReservationAttachmentNotification

These notificatons are generated when a new attachment/comment is created on a reservation (linked with an employee task)

Name	Type	Description	Remarks
ReservationAttachmentId	Guid	The id of the linked reservation attachment	
ReservationId	Guid	The id of the linked reservation	
ReservationName	string	The name/description of the linked reservation	
ReservationAttachmentCreationDate	DateTime	The creation date of the linked reservation attachment	

Type	EmployeeReservationAttachmentNotificationType	Type notification	NewAttachment = 0,
			AttachmentModified = 1

ModificationDate	DateTime	Last modification date of the linked attachment	Can be empty when the attachment has never been modified
PlaceId	Guid	The id of the linked place	
PlaceName	string	The name of the linked place	
PlaceDescription	string	The description of the linked place	
InfrastructureId	Guid	The id of the linked infrastructure	
InfrastructureName	string	The name of the linked infrastructure	
InfrastructureDescription	string	The description of the linked infrastructure	
ReservationIsOptional	bool	Is true when the reservation is optional	

FindEmployees

Is used to call up a list of employees.

Request

The FindEmployeesRequest contains on the one hand the standard service context object and on the other hand the **EmployeeSearchCriteria** object.

Name	Type	Description	Remarks
Id	GUID?	If filled in: The ID of the employee one wants to find.	Nullable
UserName	String	If filled in: The Web user name of the employee one wants to find.	

Email	String	If filled in: The e-mail address of the employee one wants to find.	
IsBlocked	Bool?	If filled in: True to call up only the blocked employees or False to call up only the the employees that are not blocked.	Nullable
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	EmployeeIncludes	Determines which parts of the employee will be included.	
LessonGroupId	GUID	Only employees that are registered as instructor of that specific lesson group will be called up.	
WorkTypeId	Guid	Filter on work type id	
SkillSubCategoryIds	IList<Guid>	Filter on skills	
AvailableFrom	DateTime?	Filter on the availability of an employee	If you want to use this, also fill in the 'AvailableUntil' property
AvailableUntil	DateTime?	Filter on the availability of an employee	If you want to use this, also fill in the 'AvailableFrom' property
TaskId	Guid?	Filter on TaskId	Nullable

EmployeeIncludes

This object allows to exclude certain parts of an employee.

Name	Type	Description	Remarks
IncludeImage	Bool?	Calls up the image of the employee or not.	
IncludeAllowedInfrastructures	Bool?	Adds allowed infrastructures to the response	
IncludeEmployeeSkills	Bool?	Adds the skills the employee has	

Response

The FindEmployeeResponse object contains a list of Employee objects that meet the criteria above.

Name	Type	Description	Remarks
Id	GUID	The ID of the employee.	
Code	String	The code of the employee.	
Name	PersonName	The name of the employee.	
BirthDate	DateTime?	The birth date of the employee.	Nullable
Gender	Gender	The gender of the employee.	
Language	String	The ISO language code of the employee.	
Address	Address	The city of the employee.	
Phone1	String	The telephone number of the employee.	
Phone2	String		
Phone3	String		
Email	String	The e-mail address of the employee.	
Comments	String	Free remarks field containing extra information concerning the employee.	
Department	Department	The department to which the employee belongs. See 15.2.3: Department	
Function	Function	The function of the employee within the organisation. See 15.2.4: Function	
Credential	Credential	The user name and password combination of the employee. See amongst others 8.3.1	
IsBlocked	Bool	True if the employee is blocked.	
Picture	Image	Picture of the employee.	

PictureUrl	String	Link to the picture of the employee. When no picture is available, the value is null.	
Organisation	PersonBase	The organisation which is linked to the employee	
TaskflowShowCustomerDetails	bool	Defines if the customer can see customer details (in taskflow)	
TaskflowHidePrices	bool	Defines if the customer can see prices (in taskflow)	
TaskflowUpdateParticipantCount	bool	Defines if the employee is allowed to update the participant count	
TaskflowHideMyTasks	bool	Defines if the employee should see their own tasks	
TaskflowAllowedInfrastructures	List<Infrastructure>	Defines what infrastructures the employee has access to	
SendEmailOnOrder	bool	Defines whether an email should be sent to an Employee when an order is done	
AllowToAddJournals	bool	Defines whether an employee is allowed to add journals	
AllowToCloseJournals	bool	Defines whether an employee is allowed to close journals	
IsSupervisor	bool	Defines wether an employee is a supervisor	
EmployeeSkills	List<EmployeeSkill>	Defines the skills the employee has	

Department

Contains the definition of the employee's department.

Name	Type	Description	Remarks
Id	GUID	The ID of the department.	
Code	String	The code of the department.	
Name	String	The name of the department.	

Function

Contains the definition of the employee's function. This is generally the task description of the employee within the organisation.

Name	Type	Description	Remarks
Id	GUID	The ID of the function.	
Code	String	The code of the function.	
Name	String	The name of the function.	

EmployeeSkills

Contains the definition of the employee's skills.

Name	Type	Description	Remarks
Id	Guid	The ID of the EmployeeSkill	
EmployeeSkillObtained	DateTime?	The date the employee got the skill	
EmployeeSkillExpires	DateTime?	The date the skill of the employee expires	
EmployeeSkillComment	string	The comment on the employee skill	
SkillSubCategoryID	Guid	The ID of the skill sub category	
SkillSubCategoryShortName	string	The shortname for the sub category	
SkillSubCategoryDescription	string	The description of the sub category	
SkillSubCategoryLevel	int	The level of the skill sub category	
SkillCategoryID	Guid	The ID of the SkillCategory	
SkillCategoryShortName	string	The shortname for the skill category	
SkillCategoryDescription	string	The description of the skill category	
SkillCategoryMandatory	bool	Determines if the skill category is mandatory	
SkillsID	Guid	The ID for the skill	

SkillsShortName	string	The shortname for the skill	
SkillsDescription	string	The description of the skill	

FindEmployeeTaskAttachments

This method is used to find one or more employee task attachments or comments.

Request

The request consists of the service context and an EmployeeTaskAttachmentSearchCriteria object.

EmployeeTaskAttachmentSearchCriteria

Name	Type	Description	Remarks
Id	Guid?	ID of a specific EmployeeTaskAttachment. (If you mean to call up only 1 entity)	Optional
EmployeeTaskId	Guid?	ID of an employee task for which you want to retrieve the attachments/comments	Optional
EmployeeId	Guid?	ID of the employee who created/modified the attachment/comment	Optional
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	EmployeeTaskAttachmentIncludes	Determines which data are included in the response and/or which extra records will be present in the return set.	

EmployeeTaskAttachmentIncludes

The EmployeeTaskAttachmentIncludes object consists of the following properties:

Name	Type	Description	Remarks
Attachment	Boolean	The result will also include the attachments (if any)	
EmployeeName	Boolean	The result will also include the names of the related employees	
AttachmentByteArray	Boolean	The result will also includes the byte array (EmployeeTaskAttachment.Attachment)	

Response

The response will contain a list of EmployeeTaskAttachment objects that (inherit from the AttachmentComment object and) meet the criteria.

EmployeeTaskAttachment

Name	Type	Description	Remarks
EmployeeTaskId	Guid	ID of the employee task to which the attachment/comment is linked	Mandatory
Id	Guid	ID of the employee task attachment/comment	
Description	String	Small description of the employee task attachment/comment	
Comment	String	The comment you want to add to the employee task	
Attachment	Attachment	The attachment you want to add to the employee task	
EmployeeId	Guid?	ID of the linked employee	
EmployeeName	PersonName	Name of the linked employee	
CreationDate	DateTime	Creation date of the employee task attachment/comment	
ModificationDate	DateTime	Last modification date of the employee task attachment/comment	
CreatedByEmployeeId	Guid	The employee who created this attachment	Can be an empty guid

FindEmployeeTasks

This method is used to find one or more employee tasks.

Request

The request consists of the service context and a EmployeeTaskSearchCriteria object.

EmployeeTaskSearchCriteria

Name	Type	Description	Remarks
Id	Guid?	ID of a specific EmployeeTask. (If you mean to call up only 1 entity)	
EmployeeId	Guid?	ID of the employee that is linked to the task.	
From	DateTime?	Find tasks which are scheduled from a certain date.	
Until	DateTime?	Find tasks which are scheduled until a certain date.	
Type	EmployeeTaskType?	Find tasks of a certain type.	
SubjectId	Guid?	ID of the subject that is linked to the task	*Ex: ID of the linked reservation when task was created from a reservation*
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	EmployeeTaskIncludes	Determines which data are included in the response and/or which extra records will be present in the return set.	

EmployeeTaskIncludes

The EmployeeTaskIncludes object consists of the following properties:

Name	Type	Description	Remarks
EmployeeName	Boolean	Include the names of the employee that is linked to the task?	
Language	Boolean	Include the language that is linked to the task?	
WorkGroup	Boolean	Include the work group that is related to the task?	Only applicable when WorkType=TRUE
WorkType	Boolean	Include the work type that is related to the task?	
Comment	Boolean	Include the all task comments?	
Attachment	Boolean	Include the attachments within the comments?	Only applicable when Comment=TRUE
Cancelled	Boolean	Include the cancelled comments?	

AssignType	EmployeeTaskAssignType	Filter on which assigned tasks you want to load	Both = 0,
			Assigned = 1,
			Unassigned = 2

Skills	Boolean	Include the required skills	
Availability	Boolean	Includes the availability for the given employee to the task	

EmployeeTaskType

The EmployeeTaskType enumaration defines to which object the task is related.
Following types are currently supported:

Value	Description
Reservation	The employee task was created from a reservation (= booking)
ActivityDay	The employee task was created from an activity day
OrganisedVisit	The employee task was created from an organised visit (= exposition visit)
SubFile	The employee task was created from a sub file
LooseTask	The employee task was losely created, so is not linked to any other object
CultureEvent	The employee task was created from a culture event

BusPlanning	The employee task was created from a bus planning event
RentalHeader	The employee task was created from a rental header
AddressFollowUp	The employee task was created from an address follow up
FileActivityPeriod	The employee task was created from a file activity period
ExpositionPeriod	The employee task was created from an exposition period

Response

The response will contain a list of EmployeeTask objects that meet the criteria.

EmployeeTask

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the employee task	
From	DateTime	Start date and time of the employee task	
Until	DateTime	End date and time of the employee task	
Status	EmployeeTaskStatus	The status of the employee task	
Comment	String	General comment for the task	
EmployeeId	Guid?	The employee that is linked to the task	
EmployeeName	PersonName	The name of the employee that is linked to the task	Only filled when EmployeeName=TRUE in EmployeeTaskIncludes
Language	Language	The language that is linked to the task	Only filled when Language=TRUE in EmployeeTaskIncludes
WorkType	WorkType	The work type of the employee task	Only filled when WorkType=TRUE in EmployeeTaskIncludes
Type	EmployeeTaskType	The type of the employee task	
SubjectId	Guid?	The unique identifier of the related object	
Attachments	IList<EmployeeTaskAttachment>	The attachments and comments that are linked to the task	Only filled when Comments=TRUE in EmployeeTaskIncludes
ActivityName	string	The activity name	
ActivityCode	string	The activity code	
RentalNo	long	The rental number	
NeededSkillIds	IList<Guid>	Required skills	
ExpositionId	Guid?	The linked exposition id	Can be null
ExpositionPeriodId	Guid?	The linked expsoition period id	Can be null
IsAvailable	bool?	Tells you wether or not the given employee is available for the task	

The EmployeeTaskStatus enumaration defines the current status of the task:

Value	Description
ToBeConfirmed	The task has not been confirmed or refused
Confirmed	The task has been confirmed
Refused	The task has been refused
Executed	The task has been executed

FindWorkTypes

Is used to call up a list of worktypes.

Request

The FindWorkTypesRequest contains the standard service context object.

Response

The FindEmployeeResponse object contains a list of Employee objects that meet the criteria above.

Name	Type	Description	Remarks
Id	Guid	The ID of the worktype.	
Code	string	The code of the worktype.	
Name	string	The name of the worktype.	
WorkGroup	WorkGroups	The linked workgroup	
Employee	Employees	The configured employees on a specific worktype	

SaveEmployee

This method is used to save an employee.

Request

The request consists of the service context and an Employee object.

```
{
  "Employee": {
    "Id": "00000000-0000-0000-0000-000000000000",
    "Code": "Beeple",
    "Name": {
      "First": "John",
      "Last": "Doe",
      "Middle": "randomString"
    },
    "BirthDate": "1985-01-01",
    "Gender": 0,
    "Language": "NL",
    "Address": {
      "Box": "1A",
      "Country": "BE",
      "Number": "25",
      "Street": "randomStreet",
      "Town": "Brussel",
      "ZipCode": 1000
    },
    "Phone1": "050 11 22 33",
    "Phone2": "050 22 33 11",
    "Phone3": "050 33 11 22",
    "Email": "null@example.com",
    "Comments": "hello world",
    "Credential": {
      "Username": "epic_username"
    },
    "IsBlocked": false,
    "Email2": "null@anotherexample.com"
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000700",
    "Password": "*****"
  }
}
```

Response

The response will contain a SaveEmployeeResult object

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
SaveEmployeeResponse, ReCreateX.WebShop.WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.WebServices.Contracts.
SaveEmployeeResult, ReCreateX.WebShop.WebServices.Contracts",
        "employee": {
            "$type": "ReCreateX.WebShop.WebServices.Contracts.Employee,
ReCreateX.WebShop.WebServices.Contracts",
            "id": "6b90bc9f-e6bb-e811-ab35-d46d6d455a91",
            "code": "Beeple",
            "name": {
                "$type": "ReCreateX.WebShop.WebServices.Contracts.
PersonName, ReCreateX.WebShop.WebServices.Contracts",
                "first": "John",
                "last": "Doe",
                "middle": "randomString",
                "initials": null
            },
            "birthDate": "1985-01-01T00:00:00",
            "gender": 0,
            "language": "NL",
            "address": {
                "$type": "ReCreateX.WebShop.WebServices.Contracts.
Address, ReCreateX.WebShop.WebServices.Contracts",
                "box": "1A",
                "country": "BE",
                "number": "25",
                "street": "randomStreet",
                "town": "Brussel",
                "zipCode": "1000",
                "locationName": null,
                "countryDescription": null
            },
            "phone1": "050 11 22 33",
            "phone2": "050 22 33 11",
            "phone3": "050 33 11 22",
            "email": "null@example.com",
            "comments": "hello world",
            "department": null,
            "function": null,
            "credential": {
                "$type": "ReCreateX.WebShop.WebServices.Contracts.
Credential, ReCreateX.WebShop.WebServices.Contracts",
                "password": null,
                "username": "epic_username"
            },
            "isBlocked": false,
            "picture": null,
            "allowLessonTransfers": false,
            "onlyReadLessons": false,
            "allowChangeLessongroupInfo": false,
            "onlyShowMyLessongroups": false,
            "email2": "null@anotherexample.com",
            "pictureUrl": null,
            "organisation": null
        },
        "validationResult": {
            "$type": "ReCreateX.WebShop.WebServices.Contracts.
ValidationResult, ReCreateX.WebShop.WebServices.Contracts",
            "isValid": true,
            "message": null,
            "brokenRuleName": null
        }
    }
}
```

SaveEmployeeResult

Name	Type	Description	Remarks
Employee	Employee	The employee you want to save	
ValidationResult	ValidationResult	If anything went wrong, this object contains additional information	

SaveEmployeeTaskAttachment

This method is used to save an employee task attachment/comment or both.

Request

The request consists of the service context and an EmployeeTaskAttachment object that inherits from the AttachmentComment object.

Response

The response consists of an EmployeeTaskAttachmentValidationResult object that inherits from the ValidationResult object.

Name	Type	Description	Remarks
EmployeeTaskAttachment	EmployeeTaskAttachment	The EmployeeTaskAttachment object as it was created	

IsValid	Bool	Indicates if this is a valid ValidationResult.	
Message	String	The standard error.	
BrokenRuleName	String	The code of the error message.	

SaveEmployeeTasks

This method is used to find one or more employee tasks.

Request

The request consists of the service context and a SaveEmployeeTaskCriteria object.

SaveEmployeeTaskCriteria

Name	Type	Description	Remarks
EmployeeTask	EmployeeTask	The employee task you want to save	
UpdateSkills	bool	Do the skills needs to be updated?	
IgnoreWarnings	bool	If any warnings, do you want to ignore them?	
SavedByEmployeeId	Guid	The ID of the employee who creates the task	

Response

The response will contain a list of SaveEmployeeTaskResult objects that meet the criteria.

SaveEmployeeTaskResult

Name	Type	Description	Remarks
EmployeeTask	EmployeeTask	The employee task you want to save	
ValidationResult	ValidationResult	If anything went wrong, this object contains additional information	

SaveEmployeeTaskStatus

This method is used to update the status of an employee task.

Request

The request consists of the service context and a SaveEmployeeTaskStatusCriteria object.

Name	Type	Description	Comment
Id	Guid	The unique identifier of the employee task	
Status	EmployeeTaskStatus	The new status for the employee task. One of the following values: • ToBeConfirmed • Confirmed • Refused • Executed	

Response

The response consists of a ValidationResult object that indicates whether or not the update was successful.

Name	Type	Description	Remarks
IsValid	Bool	Indicates if this is a valid ValidationResult.	
Message	String	The standard error.	
BrokenRuleName	String	The code of the error message.	

FindSalesShifts

Calling this will visualize the salesShifts on basis of the linked criteria

Request

The request consists of the service context and an FindSalesShiftsCriteria object.

FindSalesShiftsCriteria

Name	Type	Description	Remarks
FromDate	DateTime	All SalesShifts that have a startdate later of equal to this date	Mandatory
TillDate	DateTime	All SalesShifts that have an enddate before of equal to this date	Mandatory
EmployeeId	Guid	Retrieves all SalesShifts for that employee	Optional
PointOfSaleId	Guid	Retrieves all SalesShifts for that Point of Sale	Optional
Paging	Paging Object	This determines how many salesShifts and from where to start showing	Optional

Response

A SalesShiftsResult containing a List of SalesShifts

SalesShift

Name	Type	Description	Remarks
DivisionId	Guid	Id of the Division	
PointOfSaleId	Guid	Id of the Point of Sale	
EmployeeId	Guid	Id of the Employee	
Shift	int	SaleShift number	
StartTime	DateTime	Date and time of the start of the Shift	
EndTime	DateTime	Date and time of the end of the Shift	
StartBalance	Decimal	The amount of money in the Point of Sale when starting a Sale Shift	
Amount	Decimal	Total amount of money by selling	
LineAmount	Decimal	Total amount of money by selling without the VAT	
VATAmount	Decimal	Total amount of VAT by selling	

DeleteEmployeeAvailability

Is used to delete an employee availability.

Request

The DeleteEmployeeAvailabilitiesRequest contains on the one hand the standard service context object and on the other hand the **DeleteEmployeeAvailabilitiesCriteria** object.

Name	Type	Description	Remarks
EmployeeAvailabilities	List<Guid>	The list of ID's that will be deleted	

Response

The DeleteEmployeeAvailabilityResponse object contains a result object that indicates wether or not the call was successful.

Name	Type	Description	Remarks
Success	Boolean	This will indicate if the creation/update was successful	
Message	String	This will show a message of what went wrong when Success is false.	

FindEmployeeAvailabilities

Is used to call a list of the availabilities of an employee

Request

The FindEmployeeAvailabilityRequest contains on the one hand the standard service context object and on the other hand the **EmployeeAvailabilityCriteria** object.

Name	Type	Description	Remarks
EmployeeID	Guid	The ID of the employee you want the availabilities of	This is mandatory
From	DateTime?	Use this if you only want the availabilities after this date	
Until	DateTime?	Use this if you only want the availabilitie before this date	

Response

The FindEmployeeAvailabilityResponse object contains a list of EmployeeAvailability objects that meet the criteria above.

Name	Type	Description	Remarks
Id	Guid	The Id that represents the EmployeeAvailabilityDay	
EmployeeID	Guid	the Id that represents the Employee the Availability is linked to	
From	DateTime	The start datetime of the availability	
Until	DateTime	The end datetime of the availability	
Comment	string	The Comment on the Availability	

SaveEmployeeAvailability

Is used to create or update an employee availability.

Request

The SaveEmployeeAvailabilityRequest contains on the one hand the standard service context object and on the other hand the Save**EmployeeAvailabilityCriteria** object.

Name	Type	Description	Remarks
EmployeeAvailability	EmployeeAvailability	The object for creating the availability.	

EmployeeAvailability

The object for creating the availability.

Name	Type	Description	Remarks
Id	Guid	The Id that represents the EmployeeAvailabilityDay	filled in for an update. empty or Guid.Empty to create a new one
EmployeeID	Guid	The Id of the employee u want to create the availability for.	
From	DateTime	The date that will be used to create/update the availability start date.	
Until	DateTime	The date that will be used to create/update the availability end date.	
Comment	string	The comment that will be used to create/update the availability	

Response

The SaveEmployeeAvailabilityResponse object contains a result object dat indicates wether the call was successful.

Name	Type	Description	Remarks
Success	Boolean	This will indicate if the creation/update was successful	
Message	String	This will show a message of what went wrong when Success is false.	

GetTaskFlowSettings

Gets the settings related to TaskFlow.

Request

This API Call does not have any criteia

Response

The GetTaskFlowSettings object contains a list of settngs.

Name	Type	Description	Remarks
WebTaskCanBePicked	Boolean	Indicates if webtasks can be picked.	
EmployeeCanCancelConfirmedTasks	Boolean	Indicates if employees can cancel confirmed tasks.	

SaveEmployeeAvailabilities

Is used to create or update a list of employee availabilities.

Request

The SaveEmployeeAvailabilitiesRequest contains on the one hand the standard service context object and on the other hand the **SaveEmployee AvailabilitiesCriteria** object.

Name	Type
EmployeeAvailabilities	List<EmployeeAvailability>

EmployeeAvailability

The object for creating the availability.

Name	Type	Description	Remarks
Id	Guid	The Id that represents the EmployeeAvailabilityDay	filled in for an update. empty or Guid.Empty to create a new one
EmployeeID	Guid	The Id of the employee u want to create the availability for.	
From	DateTime	The date that will be used to create/update the availability start date.	
Until	DateTime	The date that will be used to create/update the availability end date.	
Comment	string	The comment that will be used to create/update the availability	

Response

The SaveEmployeeAvailabilityResponse object contains a result object dat indicates wether the call was successful.

Name	Type	Description	Remarks
Success	Boolean	This will indicate if the creation/update was successful	
Message	String	This will show a message of what went wrong when Success is false.	

DeletePurchaseLine

This function is used to delete existing order lines.

Request

The DeletePurchaseLineRequest contains the basic service context object and a PurchaseLineRequest (see 16.6.2.3) one wants to delete.
As is the case for saving, the ID is very important. If the ID is not valid, you will not be able to delete.
Mark that an order line can no longer be deleted using the Webservices if its status is one of the following: Ordered, Partially received, Entirely received.

Response

The DeletePurchaseOrderResponse object contains a **DeletePurchaseLineResult**, that holds the result of the Delete operation.
Its content is completely identical to that of the SavePurchaseLineResponse.

Name	Type	Description
ValidationResult	ValidationResult	The result of the validation, right before deleting the order line. If the ID is invalid, the ValidationResult will be set to 'PurchaseOrderHeaderMissingIdRule'. If the status of the order does not allow to delete the order, the ValidationResult will be set to 'PurchaseOrderHeaderStatusRule'.
ResultState	ResultState	Indicates if deleting succeeded or not.The possible values are: • None • Success • Failed
PurchaseLine	PurchaseLine	The deleted order line.This will only be filled in if ResultState: Success

DeletePurchaseOrder

This function is used to delete existing orders.

Request

The DeletePurchaseOrderRequest contains the basic service context object and a **PurchaseOrderRequest** (see 16.6.2.1) one wants to delete.
As is the case for saving, the ID is very important. If the ID is not valid, you will not be able to delete.
Mark that an order can no longer be deleted using the Webservices if its status is one of the following: Ordered, Partially received, Entirely received.

Response

The DeletePurchaseOrderResponse object contains a **DeletePurchaseOrderResult**, that holds the result of the Delete operation.
Its content is completely identical to that of the SavePurchaseOrder Response.

Name	Type	Description
ValidationResult	ValidationResult	The result of the validation, right before deleting the order. If the ID is invalid, the ValidationResult will be set to 'PurchaseOrderHeaderMissingIdRule'. If the status of the order does not allow to delete the order, the ValidationResult will be set to 'PurchaseOrderHeaderStatusRule'.
ResultState	ResultState	Indicates if deleting succeeded or not.The possible values are: • None • Success • Failed
PurchaseOrder	PurchaseOrder	The deleted order.This will only be filled in if ResultState: Success

FindPurchaseArticles

FindPurchaseArticles is used to call up the articles that can be purchased by a user.

Request

In addition to the basic service context object, the FindPurchaseArticlesRequest object also contains a PurchaseArticleSearchCriteria object in which the criteria can be specified for finding the articles that can be purchased.

Name	Type	Description	Remarks
Paging	PagingCriteria	See 4.2: Paging Criteria	
ArticleId	GUID	If filled in: The ID of the article one wants to find.	Nullable
ArticleGroupId	GUID	If filled in: only call up the articles that are part of the given article group.	Nullable
NamePattern	String	If filled in: only call up the articles of which the name contains the given pattern.	
ArticleCategoryId	GUID	If filled in: only call up the articles that are linked to the given category.	Nullable
EmployeeId	GUID	The employee of which the article rights have to be checked.	Nullable
Includes	PurchaseArticleIncludes	Include parameters for the request (which extra objects have to be included in the response?).	

Remarks:
As with the FindVendors function the EmployeeId is used here as well to call up only those articles for which the user has rights.
If the purchase parameters within ReCreateX indicate that employee rights have to be used on article group level, and if the employee is not specified in the criteria object, the resultset will be empty.
If the purchase parameters within ReCreateX indicate that no employee rights have to be used on article group level, no additional tests will be executed on the visibility of the articles for the employee. In this case the EmployeeId is not obligatory.
The ID of the employee can be found by using the functions FindEmployees (see 15.2) or AuthenticateEmployee (see 15.3).
Except for the criteria above, additional tests that determine the visibility within the Webservices will be executed when calling up the articles.

- The carticle can not be blocked.
- The option 'Show on Website' has to be ticked off.
- The ledger account for registering the purchase has to be filled in.
- It must be possible to buy the article at at least 1 supplier.Mark that the supplier status can not be blocked and that the user must have rights for the supplier.
- The article units (price unit & purchase unit) used in the purchase prices have to exist.
- A warehouse has to exist in which the article can be stocked.

PurchaseArticleIncludes

Name	Type	Description	Remarks
ImageUrl	Bool	Does the ImageUrl of the article have to be built up?	
Image	Bool	Does the image of the article have to be called up?	
ArticleCategories	Bool	Do the article categories have to be called up?	
ArticleGroup	Bool	Do the article groups have to be called up?	
StockLocations	Bool	Do the warehouses have to be called up?	
Vendors	Bool	Do the suppliers have to be called up?	
ArticleUnits	Bool	Do the units have to be called up?	

Response

The FindPurchaseArticlesResponse object contains a list of **PurchaseArticle** objects that meet the criteria above.

Name	Type	Description
Id	GUID	The unique identifier of the Article.
Code	String	The code of the Article.
Name	String	The name of the Article.
Description	String	The description of the Article.
Price	Decimal	The sales price of the Article today, without taking into account price groups.
Stock	Decimal	

		The current number in stock.If a warehouse has been specified in the request, the response will only contain the number in stock in the current warehouse.If no warehouse has been specified, the response will contain the total number of articles in stock in all warehouses.
Image	Picture	The picture of the Article.
ImageUrl	String	The link on which a picture of the ArticleGroup can be found.
IsRental	Bool	Indicates whether the article is a rental article or not.
Group	ArticleGroup	The ArticleGroup to which the article belongs.
IsStockItem	Bool	Indicates whether the article is a stock article or not.
DivisionId	GUID	The financial division to which the article belongs. Sales of this article will be registered on this division.
Ingredients	IList<ArticleIngredient>	The list of ingredients out of which this article consists.
Options	IList<ArticleOption>	The exclusive options that can be chosen when selling this article.
StockLocations	IList<StockLocation>	The warehouses in which this article is available.
Categories	IList<Category>	The list of categories to which this article belongs. The categories can mutually share a hierarchy.
PurchaseArticle Calculation Type	PurchaseArticle Calculation Type	The calculation type for calculating the total price. This is one of the enumeration values below: • PriceTimesHours • FixedPrice, • PriceTimesQuantity, • PriceTimesQuantityTimesHours
Vendors	IList<VendorArticle>	The list of suppliers where this article can be bought. See 16.4.2.1: VendorArticle for a description of this object.
BaseUnit	ArticleUnit	The basic unit of the article. This is the smallest possible unit of an article and is usually the unit that is used for calculating the stock. See 16.4.2.2: ArticleUnit for a description of this object.
ArticleUnits	IList<ArticleUnit>	The units that can be used for this article.
IsLessonGroupItem	Bool	Indicates whether the article is a lessongroup article or not.

VendorArticle

One article can usually be bought at different suppliers.
This object contains the article price and the units per article / supplier combination.
Units:
Suppose we have an article 'Jupiler 33 cl' of which the basic unit is 'Bottle'.The supplier will usually give the price per crate. In that case the *Price unit* will refer to an ArticleUnit having 'CRATE' as a definition.Moreover, it is for example possible that if we purchase beer at the supplier Inbev, we always order per pallet. In that case the PurchaseUnit will refer to an ArticleUnit having 'PALLET' as a definition.

Name	Type	Description
VendorId	GUID	The ID of the supplier.
Price	Decimal	The price of the article at this supplier for the price unit below.
PriceUnit	ArticleUnit	The price unit, this is the unit that will be used to determine the price.
PurchaseUnit	ArticleUnit	The purchase unit. If we buy this article at the supplier mentioned above, this is the unit that will be bought.

ArticleUnit

ArticleUnit describes the packaging unit of an article, and the proportion to the basic unit.

Name	Type	Description

Id	GUID	The ID of the unit.
Code	String	The code of the packaging unit.
Name	Name	The name of the packaging unit.
BaseQuantity	Decimal	The number of basic units that go into this unit. If the basic unit of the article is for example Bottle, and if 24 bottles go into a crate, the BaseQuantity for the ArticleUnit 'Crate' will be 24.

FindPurchaseOrders

This function is used to call up existing purchase orders.
This can be useful to see the order history of an employee, so that he can still modify existing orders if required.

Request

In addition to the basic service context, the FindPurchaseOrdersRequest object also contains the **PurchaseOrderSearchCriteria** object below in which you can specify which orders have to be called up.

Name	Type	Description
Includes	PurchaseOrderIncludes	Include parameters for the request (which extra objects have to be included in the response?).
Paging	PagingCriteria	See 4.2: Paging Criteria
PurchaseId	GUID?	If filled in, only the purchase order that corresponds to this specific order ID will be called up.
OrderNo	String	If filled in, only the order with this specific order number will be called up.
Reference	String	If filled in, only call up the orders of which the reference contains the given value.
EmployeeId	GUID?	If filled in, only call up the orders that have been created by the given employee.
VendorId	GUID?	If filled in, only call up the orders that have been created for the given supplier.
FromCreationDate	DateTime?	If filled in, only call up the orders that have been created after the given date.
TillCreationDate	DateTime?	If filled in, only call up the orders that have been created before the given date.
ArticleId	GUID?	If filled in, only call up the orders that contain order lines for the given article.
FromLastUpdate	DateTime?	If filled in, only call up the orders that have been modified after the given date.
TillLastUpdate	DateTime?	If filled in, only call up the orders that have been modified before the given date.

Response

The FindPurchaseOrdersResponse object contains a list of **PurchaseOrder** objects that meet the criteria above.

PurchaseOrderRequest

PurchaseOrderRequest is the basic object that can be used to create new orders.
PurchaseOrder is derived from this and contains the same fields + a number of extra fields that the user can not modify himself. These are managed by the back office.

Name	Type	Description
Id	GUID	The unique ID of the PurchaseOrder.
EmployeeId	GUID	The unique ID of the employee that created the order.
Reference	String	A free info field that can be used to save a reference defined by the user.
VendorId	GUID	The unique ID of the supplier to whom this order is addressed.
VendorContactId	GUID?	The unique ID of the contact person of the supplier above.
PublicComment	String	Free remarks field. The content of this field is usually printed on the order form that is sent to the supplier.
PrivateComment	String	Free remarks field. The content of this field is not printed on the order form and serves rather as an internal remarks field.
RequestedDeliveryDate	DateTime?	The desired delivery date of the order.

PurchaseOrder

PurchaseOrder is an extension of PurchaseOrderRequest and contains the same basic fields, completed by the following fields.
These fields can only be filled in in the back office.

Name	Type	Description
OrderNo	String	The unique sequence number of the order form in the back office.
OrderState	PurchaseOrderState	The status of this order in the back office. This is an enumeration that can contain one of the following values: • ToBeOrdered: Still has to be ordered. • Ordered: Is ordered. • PartiallyReceived: Partially received. • EntirelyReceived: Entirely received. • ToApprove: Still has to be approved. • Rejected: Rejected.
PostingDate	DateTime	Date on which the order was registered.
CreationDate	DateTime	Date on which the order was created.
PromisedDeliveryDate	DateTime	Promised delivery date.
PurchaseLines	IList<PurchaseLine>	Contains order lines of this order.1 order line is an order for 1 article at a supplier. See 16.6.2.4: PurchaseLine

PurchaseLineRequest

PurchaseLineRequest is the basic object that is used in the Webservices to add order lines to an order.
PurchaseLine is derived from this and contains extra fields that are filled in automatically and can only be modified in the back office.
1 PurchaseLineRequest (and thus also PurchaseLine) corresponds to the order of 1 article at 1 supplier.

Name	Type	Description
Id	GUID	The unique ID of the order line.
PurchaseId	GUID	The ID of the order to which this line belongs.See 16.6.2.2: PurchaseOrder
ArticleId	GUID	The ID of the article that has to be ordered.
Description	String	The description of the article that is purchased.If empty, this field will be automatically completed with the description field of the article on the moment the order line is saved.
Description2	String	The additional description of the article that is purchased.If empty, this field will be automatically completed with the additional description field of the article on the moment the order line is saved.
Purchase UnitId	GUID	The (packaging) unit of the article one wants to purchase. This corresponds to the ID of the chosen ArticleUnit. See 16.4.2.2: ArticleUnit
Quantity	Decimal	The number of the units above one wants to order. The given number will influence the price if the calculation method is PriceTimesQuantity or PriceTimesQuantityTimesHours.
StockLocationId	GUID?	The warehouse for which one wants to order this article.This field is obligatory if the article is a stock article and can not be filled in if the article is not a stock article.
Hours	Decimal?	The number of hours one wants to buy.This is mainly used if services are purchased instead of articles. This value will influence the price if the calculation method of the article is PriceTimesHours or PriceTimesQuantityTimesHours.

PurchaseLine

PurchaseLine is derived from the basic object PurchaseLineRequest and contains the same basic fields + extra fields that are filled in automatically and can only be modified in the back office.

Name	Type	Description
QtyPerPurchaseUnit	Decimal	The number of basic units per purchase unit.
QuantityBase	Decimal	The total number of basic units ordered.This is the result of Quantity x QtyPerPurchaseUnit.
QtyReceivedBase	Decimal?	The total number of basic units of this order line already received.
UnitPrice	Decimal?	The unit price VAT exclusive.
UnitPriceInc	Decimal?	The unit price VAT inclusive.
VatId	GUID?	The ID of the VAT tariff that applies to this article.This can differ in function of the chosen supplier.

VatPercentage	Decimal?	The VAT % that applies to this article.
VatAmount	Decimal?	The VAT amount.
DiscountPercentage	Decimal?	The allowed discount percentage.
DiscountAmount	Decimal?	The discount amount.
LineAmount	Decimal?	The net amount of the order line: (Number x unit price excl. VAT) - discount
BaseAmount	Decimal?	The gross amount of the order line: (Number x unit price excl. VAT)
Amount	Decimal?	The total amount of the order line incl. VATLineAmount + VatAmount
IsVatInclusive	Boolean?	Indicates if the article price is saved incl. or excl. VAT
PriceCalculationType	PurchaseArticleCalculationType	The calculation method used for determining the price: • PriceTimesHours • FixedPrice, • PriceTimesQuantity, • PriceTimesQuantityTimesHours

FindStockLocations

This function will generate the list of warehouses.
This list can be limited to the warehouses for 1 specific article and takes into account the employee's rights on warehouses.
If the purchase parameters in ReCreateX indicate that employee rights are used on warehouse level, the ID of the employee has to be filled in. If not the resultset of this function will be empty.

Request

De FindStockLocationsRequest contains the basic service context object and the **StockLocationSearchCriteria** object below:

Name	Type	Description
Paging	PagingCriteria	See 4.2: Paging Criteria
ArticleId	GUID?	If filled in, the ID of the article of which one wants to find the warehouses.
EmployeeId	GUID?	The ID of the employee of which one wants to find the warehouses.

Response

The FindStockLocationsResponse object consists of a list of **StockLocation** objects.

Name	Type	Description
Id	GUID	The ID of the warehouse.
Code	String	The code of the warehouse.
Description	String	The description of the warehouse.

FindVendorGroups

FindVendorGroups is used to call up the supplier groups.
As with the FindVendors function the EmployeeId is used here as well to call up only those groups for which the user has rights.

Request

The request consists of the basic context object and the **VendorGroupSearchCriteria** described below.

Name	Type	Description	Remarks
Paging	PagingCriteria	(See 4.2: Paging Criteria	
Id	GUID	If filled in: The ID of the supplier group one wants to find.	Nullable
Code	String	Code of the supplier group.	
NamePattern	String	If filled in: Only the supplier groups of which the name contains the name pattern will be shown.	

EmployeeId	Nullable<GUID>	The ID of the logged in employee.	

Response

The FindVendorGroupsResponse object contains a list of **VendorGroup** objects that meet the criteria above.
See 16.2.2.2: VendorGroup for a description of this object.

FindVendors

FindVendors is used to call up the list of available suppliers.
It is advised not to call up this function until an employee has authenticated himself on the Webservices because the VendorSearchCriteria expects an optional EmployeeId.
If it has been specified in ReCreateX that employee rights have to be used on supplier level, the list of suppliers will be empty if no valid EmployeeId is given in this Request.
This is why it is advised to always have the employee authenticate himself first and fill in his EmployeeId in the following requests.

Request

The request consists of the basic context object and the VendorSearchCriteria described below.

Name	Type	Description	Remarks
Paging	PagingCriteria	See 4.2: Paging Criteria	
Id	GUID	If filled in: The ID of the supplier one wants to find.	
Code	String	Code of the supplier.	
NamePattern	String	If filled in: Only the suppliers of which the name contains the name pattern will be shown.	
VendorGroupID	Nullable<GUID>	If filled in, only the suppliers that are part of the given supplier group will be visualised.	
EmployeeId	Nullable<GUID>	The ID of the logged in employee.	
IncludeBlocked	Nullable<Bool>	If False, the blocked suppliers will not be included. If NULL or true, these will be included.	
Includes	VendorIncludes	Include parameters for the request (which extra objects have to be included in the response?).	See 16.2.1.1: VendorIncludes

VendorIncludes

Name	Type	Description
IncludeVendorContacts	GUID	If true, the contact persons of the supplier will be included.

Response

The **FindVendorsResponse** object contains a list of **Vendors** as described below:

Name	Type	Description	Remarks
Id	GUID	ID of the supplier.	
Code	String	The code of the supplier.	
Name	String	The name of the supplier.	
Name2	String	The additional name of the supplier.	
Address	Address	See 8.8.1.3: AddressContains the address data of the supplier.	
Phone1	String	Telephone number of the supplier.	
Phone2	String		
Phone3	String		
Fax	String	The fax number of the supplier.	
Language	String	The ISO language code of the supplier.	
VatNo	String	The VAT number of the supplier.	
BankAccount1	String	The bank account number of the supplier.	
BankAccount2	String		

Email	String	The e-mail address of the supplier.	
Email2	String		
WebSite	String	The URL address of the Website of the supplier.	
Comment	String	Remarks field with free information concerning the supplier.	
InvoiceVendorId	Nullable<GUID>	The ID of the invoice supplier. By default this is the same ID as the supplier himself. It can, however, be different if purchases at supplier A are invoiced to supplier B.	
VendorState	VendorState	The status of the supplier.See 16.2.2.1: VendorState	
VendorGroup	VendorGroup	The group of suppliers to which this supplier belongs.See 16.2.2.2: VendorGroup	
VendorNo	Long	The identification number of the supplier.	
VatRegime	VatRegime	The VAT rate that applies to this supplier. See 16.2.2.3: VatRegime	
VendorContacts	Ilist<VendorContact>	The list of contact persons for this supplier.See 16.2.2.5: VendorContact	

VendorState

VendorState contains the definition of the supplier's status.This can be for example Active, Stopped, Blocked, ...

Name	Type	Description	Remarks
Id	GUID	ID of the supplier status.	
Code	String	The code of the supplier status.	
Name	String	The name of the supplier status.	
IsBlocked	Bool	If True, this indicates that suppliers with this status are blocked.	

VendorGroup

VendorGroup contains the definition of a supplier group.

Name	Type	Description	Remarks
Id	GUID	ID of the supplier group.	
Code	String	The code of the supplier group.	
Name	String	The name of the supplier group.	

VatRegime

VatRegime contains the definition of the VAT rate that applies, in this case, to the supplier.

Name	Type	Description	Remarks
Id	GUID	ID of the VAT rate.	
Code	String	The code of the VAT rate.	
Name	String	The name of the VAT rate.	
Vat	VAT	The VAT tariff that applies to this VAT rate. See 16.2.2.4: VAT	

VAT

VAT contains the definition of a VAT tariff.

Name	Type	Description	Remarks
Id	GUID	ID of the VAT tariff.	
Name	String	The name of the VAT tariff.	
Value	Decimal	The percentage value of the VAT tariff.	

VendorContact

Name	Type	Description	Remarks
Id	GUID	ID of the contact person.	
Name	String	The name of the contact person.	
Functions	String	The function of the contact person with the supplier.	
Phone1	String	The first telephone number of the contact person.	

Phone2	String	The second telephone number of the contact person.	
Phone3	String	The third telephone number of the contact person.	
Fax	String	The fax number of the contact person.	
Email	String	The e-mail address of the contact person.	
Comment	String	Free remarks concerning the contact person.	

Purchases – Brief overview

The Webservices for the purchase module have got nothing to do with the shopping basket. Here, purchase orders are concerned while the shopping basket exclusively contains sales.
They are, however, included in this documentation since these Webservices are included in the same package.
In case of the sales Webservices (Basket items) it allowed to shop anonymously. This is not allowed within the purchase Webservices since, for security purposes, a lot of data can not be visual for certain end users. This is why it is obligatory to specify an EmployeeId for al lot of functions.
If the EmployeeId is not filled in and if the ReCreateX settings indicate that employee rights must be used within the purchase module, these functions will not return data.
This means that the Webservices for the purchase module can only be used when an employee logs in to the system.
Employees can not be created in the Webservices. This has to be done in the back office.
Authentication, however, is possible: Therefor, you can use the functions described in chapter 15 Employees.
Overview of the available functions:

- FindVendors: Find suppliers.
- FindVendorGroups: Find supplier groups.
- FindPurchaseArticles: Find articles that can be purchased.
- FindStockLocations: Find warehouses.
- FindPurchaseOrders: Find existing purchase orders.
- ValidatePurchaseOrder: Validate a purchase order.
- ValidatePurchaseLine: Validate a purchase order line.
- SavePurchaseOrder: Save a purchase order.
- DeletePurchaseOrder: Delete a purchase order.
- SavePurchaseLine: Save a purchase order line.
- DeletePurchaseLine: Delete a purchase order line.

SavePurchaseLine

This function is used to add new order lines to existing orders and to save modifications of existing order lines.
The functioning is completely analogous to the SavePurchaseOrder.

Request

The SavePurchaseLineRequest contains the basic service context object and a PurchaseLineRequest (see 16.6.2.3) one wants to save.
The ID of the PurchaseLineRequest is very important here!
If one wants to make new order lines, the ID has to be empty.
If one wants to modify an existing order line, the ID has to be filled in with the ID of the existing order line.
Order lines can only be added to existing orders. It is thus important that the order is created before the order line and that the PurchaseId of the order line refers to the existing order.

Response

The SavePurchaseLineResponse object contains a **SavePurchaseLineResult**, that holds the result of the Save operation.

Name	Type	Description
ValidationResult	ValidationResult	The result of the validation, right before saving the order line. See amongst others 16.8: ValidatePurchaseLine
ResultState	ResultState	Indicates if saving succeeded or not.The possible values are: • None • Success • Failed
PurchaseLine	PurchaseLine	The saved order line.This will only be filled in if ResultState: Success

It is important that, once the order line has been saved, further modifications are made in the saved order line.

This is necessary because, after saving the order line, the ID will be filled in in the resulting PurchaseLine object.
If you would keep working in the original request object, the ID would remain empty and new order lines would be made each time.

SavePurchaseOrder

This function is used to save new orders or to modify existing orders.The SavePurchaseOrder only allows to save headers. Adding or saving detailed lines can be done using the SavePurchaseLine function.

Request

The SavePurchaseOrderRequest contains the basic service context object and a **PurchaseOrderRequest** (see 16.6.2.1) one wants to save.
The ID of the PurchaseOrderRequest is very important here!
If one wants to make a new order, the ID has to be empty.
If one wants to modify an existing order, the ID has to be filled in with the ID of the existing order.

Response

The SavePurchaseOrderResponse object contains a **SavePurchaseOrderResult**, that holds the result of the Save operation.

Name	Type	Description
ValidationResult	ValidationResult	The result of the validation, right before saving the order. See amongst others 16.7: ValidatePurchaseOrder
ResultState	ResultState	Indicates if saving succeeded or not.The possible values are: • None • Success • Failed
PurchaseOrder	PurchaseOrder	The saved order.This will only be filled in if ResultState: Success

It is important that, once the order has been saved, further modifications are made in the saved order.
This is necessary because, after saving the order, the ID will be filled in in the resulting PurchaseOrder object in SavePurchaseOrderResult.
If you would keep working in the original request object, the ID would remain empty and new orders would be made each time.

ValidatePurchaseLine

This function is used to validate an order line before it is saved.

Request

The ValidatePurchaseLineRequest object contains the basic service context object and the **PurchaseLineRequest** object to be validated.
See 16.6.2.3 for a description of the PurchaseLineRequest object.

Response

The ValidatePurchaseOrderResponse contains a **ValidationResult** object that indicates if the validation succeeded or not.
If the validation failed because the data are not valid, the BrokenRuleName will indicate which validation rule failed. The Message parameter shows the textual content of the error.
If the validation succeeded, the BrokenRuleName will be empty.
These are the validation rules that are used during validation:

RuleName	Description
PurchaseOrderLineMissingPurchaseIdRule	Checks if PurchaseId has been filled in.This is an obligatory field when saving order lines.
PurchaseOrderLineInvalidPurchaseIdRule	Checks if PurchaseId contains a valid value. PurchaseId has to refer to an existing order line if one wants to save an order line.
PurchaseOrderLinePurchaseIdChangedRule	Checks if it is allowed to change the PurchaseId of existing order lines afterwards.
PurchaseOrderAllreadyBookedRule	Checks if modifications of an existing line are allowed. It is not allowed to modify an order line that has already been ordered.
PurchaseOrderLineInvalidArticleRule	Checks if the given ArticleId contains a valid value.
PurchaseOrderLineInvalidArticleForVendorRule	Checks if the given article can be used with the supplier linked to the order.

PurchaseOrderLine InvalidArticleForEmployeeRule	Checks if the employee has sufficient rights to buy this article.
PurchaseOrderLineInvalidPurchaseUnitRule	Checks if the PurchaseUnitId contains a valid value in function of the chosen article.
PurchaseOrderLineInvalidQuantityRule	Checks if Quantity contains a valid value. Negative values are not allowed here.
PurchaseOrderLine StockLocationForNonStockItemRule	Checks if the warehouse is empty in case of non-stock articles.The warehouse can not be filled in if an article is not a stock article.
PurchaseOrderLineInvalidStockLocationRule	Checks if the warehouse contains a valid value for the chosen article.
PurchaseOrderLine InvalidStockLocationForEmployeeRule	Checks if the warehouse can be used by the employee.
PurchaseOrderLineInvalidHoursRule	Checks if the field Hours contains a valid value. This value can not be negative.

ValidatePurchaseOrder

This function is used to validate an order before it is saved.

Request

The ValidatePurchaseOrderRequest object contains the basic service context object and the **PurchaseOrderRequest** object to be validated.
See 16.6.2.1 for a description of the PurchaseOrderRequest object.
This function can be used to validate both new as modified orders.

Response

The ValidatePurchaseOrderResponse contains a **ValidationResult** object that indicates if the validation succeeded or not.
ValidationResult object:

Name	Type	Description	Remarks
BrokenRuleName	String	The code of the error message.	
Message	String	The standard error.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	

If the validation failed because the data are not valid, the BrokenRuleName will indicate which validation rule failed. The Message parameter shows the textual content of the error.
If the validation succeeded, the BrokenRuleName will be empty.
These are the validation rules that are used during validation:

RuleName	Description
PurchaseOrderHeaderMissingVendorRule	Checks if VendorId is filled in and contains a valid value.
PurchaseOrderHeaderMissingEmployeeRule	Checks if EmployeeId is filled in and contains a valid value.
PurchaseOrderHeaderInvalidVendorForEmployeeRule	Checks if VendorId is a valid supplier for the given EmployeeId.
PurchaseOrderHeaderInvalidVendorContactRule	Checks if VendorContactId contains a valid value for the given supplier.
PurchaseOrderHeaderVendorChangeNotAllowedRule	In case the PurchaseOrderRequest is an existing order of which the VendorId has been modified, this validation rule will check if the modification is allowed.

17 Letting module (as of RCX 5.0.5)

CancelRentalOrder

This method will cancel one specific RentalOrder.
On the Webshop it is only possible to delete orders that:

- Have the status 'Reserved'
- Have the administrative status 'New order'
- Were created on the Web

Request

The request is built up as follows:

Name	Type	Description	Remarks
RentalOrderId	GUID	ID of the order.	
ReasonID	GUID	The cancellation reason.	
ReasonDescription	String	The description that is given in case of cancellation.	

Response

The response contains a **RentalValidationResult** object. This object contains validation data as well as the original **RentalOrder** object.
The **RentalValidationResult** object looks as follows:

Name	Type	Description	Remarks
BrokenRuleName	String	The code of the error message.	
Message	String	The error message.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	
ValidatedRentalOrder	RentalOrder	RentalOrder object that was called up on the basis of the given ID.	

FindRentalActivities

This method calls up all rental activities.

Request

The request consists of the service context and a RentalActivitySearchCriteria object.

Name	Type	Description	Remarks
Id	GUID	ID of a specific rental activity. (If you mean to call up only 1 entity)	Nullable (if not applicable).
Keyword	String	Keyword when searching for specific rental activities.	Nullable (if not applicable).
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response will contain a list of RentalActivity objects that meet the criteria.
The **RentalActivity** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the rental activity.	
Code	String	The code of the rental activity.	
Name	String	The name of the rental activity.	

FindRentaLarticleavailableperiods

This method calls up all available periods of a rental article.

Request

The request consists of the service context and a RentalArticleAvailablePeriodSearchCriteria object.

Name	Type	Description	Remarks
ArticleId	GUID	All available periods of this rentalarticle will be visualised.	
FromDate	DateTime	The available periods that overlap with the selected period will be shown.	

UntilDate	DateTime	The available periods that overlap with the selected period will be shown.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response contains a **RentalArticleAvailablePeriod** object which looks as follows:

Name	Type	Description	Remarks
ID	Guid	The id of the available period.	
FromDateTime	DateTime	The fromdatetime of the available period.	
ToDateTime	DateTime	The todatetime of the available period.	
Available	Int	The number of items available in the available period.	

FindRentalArticles

This method finds all letting articles that meet de search criteria below and that meet the following conditions:

- The article is a letting article.
- The article may be shown on the Website.
- If a composed article is concerned, the components have to have the same calculation method as the main article.
- A letting location has to exist for the article.
- The collection/return dates have to be within the opening hours of the letting location.
- If rights per division are used, full rights for the article are required.

Request

The request consists of the service context and a **RentalArticleSearchCriteria** object.

Name	Type	Description	Remarks
Id	GUID	ID of a specific article. (If you mean to call up only 1 entity)	Nullable (if not applicable).
NamePattern	String	Name or part of the description that can be searched for. For partial matches, the % sign can be used. The % sign can be used before or after the word, as well as before and after the word.	Nullable (if not applicable).
ArticleGroupIds	IList<GUID>	Only articles of the selected article groups will be visualised.	
FromDate	DateTime	Only articles that are available in the chosen period will be visualised.	
UntilDate	DateTime	Only articles that are available in the chosen period will be visualised.	
DeliveryDate	DateTime	Only articles that are available in the chosen period will be visualised.	
ReturnDate	DateTime	Only articles that are available in the chosen period will be visualised.	
Includes	RentalArticleIncludes	Determines which data are included in the response and/or which extra records will be present in the returnset.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
PersonId	GUID	If filled in, only the articles for which the person's price group is allowed will be included.	
ForReservations	Bool?	List all rental articles for the reservation module?	
ArticleIds	IList<GUID>	List of IDs of the articles	Nullable (if not applicable).
IgnoreShowOnWeb	bool	Ignore the show on web setting	

The **RentalArticleIncludes** object can be built up using the following criteria:

Name	Type	Description	Remarks
ImageUrl	Bool	Find the image or not.	

Image	Bool	Find the image or not.	
ArticleGroup	Bool	Find the data of the article group or not.	
Attachment	Bool	Find the attachments or not.	
ExtraImages	Bool	Find the extra images or not.	
ExtraImageUrls	Bool	Find the extra images or not.	
Components	Bool	Find the parts of an letting article or not. (only applicable in case of composed articles)	
IncludeUnavailable	Bool	Adds the unavailable articles to the result set.	

Response

The response will contain a list of RentalArticle objects that meet the criteria.
The **RentalArticle** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the article.	
DefaultWarranty	Decimal	The warranty of the article.	
WarrantyCalculationType	WarrantyCalculationType	The calculation method of the warranty of the article. (per rental/per piece)	
TotalItemsInStock	Int	The number of articles present in the rental warehouse.	
AvailableToRent	Int	The number of articles available in the chosen period.	
MaxToRent	Int	The max. number of items of this article that can be rented in one order.	
MaxDaysToRent	Int	The max. number of days this article can be rented.	
MaxHoursToRent	Int	The max. number of hours this article can be rented.	
NegativeSaldoAllowed	Bool	Indicates if this article can be rented even though it is no longer available.	
Components	List<RentalArticleComponent>	Contains the list of Components of this article.	
RentalPriceCalculationType	RentalPriceCalculationType	The calculation method of the price of the letting article.	
MaxTimeToRent	DateTime	The max. number of hours and minutes this article can be rented.	
ToDeliver	Bool	Indicates if this article will be delivered to the customer.	
Unavailable	Bool	Indicates if this article is unavailable.	
HasAvailablePeriods	Bool	Indicates if this article has RentalArticleAvailablePeriods.	
DeliverChoice	Bool	Indicates if the choice to deliver or not can be made through the web site.	
IsLessonGroupItem	Bool	Indicates whether the article is a lessongroup article or not.	

The **WarrantyCalculationType** object is an enumeration with the following possible values:
ByRental = 0,
ByQuantity = 1

The **RentalPriceCalculationType** object is an enumeration with the following possible values:
PxH = 0,= Price * Hours
F = 1,= Fixed price
PxQ = 2,= Price * Number
PxQxH = 3,= Price * Number * Hours
PxQxD = 4,= Price * Number * Days
PxQxN = 5,= Price * Number * Nights
The **RentalArticleComponent** is an ArticleBase object with the following extension:

Name	Type	Description	Remarks
Quantity	Int	Indicates of how many components an article exists.	

FindRentalOrders

This method calls up all orders of the logged in person.

Request

The request consists of the service context and a RentalOrderSearchCriteria object.

Name	Type	Description	Remarks
Id	GUID	ID of a specific order. (If you mean to call up only 1 entity)	Nullable (if not applicable).
Number	Long	The order number of the order.	Nullable (if not applicable).
PersonId	GUID	If filled in, all orders of that person will be visualised.	
StartDateTime	DateTime	If filled in, only the orders that start after this date will be shown.	
EndDateTime	DateTime	If filled in, only the orders that end before this date will be shown.	
RentalOrderStatus	RentalOrderStatus	If filled in, only the orders that have this status will be shown.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	RentalOrderIncludes	Determines which data are included in the response and/or which extra records will be present in the returnset.	

The **RentalOrderStatus** object is an enumeration with the following possible values:
Ordered = 0,
PartiallyDelivered = 1,
Delivered = 2,
PartiallyReturned = 3,
Returned = 4,
Cancelled = 5
The **RentalOrderIncludes** object can be built up using the following criteria:

Name	Type	Description	Remarks
IncludeOrderLines	Bool	Find the order lines or not.	
IncludeOrderLineComponents	Bool	Find the order lines of the components or not.	
IncludePersonDetails	Bool	Find the customer's data.	

Response

The response will contain a list of RentalOrder objects that meet the criteria.
The **RentalOrder** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the order.	
RentalNo	Long	The order number of the order.	
RentalWebNo	Long	The order number of the order.	
WebCreated	Bool	Indicates if the order was made on the Web.	
RentalOrderStatus	RentalOrderStatus	Contains the status of the order.	
RentalAdministrativeStatus	RentalAdministrativeStatus	Contains the administrative status of the order.	
CreationDate	DateTime	The date on which the order was created.	
Customer	Person	The person that made the order.	
StartDateTime	DateTime	The start date of the order.	
EndDateTime	DateTime	The end date of the order.	
PlannedDeliveryDateTime	DateTime	The date on which the order will be collected.	
PlannedReturnDateTime	DateTime	The date on which the order will be returned.	
CancelDate	DateTime	The date on which the order was cancelled.	
CancelDescription	String	The description that is given when cancelling the order.	

Comment	String	The description of an order.	
RentalLocation	String	The rental location.	
TotalAmount	Decimal	The total amount of the order.	
Warranty	Decimal	The warranty of the order.	
RentalOrderLines	List<RentalOrderLine>	The order lines of an order.	
Deletable	Bool	Indicates if the order can be cancelled.	
DeliveryAddress	DeliveryAddress	Indicates the delivery address of the order.	See 8.17.2 for more details concerning this object.
EmployeeTaskIds	IList<Guid>	Collection of the linked employee tasks	
ReservationId	Guid	The reservation ID	

The **RentalAdministrativeStatus** object is an enumeration with the following possible values:
NewOrder = 0,
Approved = 1,
Refused = 2
The **RentalOrderLine** object can be built up using the following criteria:

Name	Type	Description	Remarks
ID	GUID	ID of the order line.	
RentalOrderId	GUID	The ID of the order.	
Sequence	Int	A sequence number of the order line.	
StartDateTime	DateTime	The start date of the order line.	
EndDateTime	DateTime	The end date of the order line.	
PlannedDeliveryDateTime	DateTime	The date on which the order line will be collected.	
PlannedReturnDateTime	DateTime	The date on which the order line will be returned.	
ItemID	GUID	The article that is ordered.	
Quantity	Int	The quantity that is ordered.	
Description	String	The description of the order line.	
Info	String	Info field.	
WebCreated	Bool	Indicates if the order line was made on the Web.	
CalculationType	RentalCalculationType	The calculation type of the order line.	
UnitPriceInc	Decimal	The unit price of the article.	
Amount	Decimal	The amount of the order line.	
Deposit	Decimal	The warranty of the article.	
Components	List<RentalOrderLine>	The order lines of the components of the article.	
ToDeliver	Bool	Indicates if this article will be delivered to the customer.	

The **RentalCalculationType** object is an enumeration with the following possible values:
PriceTimesHours = 0,
FixedPrice = 1,
PriceTimesQuantity = 2,
PriceTimesQuantityTimesHours = 3,
PriceTimesQuantityTimesDays = 4,
PriceTimesQuantityTimesNights = 5

GetRentalOrder

This method calls up the details of one specific RentalOrder.

Request

The request consists of the ID of the order.

Name	Type	Description	Remarks

RentalOrderId	GUID	ID of the order.	

Response

The response will contain a **RentalOrder** object.

Name	Type	Description	Remarks
RentalOrder	RentalOrder	RentalOrder object.	See 17.3.2 for details concerning the RentalOrder object.

GetRentalPlaceOpeningHour

Request

Name	Type	Description
Context	ServiceContext	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
Criteria	GetRentalPlaceOpeningHourCriteria	The criteria used for GetRentalPlaceOpeningHour

GetRentalPlaceOpeningHourCriteria

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
Date	Datetime?	The date for which the correct RentalPlaceOpeningHour should be retrieved
Rental PlaceId	Guid?	If passed, RentalPlaceOpeningHours will be returned that are linked to this RentalPlace. Otherwise it will take the Point of Sale's default RentalPlace

Response

GetRentalPlaceOpeningHourResult

Name	Type	Description
RentalPlaceOpeningHourDay	RentalPlaceOpeningHourDay	Returns the RentalPlaceOpeningHourDay

RentalPlaceOpeningHourDay

Name	Type	Description
From1	DateTime?	Start of the morning
Until1	DateTime?	End of the morning
From2	DateTime?	Start of the afternoon
Until2	DateTime?	End of the afternoon
PickupMax	int	Allowed people to pick up per half hour
DeliveryMax	int	Allowed people to deliver per half hour

GetRentalPricing

This method allows you to get the specific price and warranty you will pay for a rental reservation.

Request

The request contains a **RentalPricingSearchCriteria** object which contains following properties:

Name	Type	Description	Remarks
ArticleId	GUID	ID of the article you wish to rent	
PersonId	GUID?	ID of the person who wishes to rent	Not mandatory
Quantity	Int	Number of articles you wish to rent	
From	DateTime	From date of the rental reservation	
Until	DateTime	Until date of the rental reservation	

Response

The response contains a **RentalPricing** object which looks as follows:

Name	Type	Description	Remarks
PriceCalculationType	RentalPriceCalculationType	The price calculation type (see below)	
Price	Decimal	The unit price for the rental reservation.	
WarrantyCalculationType	WarrantyCalculationType	The warranty calculation type (see below).	
Warranty	Decimal	The default warranty for the rental reservation	

The **RentalPriceCalculationType** object is an enumeration with the following possible values:
PxH = 0,= Price * Hours
F = 1,= Fixed price
PxQ = 2,= Price * Number
PxQxH = 3,= Price * Number * Hours
PxQxD = 4,= Price * Number * Days
PxQxN = 5,= Price * Number * Nights
The **WarrantyCalculationType** object is an enumeration with the following possible values:
ByRental = 0,
ByQuantity = 1

Letting – Brief overview

As of now, it is possible to order letting articles on the Webshop.
The system functions as follows:

- Person chooses the period in which he wants to rent an article.
- Person chooses the quantity of the article he wants to rent.

Detailed management of letting orders is done in ReCreateX.
The Webservice methods that have been created for this purpose are the following:

- FindRentalArticles (find the letting articles)
- FindRentalOrders (find existing orders of the logged in person)
- GetRentalOrder (find one order)
- CancelRentalOrder (cancel one order)

18 Activities module (as of RCX 5.0.5)

Activities – Brief overview

It is now possible to register persons for activities on the Webshop.
The system functions as follows:

- Person chooses an activity for which he wants to register.
- If the person is already logged in, he can register himself as well as his relations. If the person is not yet logged in, the person can only register himself (anonymously).
- It can be specified per participant for which days he wants to register.
- Per participant / chosen day the person can determine the number of options.
- At payment (check-out) the registration is definitive. In case of an anonymous registration, the participant will be used as invoice customer as well.

Detailed management of the registration is done in ReCreateX.
The Webservice methods that have been created for this purpose are the following:

- ListActivityTypes (find the activity types)
- FindActivities (find the activities)
- FindPriceGroupActivityByActivity (find the possible price groups with linked prices defined for the activity)

ActivityOptionGetAvailability

This service checks the availability for a certain option on a certain activity day.

Request

The request consists of the service context and an ActivityOptionAvailabilityCriteria object.

ActivityOptionAvailabilityCriteria

Name	Type	Description	Remarks
ActivityDayID	GUID	Unique identifier of the activity day	Obligatory.
ActivityOptionID	GUID	Unique identifier of the activity option	Obligatory.

Response

The response will return an **int** which defines the available number op options.
If the return value equals -1, no maximum is defined.

CancelEntry

This method will cancel one specific registration (Entry).
On the Webshop it is only possible to delete registrations that are:

- Not yet cancelled (status = temporarily, reserved or waiting list)
- Not yet registered

Request

The request is built up as follows:

Name	Type	Description	Remarks
EntryId	GUID	ID of the registration.	
ReasonID	GUID	The cancellation reason.	

Response

The response is a simple boolean. The cancellation either succeeded or not.

FindActivities

This service visualises a list of activities.

Request

The request consists of the service context and an ActivitySearchCriteria object.
This criterion contains an object to take care of paging in case of long lists and different objects to generate the desired list of activities.

ActivitySearchCriteria

Name	Type	Description	Remarks
Paging	PagingCriteria	See 4.2: Paging Criteria	
Period	Period	Finds the activities organised within this period. See point 18.3.2.1	
AgeFrom	int?	Activities that are intended for persons older than or equal to this age.	
AgeTo	Int?	Activities that are intended for persons younger than or equal to this age.	
ActivityId	GUID	Calls up 1 activity only with an ID equal to this value.	
ActivityTypeId	GUID	Calls up the activities that belong to this type.	
Keyword	String	Calls up the activities of which the name contains this text.	

PersonId	GUID	If filled in, the activity price valid for this person will be called up.	
Includes	ActivityIncludes	Determines which parts of the Activity will be included.	
FilterDays	Bool	Enable the filter by day?	
FilterMonday	Bool	Include activities on mondays	
FilterTuesday	Bool	Include activities on Tuesdays	
FilterWednesday	Bool	Include activities on Wednesdays	
FilterThursday	Bool	Include activities on Thursdays	
FilterFriday	Bool	Include activities on Fridays	
FilterSaturday	Bool	Include activities on Saturdays	
FilterSunday	Bool	Include activities on Sundays	
ActivityTypeIdList	List<Guid>	Calls up the activities that belong to this list of activity types	
UpdatedSince	DateTime?	Find the activities that are updated since a certain date	
OrganisationTypeId	GUID	Calls up the activities that belong to this organisation type	
OrganisationTypeIdList	List<Guid>	Calls up the activities that belong to this list of organisation types	
CareTypeId	GUID	Calls up the activities that belong to this care type	
CareTypeIdList	List<Guid>	Calls up the activities that belong to this list of care types	
ActivityTypeSearchTermId	GUID	Calls up the activities that belong to an activity type which has this search term associated with it	
ActivityTypeSearchTermIdList	List<Guid>	Calls up the activities that belong to an activity type which has at least one of this list of search terms associated with it	
LocationId	GUID	Calls up the activities that have this location	
LocationIdList	List<Guid>	Calls up the activities that have a location in this list	
DaysPeriod	Period	Find the activitydays within this period	

ActivityIncludes

This object allows to exclude certain parts of an activity.

Name	Type	Description	Remarks
ImageUrl	Bool	Show the URL of the image or not.	
DayParts	Bool	Show list of day parts (List<ActivityPart>) per activity.	
Options	Bool	Show list of options (List<ActivityOption>).	
Image	Bool	Show the image byte array.	
Divisions	Bool	Show the divisions of per activity.	
OrganisationTypes	Bool	Show the organisation type details	
CareTypes	Bool	Show the care type details	
SearchTerms	Bool	Show the search terms of the activity type	

Response

The response will contain a list of activities (List<Activity>).

Name	Type	Description	Remarks
Id	GUID	ID of the activity.	
Code	String	Code of the activity.	
Name	String	Description of the activity.	
Description	String	Detailed description of the activity.	
Comments	String	Remarks concerning the activity.	

Location	Address	See 8.8.1.3: Address.Contains the address data of the place where the activity is organised.	
ImageUrl	String	The link on which a picture of the activity can be found.	
Type	ActivityType	Contains an ActivityType object to which the activity belongs.	
Period	Period	Contains the period in which the activity takes place. See point 18.3.2.1	
ClosingDate	Datetime?	Last registration date.	
Price	Decimal	Price of the activity.	
FreePlaces	Int?	Number of free places for the activity via web.	
Days	List<ActivityPart>	List of day parts (List<ActivityPart>) of the activity. See point 0	Dependent on include DayParts.
Options	List<ActivityOption>	List of options (List<ActivityOption>) of the activity. See point 18.3.2.3	Dependent on include Options.
AgeCategory	AgeCategory	Shows the age category. See point 18.3.2.4	
EntryMethod	Bool	Indicates if registrations are made on activity day level (= true) or on activity level itself (= false).	
TotalPlaces	Int?	Maximal number of registrations allowed for this activity via web.	
Agreements	String	Agreements.	
Utilities	String	Utilities.	
WebSalesStart	String	Start date for Web registrations.	
WebSalesStop	String	End date for Web registrations.	
NumberOfDays	Int?	Number of activitydays for this activity	
Division	Division	Location (division) where this activity has been created.	
LastUpdate	DateTime?	Date and time of the last update	
TotalPlacesOverall	Int?	Maximal number of registrations allowed for this activity overall (via web and back office)	
ReservedPlacesOverall	Int?	Current number of registrations for this activity overall (via web and back office)	
OrganisationType	OrganisationType	The organisation type assigned to the activity	
CareType	CareType	The care type assigned to the activity	
UitpasEventId	string	Coupled with this uitpas event id	

Period

Name	Type	Description	Remarks
From	DateTime	Start date of the period.	
Until	DateTime	End date of the period.	

ActivityPart

Name	Type	Description	Remarks
Id	GUID	ID of the day part.	
Period	Period	See point 18.3.2.1:	
Comment	String	Remarks concerning this day part.	
FreePlaces	Int?	Number of free places, 'null' is unlimited, 'zero' is occupied.	
EntryDeadline	DateTime?	Deadline to make entry	
TotalParticipants	Int?	Amount of participants that reserved	
ArrivedParticipants	Int?	Amount of participents that reserved and are present	

HasLinkedTasks	Bool	Indicates if there are any linked EmployeeTasks to the ActivityPart	

ActivityOption

Name	Type	Description	Remarks
Id	GUID	ID of the option.	
ArticleId	GUID	ID of the linked article.	Refers to the Article object.
CustomPrice	Decimal	Price per option.	
Description	String	Description of the option.	
Amount	Int	Number of options.	
MaxOptions	Int	Maximum allowed options	0 = unlimited
MaxOptionsTotal	Int	Maximum allowed options overall (over all entries and persons)	0 = unlimited

AgeCategory

Name	Type	Description	Remarks
Id	GUID	ID of the age category.	
Name	String	Short description of the age category.	
FromAge	Int	Minimal age.	
UntilAge	Int	Maximal age.	
BirthDateFrom	DateTime?	Minimal birthdate limit.	
BirthDateUntil	DateTime?	Maximal birthdate limit.	
Type	int	0 = agecategory by age1 = agecategory by birthdate	
Description	String	Description of the age category.	
Blocked	Bool	Indicates if AgeCategory is blocked	

OrganisationType

Name	Type	Description	Remarks
Id	GUID	ID of the organisation type	
Code	String	Code/short name of the organisation type	
Description	String	Description of the organisation type	
Comment	String	Comment of the organisation type	

CareType

Name	Type	Description	Remarks
Id	GUID	ID of the care type	
Code	String	Code/short name of the care type	
Description	String	Description of the care type	
Comment	String	Comment of the care type	

FindActivitiesAvailablePlaces

This service visualises a list of activities.

Request

The request consists of the service context and an ActivitySearchCriteria object.
This criterion contains an object to take care of paging in case of long lists and different objects to generate the desired list of activities.

ActivitySearchCriteria

Name	Type	Description	Remarks
Paging	PagingCriteria	See 4.2: Paging Criteria	
Period	Period	Finds the activities organised within this period. See point 18.3.2.1	
AgeFrom	int?	Activities that are intended for persons older than or equal to this age.	
AgeTo	Int?	Activities that are intended for persons younger than or equal to this age.	
ActivityId	GUID	Calls up 1 activity only with an ID equal to this value.	
ActivityTypeId	GUID	Calls up the activities that belong to this type.	
Keyword	String	Calls up the activities of which the name contains this text.	
PersonId	GUID	If filled in, the activity price valid for this person will be called up.	
Includes	ActivityIncludes	Determines which parts of the Activity will be included.	
FilterDays	Bool	Enable the filter by day?	
FilterMonday	Bool	Include activities on mondays	
FilterTuesday	Bool	Include activities on Tuesdays	
FilterWednesday	Bool	Include activities on Wednesdays	
FilterThursday	Bool	Include activities on Thursdays	
FilterFriday	Bool	Include activities on Fridays	
FilterSaturday	Bool	Include activities on Saturdays	
FilterSunday	Bool	Include activities on Sundays	
ActivityTypeIdList	List<Guid>	Calls up the activities that belong to this list of activity types	
UpdatedSince	DateTime?	Find the activities that are updated since a certain date	
OrganisationTypeId	GUID	Calls up the activities that belong to this organisation type	
OrganisationTypeIdList	List<Guid>	Calls up the activities that belong to this list of organisation types	
CareTypeId	GUID	Calls up the activities that belong to this care type	
CareTypeIdList	List<Guid>	Calls up the activities that belong to this list of care types	
ActivityTypeSearchTermId	GUID	Calls up the activities that belong to an activity type which has this search term associated with it	
ActivityTypeSearchTermIdList	List<Guid>	Calls up the activities that belong to an activity type which has at least one of this list of search terms associated with it	
LocationId	GUID	Calls up the activities that have this location	
LocationIdList	List<Guid>	Calls up the activities that have a location in this list	

ActivityIncludes

This object allows to exclude certain parts of an activity.

Name	Type	Description	Remarks
ImageUrl	Bool	Show the URL of the image or not.	
DayParts	Bool	Show list of day parts (List<ActivityPart>) per activity.	
Options	Bool	Show list of options (List<ActivityOption>).	
Image	Bool	Show the image byte array.	
Divisions	Bool	Show the divisions of per activity.	
OrganisationTypes	Bool	Show the organisation type details	

CareTypes	Bool	Show the care type details	
SearchTerms	Bool	Show the search terms of the activity type	

Response

The response will contain a list of activities (List<Activity>).

Name	Type	Description	Remarks
AvailablePlaces	ActivityAvailablePlaces	The available places object for an activity	

Period

Name	Type	Description	Remarks
From	DateTime	Start date of the period.	
Until	DateTime	End date of the period.	

ActivityPart

Name	Type	Description	Remarks
Id	GUID	ID of the day part.	
Period	Period	See point 18.3.2.1:	
Comment	String	Remarks concerning this day part.	
FreePlaces	Int?	Number of free places, 'null' is unlimited, 'zero' is occupied.	
EntryDeadline	DateTime?	Deadline to make entry	

ActivityOption

Name	Type	Description	Remarks
Id	GUID	ID of the option.	
ArticleId	GUID	ID of the linked article.	Refers to the Article object.
CustomPrice	Decimal	Price per option.	
Description	String	Description of the option.	
Amount	Int	Number of options.	
MaxOptions	Int	Maximum allowed options	0 = unlimited
MaxOptionsTotal	Int	Maximum allowed options overall (over all entries and persons)	0 = unlimited

AgeCategory

Name	Type	Description	Remarks
Id	GUID	ID of the age category.	
Name	String	Short description of the age category.	
FromAge	Int	Minimal age.	
UntilAge	Int	Maximal age.	
BirthDateFrom	DateTime?	Minimal birthdate limit.	
BirthDateUntil	DateTime?	Maximal birthdate limit.	
Type	int	0 = agecategory by age1 = agecategory by birthdate	
Description	String	Description of the age category.	

OrganisationType

Name	Type	Description	Remarks
Id	GUID	ID of the organisation type	
Code	String	Code/short name of the organisation type	
Description	String	Description of the organisation type	
Comment	String	Comment of the organisation type	

CareType

Name	Type	Description	Remarks
Id	GUID	ID of the care type	
Code	String	Code/short name of the care type	
Description	String	Description of the care type	
Comment	String	Comment of the care type	

ActivityAvailablePlaces

Name	Type	Description	Remarks
ActivityId	GUID	ID of the Activity	
Availableplaces	int	The amount of available places for the Activity	

FindActivitytypesearchterms

This service generates a list of ActivityTypeSearchTerm objects, used to do filtering on the activity list

Request

The request consists of the service context and an ActivityTypeSearchTermSearchCriteria object.

CareTypeSearchCriteria

Name	Type	Description	Remarks
ActivityTypeSearchTermId	GUID	Calls up all search terms with this ID. Set this as an empty Guid to get all search terms	Obligatory.
Keyword	String	Keyword to search for. Leave empty to ignore.	Obligatory.

Response

The response will contain a list of Activity type search terms (List<ActivityTypeSearchTerms>).

FindCARETypes

This service generates a list of CareType objects, used to do filtering on the activity list

Request

The request consists of the service context and an CareTypeSearchCriteria object.

CareTypeSearchCriteria

Name	Type	Description	Remarks
CareTypeId	GUID	Calls up all CareTypes with this ID. Set this as an empty Guid to get all CareTypes	Obligatory.
Keyword	String	Keyword to search for. Leave empty to ignore.	Obligatory.

Response

The response will contain a list of Care types (List<CareType>).

FindEntryOverview

This service generates a list of Entry objects, these are the registrations of a specific person.

Request

The request consists of the service context and an EntrySearchCriteria object.

EntrySearchCriteria

Name	Type	Description	Remarks
Paging	PagingCriteria	See 4.2: Paging Criteria	
PersonId	GUID	Calls up all registrations of which this person is either the main customer, either the participant.	Obligatory.
Until	DateTime?	The date until when the payed entries must be returned. Entries which aren't payed are still returned	Optional

Response

The response will contain a list of registrations (List<Entry>).

Entry

Name	Type	Description	Remarks
Id	GUID	ID of the Entry.	
ActivityId	GUID	ID of the activity.	
ActivityCode	String	Short code of the activity.	
ActivityDesc	String	Description of the activity.	
EntryDate	DateTime	Registration date of the registration.	
EntryNo	Long	Registration number.	
Price	Decimal	Price VAT incl. of the price article, linked to a waiting list.	
Status	Int	Status of the registration.	0 = temporarily 1 = waiting list 2 = reserved 3 = cancelled
Deletable	Bool	Indicates if the registration can be cancelled.	
ActivityStartDate	DateTime	Start date of the activity	
ActivityEndDate	DateTime	End date of the activity	
Nota	String	Nota of the entry	
CustomerId	Guid	Unique ID of the customer who created the entry	
HasPayed	bool	Indicates if the entry is already paid	
WebSalesStart	DateTime?	The start date of the websales	Optional
WebSalesStop	DateTime?	The end date of the websales	Optional
ActivityWebSalesPeriod	ActivityWebSalesPeriod	A period where web sales are available	

ActivityWebSalesPeriod

Name	Type	Description	Remarks
Id	Guid	Id of the ActivityWebSalesPeriod.	
ActivityId	Guid	Id of the activity.	
WebSalesStart	DateTime?	The start date of the websales	Optional
WebSalesStop	DateTime?	The end date of the websales	Optional

FindOrganizationTypes

This service generates a list of OrganisationType objects, used to do filtering on the activity list

Request

The request consists of the service context and an OrganisationTypeSearchCriteria object.

OrganisationTypeSearchCriteria

Name	Type	Description	Remarks
OrganisationTypeId	GUID	Calls up all OrganisationTypes with this ID. Set this as an empty Guid to get all OrganisationTypes	Obligatory.
Keyword	String	Keyword to search for. Leave empty to ignore.	Obligatory.

Response

The response will contain a list of organisation types (List<OrganisationType>).

FindPriceGroupActivityByActivity

This service visualises a list of price groups with prices that are linked to an activity. The Webservices allow to have the price of a registration calculated in 3 different ways:

1. On the basis of the activity price = 1 price set per activity
2. On the basis of the customer's price group
3. The customer can choose a price group out of a list.

For the 3th point we need the method FindPriceGroupActivityByActivity.

Request

The request consists of the service context and a PriceGroupActivitySearchCriteria object.
This criterion only contains an ActivityId that is obligatory.

ActivitySearchCriteria

Name	Type	Description	Remarks
ActivityId	GUID	ID of the activity for which you want to find the price group/prices.	

Response

The response will contain a list of activity groups (List<PriceGroupActivity>).

Name	Type	Description	Remarks
Id	GUID	ID of the activity price group.	
ActivityId	GUID	ID of the activity.	
PriceGroupId	GUID	ID of the price group.	
Price	Decimal?	Price for the combination price group - activity	
From	DateTime?	Activity price group is valid as of this date.	
Until	DateTime?	Activity price group is valid until this date.	
DayOfWeek	WeekDay?	Enumeration of the day of the week on which the activity price group is active. • EveryDay • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday • Sunday	
PriceGroupDesc	String	Description of the price group.	
PriceGroupSn	String	Code of the price group.	

GetActivityDayParticipants

Request

This request consists of a Criteria object and the service context.

Criteria

It is mandatory to specify the ActivityDayId.

Name	Type	Description
ActivityDayId	Guid	The ID of the ActivityDay you wish to see the participants of
Includes	ActivityDayIncludes	Determines which data are included in the response and/or which extra records will be present in the returnset.

Response

Name	Type	Description
Id	Guid	The ID of the ActivityDayParticipant
Attendance	bool	Boolean that indicates whether the user is registered as present on the activity day. True = present, false = absent
EntryParticipant	EntryParticipant	Information about the EntryParticipant
ParticipantInfo	ParticipantInfo	Information about the Participant

EntryParticipant

Name	Type	Description
Id	Guid	The ID of the EntryParticipant
EntryId	Guid	The ID of the Entry
PersonId	Guid	The ID of the Person
PersonName	PersonName	The Name of the Person
Remarks	String	Remarks of the EntryParticipant

ParticipantInfo

Name	Type	Description
Id	Guid	The ID of the Participant
FirstName	String	The first name of the Participant
LastName	String	The last name of the Participant

Gender	Gender	The gender of the Participant.
		Male = 0
		Female = 1
		Unknown = 2

Age	Int	The age of the Participant
BirthDate	DateTime	The birth date of the Participant
Phone	String	The phone of the Participant
Phone2	String	The second phone of the Participant
Cellphone	String	The cellphone of the Participant
AllergyInfo	String	Allergy info of the Participant

MedicationInfo	String	Medication info of the Participant
OtherMedicationInfo	String	Other medication info of the Participant
AllowedToPhotograph	Bool	Participant is allowed to be photographed or not

ActivityIncludes

Name	Type	Description
IncludeBookedTemporarily	Bool	Includes all temporarily booked day activity participant items
IncludeOnWaitingList	Bool	Includes all day activity participant items that are marked on waiting list
IncludeBooked	Bool	Includes all booked (=reserved) items
IncludeCancelled	Bool	Includes all cancelled day activity participant items

ListActivityTypes

This service visualises a list of activity types.

Request

The request only consists of the service context.

Response

The response will contain a list of activity types (List<ActivityType>).

Name	Type	Description	Remarks
Id	GUID	ID of the activity type.	
Code	String	Code of the activity type.	
Name	String	Description of the activity type.	
SearchTerms	List<ActivityTypeSearchTerm>	List of search terms of the activity type	

UpdateParticipantPrescence

This API call updates the presence of a Participant on an ActivityPart.

Request

This request consists of a Criteria object and the service context.

Criteria

It is mandatory to specify a Present field.

There are 2 ways to update the correct EntryParticipantActivityDay:

- Specify the EntryParticipantActivityDayId
- Specify the ActivityDayId & EntryParticipantId

Name	Type	Description
`ActivityDayId`	Guid	The ID of the ActivityDay you wish to update the presence of
`EntryParticipantId`	Guid	The ID of the EntryParticipant you wish to update the presence of
`EntryParticipantActivityDayId`	Guid	The ID of the EntryParticipantActivityDay you wish to update the presence of
`Present`	Bool	Indicates whether you wish to set the EntryParticipantActivityDay to present or not

Response

ValidationResult

Name	Type	Description	Remarks
BrokenRuleName	String	The code of the error message.	
Message	String	The standard error.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	

Examples

Example 1:

Use EntryParticipantActivityDayId.

Set Present = true

```
{
  "Criteria": {
    "EntryParticipantActivityDayId": "F6737072-DA20-4EC0-90B3-
1369B970A859",
    "Present": true
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000730"
  }
}
```

Example 2:

Use EntryParticipantActivityDayId

Set Present = false

```
{
  "Criteria": {
    "EntryParticipantActivityDayId": "F6737072-DA20-4EC0-90B3-
1369B970A859",
    "Present": false
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000730"
  }
}
```

Example 3:

Use ActivityDayId & EntryParticipantId

Set Present = true

```
{
  "Criteria": {
    "ActivityDayId": "3B13917B-3973-EA11-922A-D46D6D455A91",
    "EntryParticipantId": "A77CE343-3D73-EA11-922A-D46D6D455A91",
    "Present": true
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000730"
  }
}
```

Example 4:

Use ActivityDayId & EntryParticipantId

Set Present = false

```
{
  "Criteria": {
    "ActivityDayId": "3B13917B-3973-EA11-922A-D46D6D455A91",
    "EntryParticipantId": "A77CE343-3D73-EA11-922A-D46D6D455A91",
    "Present": false
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000730"
  }
}
```

Example 5:

Use non existant ID's

=> Validation Error

19 After-school child care module (as of RCX 5.1.0)

AuthenticateChildCareCentreEmployee

This method is used to validate the credentials of a employee for a specific child care centre.

Request

Within the AuthenticateChildCareCentreEmployeeRequest only a **ChildCareCentreEmployeeCredential** object is expected in addition to the service context object.

The **ChildCareCentreEmployeeCredential** object contains the following properties:

Name	Type	Description	Remarks
ChildCareCentreId	Guid	The selected childcare centre	
Username	string	The username of the employee	
Password	string	The password of the employee	

Response

AuthenticateChildCareCentreEmployeeResponse contains an AuthenticationResult that indicates if authentication succeeded or not.

Name	Type	Description	Remarks
HasSucceeded	Boolean	Indicates if the user has been successfully authenticated or not.	
Error	AuthenticationError	An enumeration that returns the error of the AuthenticationProcess. Possible values are: None InvalidParameters UnknownUser BlockedUser InvalidCredential	For security purposes no correct information is given here.
PersonId	GUID	The ID of the authenticated user, in this case a child care centre.	

Child care – Brief overview

The module after-school child care is available as of ReCreateX 5.1.0.
The system functions as follows:

- A child has to be entered in ReCreateX.
- Once a child has been registered in a certain care centre, presences can be registered using the Webservices.

The Webservice methods that have been created for this purpose are the following:

- AuthenticateChildCareCentre (authenticate a child care centre)
- FindChildCareCentres (find child care centres)
- FindChildCareCentrePeriods (find care periods)
- FindChildCareCentreEntries (find registrations)
- FindChildCareCentrePresences (find presences)
- SaveChildCareCentrePresence (find one presence)
- DeleteChildCareCentrePresence (delete a presence)
- FindChildCareCentreLogbookEntries (find entries in a persons logbook) (from RCX 5.3.0)
- SaveChildCareCentreLogbookEntry (saves entry in a persons logbook) (from RCX 5.3.0)
- DeleteChildCareCentreLogbookEntry (removes an etry in a persons logbook) (from RCX 5.3.0)

DeleteChildCareCentreLogbookEntry

Used to find delete an entry from the childcare logbook.

Request

In the request you pass the ID of the entry to be deleted.

Response

The response contains a validation result.

DeleteChildCarePresence

This method is used to delete a care presence.

Request

The Request consists of the service context and the unique ID of the presence that can be found using for example the FindChildCareCentrePresences method.

Name	Type	Description	Remarks
ChildCareCentrePresenceId	GUID	ID of the care presence.	

Response

The response contains a **ValidationResult** object. This object looks as follows:

Name	Type	Description	Remarks
BrokenRuleName	String	Name of the validation rule.	
Message	String	Error message.	

IsValid	Bool	Rule has been successfully processed.	True = success
			False = error

FindChildCareBenefit

Is used to call up a list of child care benefits.

Request

The FindChildCareBenefitsRequest contains on the one hand the standard service context object and on the other hand the ChildCareBenefitSearchCriteria object described below.

Name	Type	Description	Remarks
Id	GUID?	If filled in: The ID of the benefit one wants to find.	Nullable
Name	string	The (part of the) name of the benefit.	

Response

The FindChildCareBenefitsResponse object contains a list of ChildCareBenefit objects that meet the criteria above. The ChildCareBenefit object looks as follows:

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the parent statute.	
Code	String	Code of the parent benefit.	
Description	String	Name of the parent benefit.	

FindChildCareCentreAbsences

Is used to call up a list of child care absences.
Child care absences are children who do not turn up for periods they were registered for.

Request

The FindChildCareCentreAbsencesRequest contains both the standard service context object and a ChildCareCentreAbsenceSearchCriteria object.

ChildCareCentreAbsenceSearchCriteria

Name	Type	Description	Remarks
Id	Guid?	If filled in: The ID of the absence one wants to find.	This equals the ID of the care period entry in RCX (the period he is registered for)
PersonId	Guid?	If filled in: The ID of the person of whom you want to call up the absences.	
ChildCareCentre Id	Guid?	If filled in: The ID of the care centre of which you want to call up the absences.	
ChildCareCentre PeriodId	Guid?	If filled in: The ID of the care period of whom you want to call up the absences.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

From	DateTime?	If filled in: The from-date to find absences for	This compares the provided date with the care period date
Until	DateTime?	If filled in: The to-date to find absences for	This compares the provided date with the care period date
Includes	ChildCareCentreAbsenceIncludes	Include certain objects in the response	

ChildCareCentreAbsenceIncludes

The ChildCareCentreAbsenceIncludes object contains following properties:

Name	Type	Description	Remarks
Centres	Bool	Include the detail information for child care centres	
CentrePeriods	Bool	Include the detail information for child care centre periods	

Response

The FindChildCareCentreAbsencesResponse object contains a list of ChildCareCentreAbsence objects that meet the criteria above.µ

ChildCareCentreAbsence

Name	Type	Description	Remarks
Id	Guid	The ID of the care absence	This equals the ID of the care period entry in RCX (the period he is registered for)
PersonId	Guid	The ID of the person	
ChildCareCentreId	Guid	The ID of the care centre	
ChildCareCentrePeriodId	Guid	The ID of the care period	
ChildCareCentre	ChildCareCentre	The care centre details	Only filled when Centres include is activated
ChildCareCentrePeriod	ChildCareCentrePeriod	The care centre period details	Only filled when CentrePeriods include is activated

FindChildCareCentreEntries

Is used to call up a list of child care registrations.

Request

The FindChildCareCentreEntriesRequest contains on the one hand the standard service context object and on the other hand the ChildCareCentreEntrySearchCriteria object described below.

Name	Type	Description	Remarks
Id	GUID?	If filled in: The ID of the registration one wants to find.	Nullable
PersonId	GUID?	If filled in: The ID of the person of whom you want to call up the registrations.	
ChildCareCentreId	GUID?	If filled in: The ID of the care centre of which you want to call up the registrations.	
ChildCareCentre PeriodId	GUID?	If filled in: The ID of the care period of which you want to call up the registrations.	Only applicable if you work with registrations per period in ReCreateX.
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	ChildCareCentreEntry Includes	Determines which parts of the registration will be included.	
Ids	IList<Guid>	If filled in: The IDs of the registrations one wants to find	

ChildCareCentreEntryIncludes

This object allows to exclude certain parts of the registrations.

Name	Type	Description	Remarks
PersonDetails	Bool	Calls up the detailed data of the person or not.	
PersonContacts	Bool	Calls up the contact persons of the person or not.	
PersonParentRelations	Bool	Calls up the parent relations of the person or not.	
Periods	Bool	Calls up the periods or not.	
Doctor	Bool	Calls up the doctor of the person or not.	
Image	Bool	Calls up the image and imageurl of the person or not	
ParentContacts	Bool	Calls up the contacts of the parent relations	

Response

The FindChildCareCentreEntriesResponse object contains a list of ChildCareCentreEntry objects that meet the criteria above. The ChildCareCentreEntry object looks as follows.

ChildCareCentreEntry

Name	Type	Description	Remarks
Id	GUID	The ID of the care registration.	
PersonId	GUID	Unique ID of the registered person.	
Person	Person	Detailed data of the registered person.	
ChildCareCentreId	GUID	The ID of the care centre for which the person is registered.	
EntryDate	DateTime	Registration date.	
ChildCareEntryPeriod	ChildCareEntryPeriod	Details concerning the registration period for the registration.	
Periods	List<ChildCareCentrePeriodEntry>	Defines all periods the person is registered for.	
Deletable	Bool	Defines whether the registration can be cancelled.	
Status	Int	Defines the status for the registration. • -1 = not defined • 0 = registered temporary • 1 = registered definite • 2 = cancelled	
AmountPayed	Decimal	The amount that was already payed for this registration.	
ChildCareCentre	ChildCareCentre	Detailed data of the care centre for which the person is registered.	

ChildCareEntryPeriod

The ChildCareEntryPeriod object contains details concerning the period for which a registration is valid. (example: a school year = 1-Sept-2012 unitil 31-Aug-2013).

Name	Type	Description	Remarks
Id	GUID	The ID of the registration period.	
Code	String	The code of the registration period.	
Name	String	The name of the registration period.	
From	DateTime	Start date of the registration period.	
Until	Datetime	End date of the registration period.	

FindChildCareCentreLogbookEntries

Used to find entries in a persons childcare logbook.

Request

You pass a ChildCareCentreLogbookSearchCriteria object in the request. This object contains the following information.

Name	Type	Description	Remarks

Id	GUID?	If filled in: The ID of the logbook entry	Nullable
PersonId	GUID?	If filled in: The ID of the person of whom you want to call up the logbook entries	
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The response contains a list of ChildCareCentreLogbookEntry objects. These contain the following information:

Name	Type	Description	Remarks
Id	GUID	Entry ID	
PersonId	GUID	The Person ID	
Text	String	The logbook entry's text	
CreationDate	DateTime	The date this logbook entry was created	
LastModificationDate	DateTime	The date this logbook entry has last been modified	

FindChildCareCentrePeriods

Is used to call up a list of child care periods.

Request

The FindChildCareCentrePeriodsRequest contains on the one hand the standard service context object and on the other hand the ChildCareCentrePeriodSearchCriteria object described below.

Name	Type	Description	Remarks
Id	GUID?	If filled in: The ID of the care period one wants to find.	Nullable
ChildCareCentreId	GUID?	If filled in: The ID of the care centre of which one wants to find the periods.	
From	DateTime?	Only call up the periods that end after the given date.	
Until	DateTime?	Only call up the periods that start before the given date.	
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The FindChildCareCentrePeriodsResponse object contains a list of ChildCareCentrePeriod objects that meet the criteria above. The ChildCareCentrePeriod object looks as follows:

Name	Type	Description	Remarks
Id	GUID	The ID of the care period.	
ChildCareCentreId	GUID	The ID of the care centre to which this period belongs.	
From	DateTime	The start time of the care period.	
Until	DateTime	The end time of the care period.	
MaxChildren	Int	The maximal allowed number of children.	
Comment	String	Extra remarks concerning this care period.	

FindChildCareCentrePresenceHeaders

Is used to call up a list of care presences that correspond to the sales within ReCreateX. The presences are shown grouped per child, per day and per care centre and contain all details concerning the amounts.

Request

The FindChildCareCentrePresenceHeadersRequest contains on the one hand the standard service context object and on the other hand the ChildCareCentrePresenceHeaderSearchCriteria object described below.

ChildCareCentrePresenceHeaderSearchCriteria

Name	Type	Description	Remarks

Id	GUID?	If filled in: The ID of the care registration one wants to find.	Nullable
PersonId	GUID?	If filled in: The ID of the person of whom you want to call up the presences.	
ChildCareCentreId	GUID?	If filled in: The ID of the care centre of which you want to call up the presences.	
From	DateTime?	Call up only the presences for days later than or equal to this date.	
Until	DateTime?	Call up only the presences for days earlier than or equal to this date.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	ChildCareCentrePresenceHeaderIncludes	Determines which details of the presences will be included.	

ChildCareCentrePresenceHeaderIncludes

This object allows to exclude certain parts of the registrations.

Name	Type	Description	Remarks
Tariffs	Bool	Finds the tariffs that have been used for the presences.	

Presences	Bool	Finds the underlying presences or not.	Header = grouped per day
			Presences =details per presence

PresenceOptions	Bool	Finds the corresponding options for all underlying presences or not.	Only applicable if Presences = True
Centres	Bool	Finds the linked child care centres	
CentrePeriods	Bool	Finds the corresponding periods for all underlying presences or not.	Only applicable if Presences = True

Response

The FindChildCareCentrePresenceHeadersResponse object contains a list of ChildCareCentrePresenceHeader objects that meet the criteria above. The ChildCareCentrePresenceHeader object looks as follows:

ChildCareCentrePresenceHeader

Name	Type	Description	Remarks
Id	GUID	The ID of the care presence.	
PersonId	GUID	The ID of the person.	
ChildCareCentreId	GUID	The ID of the care centre.	
Day	DateTime	Day of the presence registration.	
PresenceLineAmount	Decimal	Amount per unit excl. VAT for presences.	
PresenceVATAmount	Decimal	VAT amount per unit for presences.	
PresenceAmount	Decimal	Unit amount incl. VAT for presences.	
PresenceQuantity	Int	Quantity presences.	
FineLineAmount	Decimal	Amount per unit excl. VAT for fines.	
FineVATAmount	Decimal	VAT amount per unit for fines.	
FineAmount	Decimal	Unit amount incl. VAT for fines.	
FineQuantity	Int	Quantity fines.	
TotalPresenceAmount	Decimal	Total amount for the presences.	
TotalFineAmount	Decimal	Total amount for the fines.	
TotalPAmount	Decimal	Total amount invoiced (presences + fines).	
ChildCareTariff	ChildCareTariff	Tariff used for calculating the prices of the sales registration.	See 19.10.2.2:
Presences	IList<ChildCareCentrePresences>	List of presences that have been charged.	See 19.6.2:

ChildCareTariff

Name	Type	Description	Remarks
Id	GUID	The ID of the care tariff.	
Code	String	The code of the care tariff.	
Name	String	The name of the care tariff.	
Description	String	Description of the care tariff.	

FindChildCareCentrePresences

Is used to call up a list of child care presences.

Request

FindChildCareCentrePresencesRequest

The FindChildCareCentrePresencesRequest contains on the one hand the standard service context object and on the other hand the ChildCareCentrePresenceSearchCriteria object described below.

Name	Type	Description	Remarks
Id	GUID?	If filled in: The ID of the care registration one wants to find.	Nullable
PersonId	GUID?	If filled in: The ID of the person of whom you want to call up the presences.	
ChildCareCentreId	GUID?	If filled in: The ID of the care centre of which you want to call up the presences.	
ChildCareCentrePeriodId	GUID?	If filled in: The ID of the care period of whom you want to call up the presences.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
From	DateTime?	If filled in: The from-date to find presences for	
Until	DateTime?	If filled in: The to-date to find presences for	
Includes	ChildCareCentrePresenceIncludes	Include certain objects in the response	
PostedSelectionType	PostedSelectionType	Find posted or un-posted presences ▪ All ▪ UnPosted ▪ Posted	

ChildCareCentrePresenceIncludes

The ChildCareCentrePresenceIncludes object contains following properties:

Name	Type	Description	Remarks
Centres	Bool	Include the detail information for child care centres	
CentrePeriods	Bool	Include the detail information for child care centre periods	
Options	Bool	Include the detail information for precense options	

Response

The FindChildCareCentrePresencesResponse object contains a list of ChildCareCentrePresence objects that meet the criteria above. The ChildCareCentrePresence object looks as follows:

Name	Type	Description	Remarks
Id	GUID	The ID of the care presence.	
PersonId	GUID	The ID of the person.	
ChildCareCentreId	GUID	The ID of the care centre.	
ChildCareCentrePeriodId	GUID	The ID of the care period.	
From	DateTime	Start time of the presence.	
Until	DateTime	End time of the presence.	
Posted	Bool	Has the presence already been posted?	If yes, it can no longer be edited.

FindChildCareCentres

Is used to call up a list of child care centres.

Request

The FindChildCareCentresRequest contains on the one hand the standard service context object and on the other hand the ChildCareCentreSearchCriteria object described below.

Name	Type	Description	Remarks
Id	GUID?	If filled in: The ID of the care centre one wants to find.	Nullable
Name	String	If filled in: The name of the care centre one wants to find.	
IsBlocked	Bool?	If filled in: True to call up only the blocked care centres. False to call up only the care centres that are not blocked	Nullable
Paging	PagingCriteria	See 4.2: Paging Criteria	
Includes	ChildCareCentreIncludes	Determines which parts of the care centre will be included.	

ChildCareCentreIncludes

This object allows to exclude certain parts of the care centres.

Name	Type	Description	Remarks
Locations	Bool	Include the location of the care centre or not.	
Options	Bool	Include the options of the care centre or not.	
AllowedEntryPeriods	IList<Guid>	Include the list of all allowed entry periods for the care centre or not.	

Response

The FindChildCareCentresResponse object contains a list of ChildCareCentre objects that meet the criteria above. The ChildCareCentre object looks as follows:

Name	Type	Description	Remarks
Id	GUID	The ID of the care centre.	
Code	String	The code of the care centre.	
Name	String	The name of the care centre.	
Location	AddressLocation	The location of the care centre.	See 10.4.2 for the AddressLocation object.
MaxChildren	Int	The maximal allowed number of children.	
IsBlocked	Bool	True if the care centre is blocked.	
Options	IList<ChildCareCentreOption>	The list of care centre options.	
AgeCategory	AgeCategory	The age category for the care centre.	
UseEntriesOnPeriods	Bool	True if the care centre uses entries on periods. False if the care centre uses entries on centres (without periods).	
AllowedEntryPeriodIds	IList<Guid>	The list of IDs for the allowed entry periods.	
Comments	String	Extra comments to display.	In HTML format
NeedExtraInformationFromUser	Bool	Indicates if the child care centre requires extra information from users	
IsWaitingListEnabled	Bool	Indicates if the waiting list is enabled for this child care centre	
AllowUnregisteredPresences	Bool	Indicates if unregistered presences may be recorded	

ChildCareCentreOption

Name	**Type**	**Description**	**Remarks**
Id	Guid	The unique identifier of the option	
ArticleId	Guid	The article linked with the option	
ChildCareCentreId	Guid	The child care centre linked with the option	
CustomPrice	decimal	The custom price of the option	

Description	string	The description of the option	
IsSelected	bool	Is the option selected?	
Amount	int	The amount of the option	
MaxOptions	int	The maximum options	
Activeperiods	List<ChildCareCentreOptionActivePeriod>	Indicates in which periods the option is allowed	
MaxOptionsTotal	int	The maximum total options	
AgeCategory	AgeCategory	The linked age category of the option	

FindChildCareParentStatuTes

Is used to call up a list of child care parentstatutes.

Request

The FindChildCareParentStatutesRequest contains on the one hand the standard service context object and on the other hand the ChildCareParentStatuteSearchCriteria object described below.

Name	Type	Description	Remarks
Id	GUID?	If filled in: The ID of the ParentStatute one wants to find.	Nullable
Name	string	The (part of the) name of the school.	

Response

The FindChildCareParentStatutesResponse object contains a list of ChildCareParentStatute objects that meet the criteria above. The ChildCareParentStatute object looks as follows:

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the parent statute.	
Code	String	Code of the parent statute.	
Description	String	Name of the parent statute.	

FindChildCareschools

Is used to call up a list of child care school.

Request

The FindChildCareSchoolsRequest contains on the one hand the standard service context object and on the other hand the ChildCareSchoolSearchCriteria object described below.

Name	Type	Description	Remarks
Id	GUID?	If filled in: The ID of the school one wants to find.	Nullable
Name	string	The (part of the) name of the school.	

Response

The FindChildCareSchoolsResponse object contains a list of ChildCareSchool objects that meet the criteria above. The ChildCareSchool object looks as follows:

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the school.	
Code	String	Code of the school.	
Name	String	The name of the school.	
Location	AddressLocation	Location of the school.	See 0: Response

GetChildCareCentrePresenceOptions

Is used to call up a list of presence options for available children

Request

The request contains on the one hand the standard service context object and on the other hand the ChildCareCentrePresenceOptionSearchCriteria object described below.

Name	Type	Description	Remarks
Id	GUID?	If filled in: The ID of presence option	Nullable
PersonId	GUID?	If filled in: The ID of the person of whom you want to call up the presence options.	
ChildCareCentrePeriodId	GUID?	If filled in: The ID of the period you want to search presence options for	
ChildCareCentrePeriodId	GUID?	If filled in: The ID of the care period of whom you want to call up the presence options for.	

Response

The GetChildCareCentrePresenceOptions object contains a list of ChildCareCentrePresenceOption objects that meet the criteria above. The ChildCareCentrePresenceOption object looks as follows:

Name	Type	Description	Remarks
Id	GUID	The ID of the care presence.	
ChildCareCentrePresenceId	GUID	The ID of the childcarecentrepresence.	
ChildCareCentreEntryOptionId	GUID	The ID of the entry option.	
ChildCareCentreOptionId	GUID	The ID of the option	
Description	String	Description	
PersonId	GUID	The id of the person for this option	
ChildCareCentrePeriodId	GUID	The id of the ChildCareCentrePeriod	
NumberOfOptions	Int	The amount of options the user registered for	
NumberOfOptionsUsed	Int	The actual amout of options the user has used	
MaxOptions	Int	The maximum amount of options that can be used in this ChildCareCentrePresence	
MaxOptionsTotal	Int	The maximum amount of options that can be used by everyone	
NumberOfOptionsAvailable	Int	The amount of options left that can be used	
LineAmount	Decimal	The NET amount payed for the option(s)	
VatAmount	Decimal	The VAT amount payed for the option(s)	
Amount	Decimal	The total amount payed = LineAmount + VatAmount	

GetChildCareChipKnip

The **GetChildCareChipKnip** method is used to retrieve the chip knip for a certain child.

Request

The **ChildCareChipKnipRequest** contains on the one hand the standard service context object and on the other hand the **ChildCareChipKnipSearchCriteria** object which is described below.

Name	Type	Description	Remarks
PersonId	Guid?	If filled in: The ID of the child for which you want to find the chip knip	Nullable
Paging	PagingCriteria	See 4.2: Paging Criteria	

Response

The **FindChildCareCentreChipKnipResponse** object contains a **PersonChipKnip** object that meets the criteria above.

GetChildCareChipKnips

The **GetChildCareChipKnips** method is used to retrieve the chip knip for one or multiple children.

Request

The request contains on the one hand the standard service context object and on the other hand the **ChildCareChipKnipsSearchCriteria** object which is described below.

Name	Type	Description	Remarks
PersonIds	List<Guid>	List of children IDs for which you want to find the chip knip	When not filled, no results will be returned

Response

The response object contains a list of **ChildCareChipKnip** objects that meets the criteria above.
The ChildCareChipKnip object contains following parameters:

Name	Type	Description	Remarks
Id	Guid	The ID of the child corresponding with the PersonChipKnip object	This ID will match one of the IDs that were provided in the request
PersonChipKnip	PersonChipKnip	The chip knip details for the child	See GetPersonChipKnip for more information about the PersonChipKnip object

SaveChildCareCentreLogbookEntry

Used to save entries in a persons childcare logbook.

Request

You pass a ChildCareCentreLogbookSearchCriteria object in the request. This object contains the following information.

Name	Type	Description	Remarks
Id	GUID?	If filled in: The ID of the logbook entry	If no ID is provided, a new log will be created, else you will be updating a previous record
PersonId	GUID	The ID of the person.	
Text	String	The text to write in the logbook entry	

Response

The response contains a validation result.

SaveChildCareCentrePresence

This method is used to save care presences.

Request

The SaveChildCarePresenceRequest contains a ChildCareCentrePresence object. See 19.6.2 for more details concerning this object.

Response

The SaveChildCareCentrePresenceResponse object contains a SaveChildCareCentrePresenceResult object. This object looks as follows:

Name	Type	Description	Remarks
ChildCareCentrePresence	ChildCareCentrePresence	Presence object.	See 19.6.2 for details
ValidationResults	List<ValidationResult>	Details concerning the transaction.	

ValidationResult

Name	Type	Description	Remarks
BrokenRuleName	String	The code of the error message.	
Message	String	The standard error.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	

SaveChildCareCentrePresenceList

This method is used to save a list of child care presences.

Request

The SaveChildCarePresenceListRequest contains a list of ChildCareCentrePresence object. See 19.6.21 for more details concerning this object.

Response

The SaveChildCareCentrePresenceListResponse object contains a list of SaveChildCareCentrePresenceResult object. See 19.6.2 for more details concerning this object.

SaveChildCareCentrePresenceOption

Request

In the request you provide a ChildCareCentrePresenceOption object that must be saved.
Note: only the NumberOfOptionsUsed member will actually be saved. Other values are for validation.

Response

The response consists of a SaveChildCareCentrePresenceOptionResult.

Name	Type	Description	Remarks
PresenceOption	ChildCareCentrePresenceOption	The option	
ValidationResults	Array of ValidationResult	Validation results while saving	

SaveChildCarEExtraInfoParents

Request

In the request you provide a ChildCareExtraInfoParent object that must be saved.

Response

The response consists of a SaveChildCareExtraInfoParentResult.

Name	Type	Description	Remarks
ChildCareExtraInfoParent	ChildCareExtraInfoParent	The extra info about the family situation	
ValidationResults	Array of ValidationResult	Validation results while saving	

SaveChildCareSchool

Request

In the request you provide a ChildCareSchool object that must be saved.

Response

The response consists of a SaveChildCareSchoolResult.

Name	Type	Description	Remarks
ChildCareSchool	ChildCareSchool	The school	
ValidationResults	Array of ValidationResult	Validation results while saving	

ValidateChildCareEntryPriorityRule

Is used to check if a person can use priority rules or not.

Request

The ValidateChildCareEntryPriorityRuleRequestcontains on the one hand the standard service context object and on the other hand the ValidateChildCareEntryPriorityRuleCriteriaobject described below.

ValidateChildCareEntryPriorityRuleCriteria

Name	Type	Description	Remarks
ChildrenIds	IList<Guid>	A list with children	
ChildCareEntryPeriodId	Guid	The child care entry period where the children are subscribing for	
ChildCareCentreId	Guid	The child care centre where the children are subscribing for	

Response

The ValidateChildCareEntryPriorityRuleResponse contains a ValidateChildCareEntryPriorityRuleResult object.

This object inherits from the ValidationResult object.

Name	Type	Description	Remarks
IsBlocking	bool	Indicates when the validation fails if it is because it is a blocking rule or not.	

20 Membership module (as of RCX 5.1.0)

AddCreditCardTokenToDigitalWallet

Adds a Credit Card Token to a specified Digital Wallet

Request

The request consists of the service context and a **AddCreditCardTokenToDigitalWalletCriteria** object.

AddCreditCardTokenToDigitalWalletCriteria

Name	Type	Description	Remarks
DigitalWalletId	GUID	The ID of the Digital Wallet you wish to add the CreditCardToken to	
CreditCardToken	CreditCardToken	The CreditCardToken you wish to add to the Digital Wallet	

CreditCardToken

Name	Type	Description	Remarks
TokenUsage	int	Defines the usage of the token (Single or Multiple)	0 = Single, 1 = Multiple
Token	string	Defines the Token string of the CreditCardToken	
Enabled	bool	Defines whether the CreditCardToken is enabled	
ValidUntil	DateTime	Defines until when the CreditCardToken is valid	
Used	bool	Defines whether the CreditCardToken is used	
TransactionId	string	Defines the TransactionId of the CreditCardToken	
UsedTime	DateTime?	Defines the DateTime when the CreditCardToken was used	
MetaData	string	Defines MetaData of the CreditCardToken	

Samples

Multiple

```
{
  "Criteria": {
    "DigitalWalletId": "5A0DE1E8-85A9-EA11-9232-D46D6D455A91",
    "CreditCardToken": {
      "TokenUsage": 1,
      "Token": "token",
      "Enabled": true,
      "MetaData": "metaData"
    }
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000730"
  }
}
```

Single

```
{
  "Criteria": {
    "DigitalWalletId": "5A0DE1E8-85A9-EA11-9232-D46D6D455A91",
    "CreditCardToken": {
      "TokenUsage": 0,
      "Token": "token",
      "Enabled": true,
      "ValidUntil": "2020-12-08 16:23:22.973",
      "Used": false,
      "TransactionId": "transactionID",
      "UsedTime": null,
      "MetaData": "metaData"
    }
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000730"
  }
}
```

Response

The response will contain a **AddCreditCardTokenToDigitalWalletResult** object.

AddCreditCardTokenToDigitalWalletResult

The **AddCreditCardTokenToDigitalWalletResult** object looks as follows:

Name	Type	Description	Remarks
Success	Bool	Indicates whether the call succesfully blocked/unblocked the Division Card	
ErrorMessage	String	When success is false, an error message will be specified for more information	

Sample

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
AddCreditCardTokenToDigitalWalletResponse, ReCreateX.WebShop.
WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
AddCreditCardTokenToDigitalWalletResult, ReCreateX.WebShop.Model",
        "success": true,
        "errorMessage": null
    }
}
```

Error Messages

The following values are possible:

- Invalid Digital Wallet ID

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
AddCreditCardTokenToDigitalWalletResponse, ReCreateX.WebShop.
WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
AddCreditCardTokenToDigitalWalletResult, ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "The entered Digital Wallet ID is invalid"
    }
}
```

- Invalid Digital Wallet

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
AddCreditCardTokenToDigitalWalletResponse, ReCreateX.WebShop.
WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
AddCreditCardTokenToDigitalWalletResult, ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "The Digital Wallet could not be found or
is invalid"
    }
}
```

- InvalidCreditCardToken

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
AddCreditCardTokenToDigitalWalletResponse, ReCreateX.WebShop.
WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
AddCreditCardTokenToDigitalWalletResult, ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "The entered Credit Card Token is invalid."
    }
}
```

- InvalidCreditCardTokenTokenUsage

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
AddCreditCardTokenToDigitalWalletResponse, ReCreateX.WebShop.
WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
AddCreditCardTokenToDigitalWalletResult, ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "CreditCardToken TokenUsage must be 0
(Single) or 1 (Multiple)"
    }
}
```

- Exception

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
AddCreditCardTokenToDigitalWalletResponse, ReCreateX.WebShop.
WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
AddCreditCardTokenToDigitalWalletResult, ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "System.NullReferenceException: Object
reference not set to an instance of an object."
    }
}
```

AddPersonCardSubscription

This method adds the given subscription to the given person card.

Request

The request consists of the service context and a **PersonCardSubscriptionCriteria** object.

PersonCardSubscriptionCriteria

Name	Type	Description	Remarks
PersonCardId	GUID	ID of the person card.	
SubscriptionId	GUID	ID of the subscription.	

Response

The response will contain a **ValidationResult** object that defines the result of the transaction.

ValidationResult

Name	Type	Description	Remarks
BrokenRuleName	String	The error code.	
Message	String	The standard error.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	

BlockDivisionCard

Blocks or unblocks a given Division Card

Request

The request consists of the service context and a **BlockDivisionCardCriteria** object.

BlockDivisionCardCriteria

Name	Type	Description	Remarks
Id	GUID	The ID of the Division Card	
Blocked	Bool	Indication if the Division Card should be blocked or not	

Sample

```
{
  "Criteria": {
    "Id": "f536e30a-8802-4111-bbc7-cf16c88076f3",
    "Blocked": true
  },
  "Context": {
    "Language": "en",
    "ShopId": "00000000-0000-0000-0000-000000000730"
  }
}
```

Response

The response will contain a **BlockDivisionCardResult** object.

BlockDivisionCardResult

The **BlockDivisionCardResult** object looks as follows:

Name	Type	Description	Remarks
Success	Bool	Indicates whether the call succesfully blocked/unblocked the Division Card	
ErrorMessage	String	When success is false, an error message will be specified for more information	

Sample

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
BlockDivisionCardResponse, ReCreateX.WebShop.WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.BlockDivisionCardResult,
ReCreateX.WebShop.Model",
        "success": true,
        "errorMessage": null
    }
}
```

Error Messages

The following values are possible:

- Invalid ID

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
BlockDivisionCardResponse, ReCreateX.WebShop.WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.BlockDivisionCardResult,
ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "The entered ID is invalid"
    }
}
```

- Invalid Card

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
BlockDivisionCardResponse, ReCreateX.WebShop.WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.BlockDivisionCardResult,
ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "The card could not be found or is invalid"
    }
}
```

- Exception

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
BlockDivisionCardResponse, ReCreateX.WebShop.WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.BlockDivisionCardResult,
ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "System.NullReferenceException: Object
reference not set to an instance of an object."
    }
}
```

CancelSubscription

This method cancels a subscription

Request

The request consists of the service context and a **CancelSubscriptionCriteria** object.

CancelSubscriptionCriteria

Name	Type	Description	Remarks
SubscriptionId	Guid	The id of the subscription you want to cancel	

Response

The response will contain a **bool** value which indicates if the subscription is sucessfully cancelled or not.

CheckParticipantsAlreadyRegistered

Checks if a Participant is already registered for an Activity

Request

The request consists of the service context and a **CheckParticipantsAlreadyRegisteredCriteria** object.

CheckParticipantsAlreadyRegisteredCriteria

Name	Type	Description	Remarks
ActivityReservation	ActivityReservation	The new ActivityReservation that the Participant wishes to register for	

Response

The response will contain a **CheckParticipantsAlreadyRegisteredResult** object.

CheckParticipantsAlreadyRegisteredResult

The **CheckParticipantsAlreadyRegisteredResult**object looks as follows:

Name	Type	Description	Remarks
ParticipantRegisteredInfos	List<ParticipantRegisteredInfo>	Contains information about the participant's registration validity	

ParticipantRegisteredInfo

The **ParticipantRegisteredInfo** object looks as follows:

Name	Type	Description	Remarks
ParticipantId	Guid	The Id of the Participant	
AlreadyRegistered	Bool	Indicates whether the Participant is already registered or not	
Blocking	Bool	Indiciates whether the Participant is blocked from registering again	

DevaluateSubscription

This method registers a refusation log for a certain person card and reader.

Request

The request consists of the service context and a **DevaluateSubscriptionCriteria** object.

DevaluateSubscriptionCriteria

Name	Type	Description	Remarks
ScannedBarCode	String	The barcode of the card.	
ReaderId	Guid?	Unique identifier of the reader where devaluation needs to occur.	
SubscriptionId	Guid?	Unique identifier of the subscription to devaluate.	
IncludeImage	bool	Indicates whether to include the image in the response or not.	
EmployeeId	Guid?	Unique identifier of the employee that is performing the devaluation.	
Markdowns	int	Number of devaluations to perform	
IgnoreExclusionCalendar	bool	Indicates whether to ignore the exclusion calendar or not.	
ReturnCurrentPersonCard	bool	Only return the person card in the response linked with the scanned barcode in the request	
Includes	DevaluateSubscriptionIncludes		

DevaluateSubscriptionIncludes

Name	Type
CultureEventInfo	bool
AdmissionImage	bool

Response

The response will contain a **DevaluateSubscriptionResult** object that defines the result of the transaction.

DevaluateSubscriptionResult

Name	Type	Description	Remarks
DevaluateSubscriptionState	DevaluateSubscriptionState	The result state ▪ 0 = Ok	

		▪ 1 = NOk ▪ 2 = SelectSubscription	
Message	String	Extra message info	
Subscriptions	IList<Subscription>	List of subscriptions that were devaluated successfully.	
SubscriptionWithInvalidityReasons	IList<SubscriptionWithInvalidityReasons>	List of subscriptions that were not devaluated due to one or more invalidity reasons.	

SubscriptionInvalidityReasons

- InactiveSubscription = 0
- InactiveSetting = 1
- InvalidTurns = 2
- InvalidTurnsPerPeriod = 3
- InvalidPeriod = 4
- InvalidHolidayCalendar = 5
- InvalidSubscriptionCalendar =6
- InvalidActivation = 7
- InvalidSubscriptionMinutes = 8
- InvalidLinkedExposition = 9
- HallForSubscriptionIsInvalid = 10
- ReservationForSubscriptionIsInvalid = 11
- TimeIsInvalid =12
- InvalidAccessGroupCalendar =13
- InvalidAntiPassBack = 14
- InvalidChipKnip = 15
- InvalidCredits = 16
- InvalidMaxTime = 17
- InvalidMinTime = 18
- InvalidCollectionContract = 19
- DigitalWalletHasOutstandingCredit = 20
- DigitalWalletHasInsufficientBalance = 21
- DigitalWalletNotFoundForOverdueMinutes = 22
- LinkedExpositionUnavailable = 23

DisconnectDivisionCard

Disconnect all subscriptions from a given division card

Request

The request consists of the service context and a **DisconnectDivisionCardCriteria** object.

DisconnectDivisionCardCriteria

Name	Type	Description	Remarks
DivisionCardNumber	String	The cardnumber of a divisioncard	
PeripheralStringId	GUID	The peripheralstringid of the divisioncard	

Sample

```
{
  "Criteria": {
    "DivisionCardNumber": "999900010003464",
    "PeripheralStringId": "26CD00C0-A1CA-4D40-8FEB-3A795664D2DD"
  },
  "Context": {
    "Language": "en",
    "ShopId": "00000000-0000-0000-0000-000000000710"
  }
}
```

Response

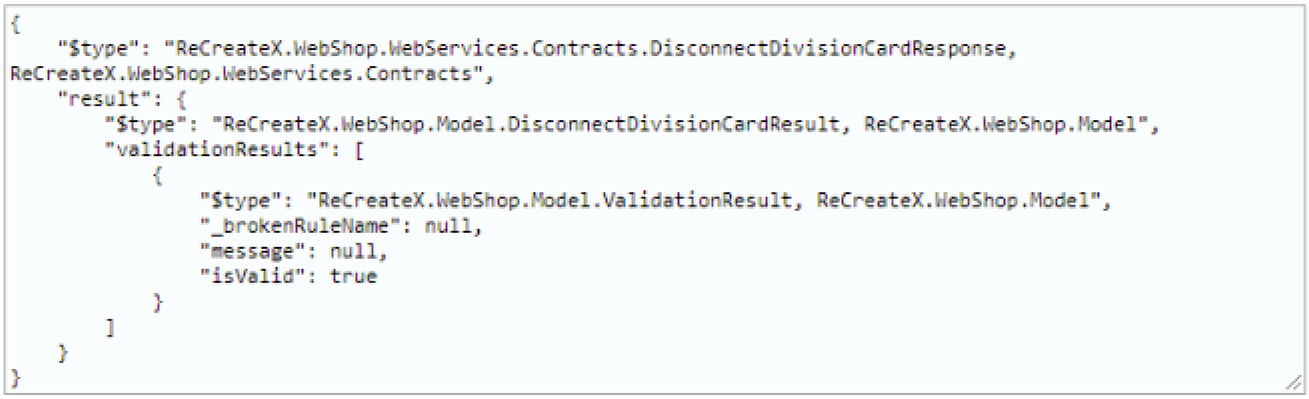
The response will contain a **DisconnectDivisionCardResult** object.

DisconnectDivisionCardResult

The **DisconnectDivisionCardResult** object looks as follows:

Name	Type	Description	Remarks
ValidationResults	ValidationResult	If anything went wrong, this is where the errors will be displayed	

Sample



```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.DisconnectDivisionCardResponse, ReCreateX.WebShop.WebServices.Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.DisconnectDivisionCardResult, ReCreateX.WebShop.Model",
        "validationResults": [
            {
                "$type": "ReCreateX.WebShop.Model.ValidationResult, ReCreateX.WebShop.Model",
                "_brokenRuleName": null,
                "message": null,
                "isValid": true
            }
        ]
    }
}
```

ValidationResults

Type of validation. The following values are possible:

- isValid: true - brokenRuleName: null
  - Everything is correct
- isValid: false - brokenRuleName: PeripheralStringDoesNotMeetTheCardNumber
  - The peripheralStringId does not match the structure of the cardnumber
- isValid: false - brokenRuleName: Division Card Unknown
  - No Division Card can be found with given Division Card Number
- isValid: false - brokenRuleName: Address Card Unknown
  - No Address Card can be found with given Division Card Number (Nothing to disconnect)

FindFamilyMembersForFamilySubscriptionAdjustment

This method finds the family members that must be adjusted for a family subscription.

Request

The request consists of the service context and a **FamilyMembersForFamilySubscriptionAdjustmentCriteria** object.

FamilyMembersForFamilySubscriptionAdjustmentCriteria

Name	Type	Description	Remarks
SubscriptionId	Guid	The id of the subscription you want to adjust	

Response

The response will contain an **array of Guids** which indicates the ID of each family member/person that will be adjusted.

FindPersonSubscriptionData

Finds all the subscription data of one or all persons that changed after a given date.

Request

The request consists of the service context, a **FindPersonSubscriptionData** object and a **Paging** Object.

DisconnectDivisionCardCriteria

Name	Type	Description	Remarks
`PersonId`	Guid	The id of a Person	Will show all persons if this is null
`ModifiedSince`	DateTime	Subscription data changes will be shown from after this datetime	No filtering on date if this is null
Paging	PagingCriteria	Used to manage how	

PagingCriteria

Name	Type	Description	Remarks
`PageIndex`	int	The page index	
`PageSize`	int	The page size	Only accepts pagesizes between 1 and 100. All other PageSizes will be automatically set to 100.
Sorting	SortingCriteria	Always set this as null in this call. This is not used.	Always set this as null in this call. This is not used.

Sample

```
{
  "Criteria": {
    "PersonId": null,
    "ModifiedSince": "2019-03-01 00:00:00",
    "Paging": {
      "PageIndex": 0,
      "PageSize": 5,
      "Sorting": null
    }
  },
  "Context": {
    "Language": "en",
    "ShopId": "00000000-0000-0000-0000-000000000711"
  }
}
```

Response

The response will contain a **FindPersonSubscriptionDataResult** object.

FindPersonSubscriptionDataResult

The **FindPersonSubscriptionDataResult** object looks as follows:

Name	Type	Description	Remarks
PersonSubscriptions	IList<PersonSubscription>	A list of PersonSubscriptions	

PersonSubscription

The **PersonSubscription** object looks as follows:

Name	Type	Description	Remarks
Person	Person	A Person	
Subscriptions	IList<Subscription>	A List of Subscription	

Sample

FindPersonSubscriptionData Result

FindSubscriptionArticles

This method is used to call up a list of Articles of the type Subscription /E-Ticket. An E-Ticket is a subscription of short duration, in other words a day ticket.

Request

The request consists of a service context and a CultureEventSearchCriteria object.
The properties of a CultureEventSearchCriteria look as follows:

SubscriptionArticleSearchCriteria

Name	Type	Description	Remarks
SubscriptionArticleId	Guid?	ID of the article.	
NamePattern	String	A search pattern on name.	
BuyerId	Guid	The ID of the person (buyer). This is required to find the correct price.	
Types	SubscriptionType	0 = Subscription 1 = ETicket 2 = All	0 = regular subscription 1 = E-ticket or day ticket 2 = both
Paging	PaginCriteria		
Includes	SubscriptionArticleIncludes	Additional information	
ForVouchers	Bool?	If true, only the subscription articles that are marked to show in vouchers will be called up.	
Barcode	string	If filled in, only load the articles linked with this barcode	
ArticleGroupId	Guid?	Will only retrieve articles from specific ArticleGroup when specified	
DivisionIds	List<Guid>	Will only return SubscriptionArticles with a division of one of the specified DivisionIds	

SubscriptionArticleIncludes

Name	Type	Description	Remarks
ImageUrl	Bool	Does the url of the image have to be included?	

Image	Bool	Does the image of the article have to be included.	
ArticleGroup	Bool	Does the article group have to be included.	
Attachment	Bool	Do possible attachments have to be included.	
DirectDebits	Bool	Does the direct debit articles have to be included?	
Prices	Bool	Does the prices of the article have to be included?	
LessonGroups	Bool	Does the lesson groups have to be included?	
Barcodes	Bool	Does the barcodes have to be includes?	
IsLessonGroupItem	Bool	Returns only LessonGroupItems	
CardCreations	Bool	Returns the CardCreations of the Item if true	

Response

A list of SubscriptionArticles. SubscriptionArticles are articles of the type Subscription or Eticket and can be used on the same places in the basket as regular Articles. In other words, a SubscriptionArticle inherits from an Article.

SubscriptionArticle

In addition to the standard Article properties the SubscriptionArticle also has the following specific properties.

Name	Type	Description	Remarks
CheckTurns	Bool	Check number of sessions.	
NumberOfTurns	Int?	Number of sessions.	
MinimumTurns	Int?	Minimal sessions.	Generally 0
CheckPeriod	Bool	Check period.	
PeriodStartsOnFirstVisit	Bool	Start period at sale or at first visit.	
PeriodDuraton	Duration	Number of days, weeks or months.	
IsFixedPeriod	Bool?	Is it a fixed period or a period that is determined at sale or at first visit.	
FixedPeriod	Period	From/till date period.	
CheckMaximumDurationVisit	Bool	Check maximal duration of visit.	
MaximumDurationVisit	TimeSpan?	Maximal duration of a visit.	
CheckMinimumDurationNextVisit	Bool	Chick minimal duration until next visit.	
MinimumDurationNextVisit	TimeSpan?	Minimal duration until the next visit.	
CheckTurnsPerVisit	Bool	Check sessions per period.	
TurnsPerPeriod	Int?	Number of sessions per period.	
SubscriptionPeriod	SubscriptionPeriod	Day, week, month etc.	
IsEticket	Bool	Indication that this article type is an ETicket.	
DirectDebitProcessing	Bool	Is this a Direct Debit article?	
DirectDebitAskStartDate	Bool	Does the Direct Debit article needs a start date?	
DirectDebitAskCollectionDay	Bool	Does the Direct Debitt article needs a collection day?	
FamilyItem	Bool	Indicates if the article is a family item	
Prices	IList<SubscriptionArticlePrice>	A collection of type SubscriptionArticlePrice	
IsLessonGroupItem	Bool	Indicates whether the article is a lessongroup article or not.	
LessonGroups	IList<LessonGroup>	All lesson groups linked to the subscription	
AllowMultiplePersonLinking	Bool	Indicates whether multiple persons can be linked to the same subscription or not.	
AllowMultiplePurchasesSubscriptions	Bool	Indicates if multiple subscriptions can be purchased	

Barcodes	IList<string>	The barcodes linked with the article	
AccumulateMemberships	Bool	Indicates if memberships should be accumulated or not.	
CardCreations	List<CardCreation>	Returns the list of configured CardCreations for the Item	
HigherPriceAtSale	Bool	Indicates whether the collection contract price can be set higher	

Duration

Name	Type	Description	Remarks
Days	Int	Number of days.	
Weeks	Int	Number of weeks.	
Months	Int	Number of months.	

SubscriptionPeriod

Name	Type	Description	Remarks
Id	Guid	ID of the subscription period.	
Code	String	Code of the period.	
Name	String	Name of the period.	
Duration	Duration	Days, weeks, months	

SubscriptionArticlePrice

Name	Type	Description	Remarks
Id	Guid	Id of the article price.	
Decimal	Price	Price of the article	

CardCreation

Name	Type	Description	Remarks
Id	Guid	Id of the CardCreation	
ItemId	guid	The Id of the CardCreation's Item	
PeripheralStringId	Guid	The Id of the CardCreation's PeripheralString	
RegistrationItemId	Guid	The id of the CardCreation's RegistrationItem	
Sequence	Guid	The sequence the CardCreations should be displayed in	
Description	String	The description of the CardCreation	
Price	Decimal	The price of the CardCreation	

FindSubscriptionConversionArticles

This method is used to call up a list of Articles of the type Subscription /E-Ticket where a subscription may be converted into.

Request

The request consists of a service context and a SubscriptionConversionArticleCriteria object.
The properties of a SubscriptionConversionArticleCriteria look as follows:

SubscriptionConversionArticleCriteria

Name	Type	Description	Remarks
SubscriptionId	Guid	ID of the subscription	
Paging	PaginCriteria		

Response

A list of SubscriptionConversionArticle.

SubscriptionConversionArticle

Name	Type	Description
Id	Guid	The ID of the article
Code	string	The code of the article
Name	string	The name of the article
Description	string	The description of the article
Prices	IList<SubscriptionArticlePrice>	The prices linked with the article
DirectDebitProcessing	bool	Is it an direct debit article?

FindSubscriptionLogs

This method calls up all loggings of a subscription on the basis of certain search criteria.

Request

The request consists of the service context and a **SubscriptionLogSearchCriteria** object.

SubscriptionLogSearchCriteria

Name	Type	Description	Remarks
SubscriptionLogId	GUID?	ID of a specific logging.	
SubscriptionId	GUID?	ID of the subscription.	
PersonCardId	GUID?	Only the logging that has been created for a specific customer card.	
PersonId	GUID?	Only the logging of a specific person.	
CreatedFrom	DateTime?	Only the loggings after this date.	
CreatedUntil	DateTime?	Only the loggings before this date.	
Paging	PagingCriteria	Paging criteria.	
Includes	SubscriptionLogIncludes	Which data have to be included.	
EmployeeId	Guid?	Unique identifier of the employee	
ReaderId	Guid?	Unique identifier of the reader	
SubscriptionLogType	SubscriptionLogType	The type of log. • NoAccess • Access • SubscriptionUnlinked • Swallowed • SubscriptionLinked	
AccessZoneId	Guid?	Only the loggings of a specific access zone.	

SubscriptionLogIncludes

Name	Type	Description	Remarks
Person	Bool	Include Person object.	
PersonBase	Bool	Include only the PersonBase object. (This contains the id, name, address and birthdate)	

PersonImage	Bool	Include the Picture datamember in the Person object.	Will only be included if Person include is also set to TRUE

Response

The response will contain a list of **SubscriptionLog** objects that meet the criteria.

SubscriptionLog

The **SubscriptionLog** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the logging.	
SubscriptionId	GUID	Linked membership/E-ticket.	
CreationDate	DateTime	Creation date of the logging.	
LogType	SubscriptionLogType	Type of logging.	
SubType	SubscriptionLogSubType	More detailed description of the logging.	
NumberOfTurns	Int	Number of deducted sessions.	
RemainingTurns	Int	Remaining sessions at this logging.	
RemainingDays	Int	Remaining days at creation of this logging.	
Card	Card	Card that caused this logging.	
PersonId	GUID	Customer that created this logging.	
Person	Person	Person object.	
Division	Division	Location (division) where this logging has been created.	
PersonBase	PersonBase	PersonBase object.	

SubscriptionLogType

Type of logging. The following values are possible:

- NoAccess (no access)
- Access (access)
- SubscriptionUnlinked (subscription disconnected from the customer card)
- Swallowed (subscription disconnected and card swallowed)
- SubscriptionLinked (subscription linked to a customer card)

SubscriptionLogSubType

The following values are possible:

- NoAccesInactiveSubscription (no access because subscription was not active)
- NoAccessInvalidTurns (no access due to insufficient sessions)
- NoAccessInvalidPeriod (no access because the visit is not within the access period)
- NoAccessOnHoliday (no access because the visit is on a holiday)
- NoAccessSubscriptionExclusionCalendar (No access because the visit is on an exclusion day of the subscription)
- NoAccessAccessGroupExclusionCalendar (No access because the visit is on an exclusion day of the access group)
- NoAccessInactiveReaderSetting (No access because of inactive setting on the access reader)
- NoAccessInvalidAccessTime (No access because the visit is on an invalid time)
- NoAccessAntiPassBack (No access because of anti-passback)
- NoAccessNoRemainingTurnsPerPeriod. (No access because no more remaining sessions per period)
- NoAccessSubscriptionNotActivated (No access because subscription has not been activated before the visit)
- NoAccessVisitExceedsMaximumTime (No access because the visit was too long)
- NoAccessMinimumTimeNextVisitFailure (No access because the next visit came too fast after the previous one)
- NoAccessLinkedSubscriptionIsInvalid (No access because the linked subscription was not valid)
- NoAccessNoValidPlaceReservationFound (No access because no valid facility booking was found)
- NoAccessNoValidCultureReservationFound (No access because no valid ticketing booking was found)
- NoAccessInsufficientCredits (No access because the customer does not have enough credits)
- NoAccessInsufficiantAmountInPurse (No access because the customer's e-purse balance is insufficient)
- CardCopy (Subscription has been copied to another card)
- CardMove (Subscription has been transferred to another card)

- SubscriptionInsentive (An extra subscription has been created)
- OfflineCounterRegistration
- NoAccessInvalidMuseumCard(No access because the museumcard is invalid)
- NoAccessInvalidCollectionContract (No access because the collection contract is invalid)
- ManualDevaluation
- DigitalWalletHasOutstandingCredit
- DigitalWalletHasInsufficientBalance
- DigitalWalletNotFoundForOverdueMinutes
- DevaluedByOtherSubscriptionComponent
- DivisionCardBlocked
- CarrierSwitchCouple
- CarrierSwitchUncouple
- CarrierSwitchBlock
- CarrierSwitchUnblock
- ZonePresenceMaxReached
- DevaluedDueToDevaluationOfLinkedSubscription
- NoAvailableExpositionTimeSlot

FindSubscriptions

This method calls up all subscriptions on the basis of certain search criteria.

Request

The request consists of the service context and a **SubscriptionSearchCriteria** object.

SubscriptionSearchCriteria

Name	Type	Description	Remarks
SubscriptionId	GUID?	ID of a specific subscription. (If you mean to call up only 1 entity)	Nullable (if not applicable).
PersonId	GUID?	ID of the ReCreateX person of who you want to call up all subscriptions.	Nullable (if not applicable).
PersonCardId	GUID?	ID of a ReCreateX PersonCard to which the subscriptions are linked.	Nullable (if not applicable).
CreatedFrom	DateTime?	If filled in: Only subscriptions that have been created after the given date will be shown.	
CreatedUntil	DateTime?	If filled in: Only subscriptions that have been created before the given date will be shown.	
Includes	SubscriptionIncludes	Determines which data are included in the response and/or which extra records will be present in the returnset.	
Paging	PagingCriteria	See 4.2: Paging Criteria	
ListForProlongation	bool	Do you want to list the subscriptions for prolongation only?	
ListForReader	bool	Do you want to list the subscriptions for readers?	
ReaderId	Guid?	ReaderId is used when checking the invalidity reasons	Nullable (if not applicable)
IgnoreExclusionCalendar	bool	IgnoreExclusionCalendar is used to ignore the validity of the calendar	
ArticleId	Guid?	The ID of the article for which you want to find subscriptions	Nullable (if not applicable)
ModifiedSince	DateTime?	Filter subscriptions that are last updated after the ModifiedSince datetime	Nullable (if not applicable)
ExpositionId	Guid?	Filters subscriptions that are required for given Exposition	
ExpositionItemId	Guid?	Filters subscriptions that are required for given ExpositionItem	
ModifiedUntil	DateTime?	Filter subscriptions that are last updated before ModifiedUntil	Nullable (if not applicable)

SubscriptionIncludes

The **SubscriptionIncludes** object can be built up using the following criteria:

Name	Type	Description	Remarks

Inactive	Bool	Include inactive subscriptions in the response.	
Invalid	Bool	Include invalid subscriptions in the response.	
Logs	Bool	Include the logging of this subscription.	
PersonCards	Bool	Include the linked customer cards.	
Image	Bool	Include an Image	
OnlyCurrentPersonCard	Bool	Include only the current person card	
InvalidityReasons	Bool	Include invalidity reasons	
LessonGroups	Bool	Include lesson groups	
CustomerPriceGroup	Bool	Include the customer price group	
AdmissionImage	Bool	Includes the image for Admission	

Response

The response will contain a list of **Subscription** objects that meet the criteria.

Subscription

The Subscription object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the subscription.	
CreationDate	DateTime	Creation date of the subscription.	
Article	Article	The linked SubscriptionArticle object.	See 20.2.2.1 for details concerning the SubscriptionArticle object.
Active	Bool	Is the subscription active?	
Valid	Bool	Is the subscription currently valid?	
Person	Person	The Person that owns the subscription.	
StartDateTime	DateTime?	Start date of the period during which a visit can take place.	
EndDateTime	DateTime?	End date of this period.	
NumberOfDays	Int?	Number of days in this period.	
RemainingDays	Int?	Remaining days.	
NumberOfTurns	Int?	Number of sessions on this subscription.	
RemainingTurns	Int?	Remaining sessions.	
LastProlongationDate	DateTime?	Date of last renewal.	
ProlongationAllowed	Bool	Renewal allowed.	
Logs	List<SubscriptionLog>	List of loggings.	
PersonCards	List<PersonCard>	List of linked customer cards.	
CheckPeriod	Bool	Validation of the period is active?	
CheckNumberOfTurns	Bool	Validation of the number of turns is active?	
ExtraInfo1	String	Extra information 1 of the subscription	
ExtraInfo2	String	Extra information 2 of the subscription	
ExtraInfo3	String	Extra information 3 of the subscription	
InvalidityReasons	IList<SubscriptionInvalidityReasons>		0 = InactiveSubscription 1 = InactiveSetting 2 = InvalidTurns 3 = InvalidTurnsPerPeriod 4 = InvalidPeriod 5 = InvalidHolidayCalendar 6 = InvalidSubscriptionCalendar 7 = InvalidActivation

			8 = InvalidSubscriptionMinutes 9 = InvalidLinkedExposition 10 = HallForSubscriptionIsInvalid 11 = ReservationForSubscriptionIsInvalid 12 = TimeIsInvalid 13 = InvalidAccessGroupCalendar 14 = InvalidAntiPassBack 15 = InvalidChipKnip 16 = InvalidCredits 17 = InvalidMaxTime 18 = InvalidMinTime 19 = InvalidCollectionContract 20 = DigitalWalletHasOutstandingCredit 21 = DigitalWalletHasInsufficientBalance 22 = DigitalWalletNotFoundForOverdueMinutes 23 = LinkedExpositionUnavailable ...
UnitPrice	decimal		
UnitPriceInc	decimal		
LessonGroupStudentSubscriptions	IList<LessonGroupStudentSubscription>	List of LessonGroupStudentSubscription	
IsCancelled	bool	Indicates if the subscription is cancelled	When it is cancelled, it's still possible that the subscription is active and can be used
ConversionAllowed	bool	Indiciates if the subscription can be converted into another	
IsFamilyMembership	bool	Indiciates if the subscription is a family membership	
PriceGroup	CustomerPriceGroup	The pricegroup of the customer	
LastEntryDate	DateTime?	Last time the customer received entry with this subscription	Null when no entries are registered
LastUpdate	DateTime?	Last time the subscription was updated	
EffectiveRemainingTurns	int?	Indicates the remaining turns taking into account the upcoming reserved usages	
OrderNumber	String	The order number of the subscription	
TurnsPerVisit	Int?	The number of turns used per visit	
AdmissionImage	Picture	The image for Admission	
AdmissionManualValidation	Bool	Indicates whether it should be manually validated in Admission	
HigherPriceAtSale	Bool	Indicates whether the collection contract price can be set higher	

LessonGroupStudentSubscription

Name	Type	Description	Remarks
Id	GUID	ID of the LessonGroupStudentSubscription.	
LessonGroupStudent	LessonGroupStudent	Information about the student	
SubscriptionId	Guid	The Guid of the subscription	

LessonGroupStudent

Name	Type	Description	Remarks
Id	GUID	ID of the LessonGroupStudent.	
LessonGroupId	GUID	ID of the lesson group	
LessonGroupShortName	String	Code of the lesson group	
LessonGroupDescription	String	Description of the lesson group	
LessonGroupComment	String	Comment of the lesson group	

LessonGroupFullName	String	Full name (code & description) of the lesson group	
EntryDate	DateTime	Entry date of the lesson group student	
AddressId	GUID	ID of the address	
Comment	string	A comment	
LastSync	DateTime	Last date time when this object as been synced	
LastUpdate	DateTime	Last date time when this object has been updated	

GetDirectDebitParameters

Calling up this Webservice operation will visualise the direct debit parameters linked to the division.

Request

The GetCurrentDivision consists of the service context object and a DirectDebitParametersCriteria object

The DirectDebitParametersCriteria looks as follows:

Name	Type	Description	Remarks
DivisionId	Guid	Unique identifier of the division.	

Response

The response is a DirectDebitParameters object.

The **DirectDebitParameters** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	Unique identifier of the settings.	
BacsServiceUserNumber	String	The user number for Bacs	

GetDirectDebitSummary

This method is used to get the contract summary when buying a direct debit subscription.

Request

The request consists of a service context and a DirectDebitSummaryCriteria object.
The properties of a DirectDebitSummaryCriteria look as follows:

DirectDebitSummaryCriteria

Name	Type	Description	Remarks
PersonId	GUID	The ID of the person (buyer). This is required to find the correct price.	
ArticleId	GUID	The ID of the article.	The article must be a SubscriptionArticle with property DirectDebitProcessing active
Quantity	Int	The number of articles.	
ContractStartDate	DateTime?	The start date of the direct debit contract	Mandatory when the SubscriptionArticle has property DirectDebitAskStartDate active

Response

The response consists of a DirectDebitSummaryResult object.
The properties of a DirectDebitSummaryResult look as follows:

DirectDebitSummaryResult

Name	Type	Description	Remarks
Valid	Bool		

		Defines whether the direct debit contract summary could be calculated	
ContractStartDate	DateTime	The start date of the direct debit contract	
FirstPaymentDate	DateTime	The first payment date of the direct debit contract	
CollectionFrequency	DirectDebitCollectionFrequency	The payment frequency of the direct debit contract	
CollectionAmount	Decimal	The payment amount of the direct debit contract	
ProRataAmount	Decimal	The pro rate amount of the direct debit contract	This is the amount that needs to be paid at basket checkout
ProRataFactor	Decimal	The pro rata factor of the direct debit contract	

DirectDebitCollectionFrequency

Name	Type	Description	Remarks
Id	Guid	The ID of the direct debit contract collection frequency	
Code	String	The code of the direct debit contract collection frequency	
Name	String	The name of the direct debit contract collection frequency	
NumberOfMonths	Int	The number of months for the direct debit contract collection frequency	

GetReaders

This method calls up all readers on the basis of certain search criteria.

Request

The request consists of the service context and a **ReaderSearchCriteria** object.

ReaderSearchCriteria

Name	Type	Description	Remarks
No parameters available (yet).			

Response

The response will contain a list of **Reader** objects that meet the criteria.

Reader

The Reader object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the reader.	
ShortName	String	Code of the reader.	
Description	String	Description of the reader.	
Number	byte?		
Address	String		
AccessWithoutSubscription	bool		

InterSolveDoPurchase

This method checks if the given card is valid, if the card is not valid the method tries to activate the card. The method also does a purchase.

Request

The request consists of the service context and a **InterSolveDoPurchaseRequest** with an IntersolveCriteria object.

Response

The response will contain a **IntersolveResult** object that defines the result of the method.

InterSolveIsCardValid

This method checks if the given card is valid, if the card is not valid the method tries to activate the card

Request

The request consists of the service context and a **InterSolveIsCardValidRequest** with an IntersolveCriteria object.

IntersolveCriteria

Name	Type	Description	Remarks
CardString	String	Text that represents the cardnumber	

Response

The response will contain a **IntersolveResult** object that defines the result of the method.

ValidationResult

Name	Type	Description	Remarks
Success	Bool	True if the function succeed, false if not	
ErrorMessage	String	Text that represends an error message	

Possible ErrorMessages:

- InterSolveServiceActivateFailed
- InterSolveServiceCardStatusBlocked
- InterSolveServiceCardStatusExpired
- InterSolveServiceCardStatusNoCardStatusFound
- InterSolveServiceCardStatusUnknown
- InterSolveServiceGetCardDetailsFailed
- InterSolveServiceNotAvailable
- InterSolveServicePurchaseFailed
- MuseumcardNotCorrectConfigured

ModifySubscriptionDates

Modify the StartDate and/or EndDate of an existing subscription

Request

The request consists of the service context and a **ModifySubscriptionDatesCriteria** object.

ModifySubscriptionDates

Name	Type	Description	Remarks
SubscriptionID	GUID?	The ID of an existing Subscription	
StartDate	DateTime?	The DateTime you wish to set as StartDate of the Subscription	
EndDate	DateTime?	The DateTime you wish to set as EndDate of the Subscription	

Sample

```
{
  "Criteria": {
    "SubscriptionId": "65692D67-81DE-4BA2-B857-90E818CCBE24",
    "StartDate": "2020-11-07 00:00:00.000",
    "EndDate": "2021-11-06 00:00:00.000"
  },
  "Context": {
    "ShopId": "00000000-0000-0000-0000-000000000730",
    "Password": "string_example"
  }
}
```

Response

The response will contain a **ModifySubscriptionDatesResult** object.

ModifySubscriptionDatesResult

The **ModifySubscriptionDatesResult** object looks as follows:

Name	Type	Description	Remarks
Success	Bool	Indicates whether the call succesfully modified the StartDate and/or EndDate of the Subscription	
ErrorMessage	String	When success is false, an error message will be specified for more information	

Sample

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
ModifySubscriptionDatesResponse, ReCreateX.WebShop.WebServices.
Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
ModifySubscriptionDatesResult, ReCreateX.WebShop.Model",
        "success": true,
        "errorMessage": null
    }
}
```

Error Messages

The following values are possible:

- Invalid SubscriptionId

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
ModifySubscriptionDatesResponse, ReCreateX.WebShop.WebServices.
Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
ModifySubscriptionDatesResult, ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "The entered SubscriptionId is invalid"
    }
}
```

- Invalid Subscription

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
ModifySubscriptionDatesResponse, ReCreateX.WebShop.WebServices.
Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
ModifySubscriptionDatesResult, ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "The subscription could not be found or is
invalid"
    }
}
```

- EndDate < StartDate

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
ModifySubscriptionDatesResponse, ReCreateX.WebShop.WebServices.
Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
ModifySubscriptionDatesResult, ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "EndDate cannot be earlier than StartDate"
    }
}
```

- Exception

```
{
    "$type": "ReCreateX.WebShop.WebServices.Contracts.
ModifySubscriptionDatesResponse, ReCreateX.WebShop.WebServices.
Contracts",
    "result": {
        "$type": "ReCreateX.WebShop.Model.
ModifySubscriptionDatesResult, ReCreateX.WebShop.Model",
        "success": false,
        "errorMessage": "System.NullReferenceException: Object
reference not set to an instance of an object."
    }
}
```

RegisterSubscriptionRefusationLog

This method registers a refusation log for a certain person card and reader.

Request

The request consists of the service context and a **SubscriptionRefusationLogCriteria** object.

SubscriptionRefusationLogCriteria

Name	Type	Description	Remarks
PersonCardId	Guid	Unique identifier of the person card.	
ReaderId	Guid?	Unique identifier of the reader.	Marked as optionally, but no logging will occur when no reader is passed!
SubscriptionId	Guid?	Unique identifier of the subscription	Marked as optionally, only register a log on this specific subscription
EmployeeId	Guid?	Unique identifier of the employee	Marked as optionally

Response

The response will contain a **ValidationResult** object that defines the result of the transaction.

ValidationResult

Name	Type	Description	Remarks
BrokenRuleName	String	The error code.	
Message	String	The standard error.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	

RemovePersonCardSubscription

This method removes the given subscription from the given person card.

Request

The request consists of the service context and a **PersonCardSubscriptionCriteria** object.

PersonCardSubscriptionCriteria

Name	Type	Description	Remarks
PersonCardId	GUID	ID of the person card.	
SubscriptionId	GUID	ID of the subscription.	

Response

The response will contain a **ValidationResult** object that defines the result of the transaction.

ValidationResult

Name	Type	Description	Remarks
BrokenRuleName	String	The error code.	
Message	String	The standard error.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	

Subscriptions – Brief overview

The membership (subscriptions) module is available as of ReCreateX 5.1.0.
The system functions as follows:

- A membership can be created using the Webshop on the basis of an ArticleSale.
- A membership can be renewed on the basis of a SubscriptionProlongation basket item.

The Webservice methods that have been created for this purpose are the following:

- FindSubscriptionArticles (find Articles of the membership type)
- FindSubscriptions
- FindSubscriptionLogs

21 Files module (as of RCX 5.2.0)

Files – Brief overview

The files module is available as of ReCreateX 5.2.0.
The system functions as follows:

- A file can be created using the api services.

The Webservice methods that have been created for this purpose are the following:

- FindFileActivities
- ListFileActivityCategories
- ListFileAgeGroups
- FindFileTypes
- ListFileActivityPeriods
- FindFileActivityArticles
- FindFiles

FindFileActivities

This method is used to find the file activities of the type CounterSale.
It will return all the file activities which may be visualised on the website.

Request

The request consists of a service context and a FileActivitySearchCriteria object.
The properties of a FileActivitySearchCriteria look as follows:

FileActivitySearchCriteria

Name	Type	Description	Remarks
FileActivityId	GUID?	ID of the file activity.	
NamePattern	String	A search pattern on name.	
FileActivityCategoryId	GUID?	The ID of the file activity category.	
FileAgeGroupId	GUID?	The ID of the file age group.	
FileTypeId	GUID?	The ID of the file type.	
TranslationRequired	Boolean	If true, only the file activities that have a translation will be included.	
Paging	PagingCriteria	Paging criteria.	
From	DateTime?	If filled in: Only the file activities that have been registered after the given date will be visualised.	
Until	DateTime?	If filled in: Only the file activities that have been registered before the given date will be visualised.	

Response

The response will contain a list of FileActivity objects that meet the criteria.
The **FileActivity** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the file activity.	
Code	String	The shortname of the file activity.	
Name	String	The name of the file activity.	
Description	String	The description of the file activity.	
FileActivityCategory	FileActivityCategory	The FileActivityCategory to which the file activity belongs.	
SalesLocationID	Guid	The SalesLocation of the file activity.	
Duration	DateTime	The duration of the file activity.	
MaxPeoplePerGroup	Int	The max. number of people that can take part in this file activity.	

MaxPeoplePerService	Int	The max. number of people that use this service.	

FindFileActivityArticles

This method is used to visualise the articles that can be ordered within a file.

Request

The request consists of a service context and a FileActivityArticleSearchCriteria object.
The properties of a **FileActivityArticleSearchCriteria** look as follows:

FileActivityArticleSearchCriteria

Name	Type	Description	Remarks
ArticleId	Guid?	The unique identifier of the article.	
NamePattern	String?	A search pattern on name.	
FileActivityID	Guid?	If filled in, only the file articles that may be ordered for this file activity will be shown.	
Paging	PagingCriteria	Paging Criteria.	
Includes	FileActivityArticleIncludes	Which data has to be included.	
BuyerID	Guid?	If filled in: The pricegroup of the buyer is used to get the price of the article.	

FileActivityArticleIncludes

Name	Type	Description	Remarks
ImageUrl	Boolean	Include the url of the image of the article.	
Image	Boolean	Include the image of the article.	
ArticleGroup	Boolean	Include the articlegroup of the article.	
ArticleCategories	Boolean	Include the articlecategory of the article.	

Response

The response will contain a list of Article objects that meet the criteria.
The **Article** object is already described in section 5.1.2.

FindFiles

This method is used to visualise files.

Request

The request consists of a service context and a FileSearchCriteria object.
The properties of a **FileSearchCriteria** look as follows:

FileSearchCriteria

Name	Type	Description	Remarks
FileId	Guid?	The unique identifier of the file.	
From	DateTime?	Start time of the file.	
Until	DateTime?	End time of the file.	
PersonId	Guid?	The unique identifier of the person that ordered the file.	
Paging	PagingCriteria	Paging Criteria.	
Includes	FileIncludes	Which data has to be included.	
Number	string	The number of the file.	

FileIncludes

Name	Type	Description	Remarks
SubFiles	Boolean	Include the subfiles of the file. It is obligatory if you want to include the fileitems of the subfile of the file.	

FileItems	Boolean	Include the file items of the subfile of the file.	
PersonDetails	Boolean	Include the person details of the person that ordered the file.	

Response

The response will contain a list of File objects that meet the criteria.
The **File** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	ID of the file.	
Number	String	Number of the file.	
DivisionId	Guid	DivisionId of the file.	
Status	FileStatus	Status of the file.	
FileTypeId	Guid	ID of the filetype of the file.	
FileType	FileType	FileType of the file.	
PersonId	Guid	ID of the person that ordered the file.	
Person	Person	Person that ordered the file.	
InvoicePersonId	Guid	ID of the invoice person of the file.	
Description	String	Description of the file.	
Comment	String	Comment of the file.	
CreationDate	DateTime	Creationdate of the file.	
OrderDate	DateTime	Orderdate of the file.	
ExecutionDate	DateTime	Executiondate of the file.	
ExecutionTime	TimeSpan	Executiontime of the file.	
NumberOfVisitors	Int	Number of visitors of the file.	
Invoiced	Boolean	Indicates whether the file is invoiced.	
FileAgeGroupId	Guid	ID of the file age group of the file.	
FileAgeGroup	FileAgeGroup	FileAgeGroup of the file.	
WebCreated	Boolean	Indicated whether the file was created via the web.	
SubFiles	List<SubFile >	List of subfiles that belong to the file.	

The **FileStatus** object is an enumeration with the following possible values:
Offer = 0,
Check = 1,
Approved = 2,
Disapproved = 3,
Ordered = 4,
Cancelled = 5
The **SubFile** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	ID of the subfile.	
FileId	Guid	ID of the file to which the subfile belongs.	
Sequence	Int	Sequence number of the subfile.	
FileActivityId	Guid	ID of the file activity of the subfile.	
FileActivity	FileActivity	FileActivity of the subfile.	
Description	String	Description of the subfile.	
InvoiceAddressId	Guid	ID of the invoice person of the subfile.	

Date	DateTime	Startdate of the subfile.	
StartTime	TimeSpan	Starttime of the subfile.	
EndDate	DateTime	Enddate of the subfile.	
EndTime	TimeSpan	Endtime of the subfile.	
NumberOfPeople	Int	Number of people of the subfile.	
NumberOfPlaces	Int	Number of places of the subfile.	
NumberOfGroups	Int	Number of groups of the subfile.	
Comment	String	Comment of the subfile.	
Invoiced	Boolean	Indicates whether the subfile is invoiced.	
FileAgeGroupId	Guid	ID of the file age group of the subfile.	
FileItems	List<FileItem >	List of file items that belong to the subfile.	
SubFileWorkTypes	List<SubFileWorkType>	List of subfile worktypes that belong to the subfile.	
Location	AddressLocation	The location of the subfile.	
LocationAddressID	Guid	The ID of the location of the subfile.	
CustomerName	string	The name of the invoice customer	
CustomerFirstname	string	the firstname of the invoice customer	

The **FileItem** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	ID of the file item.	
SubFileId	Guid	ID of the subfile to which the file item belongs.	
ArticleId	Guid	ID of the article of the file item.	
Article	Article	Article of the file item.	
CustomerPriceGroupId	Guid	ID of the customer pricegroup of the file item.	
CustomerPriceGroup	CustomerPriceGroup	CustomerPriceGroup of the file item.	
Quantity	Int	The quantity that is ordered of the file item.	
Hours	DateTime	The period of time that the file item is ordered.	
CalculationType	FileItemCalculationType	The calculationtype of the file item.	
UnitPrice	Decimal	The unitprice of the file item.	
UnitPriceInc	Decimal	The unitprice inclusive vat of the file item.	
VatPercentage	Decimal	The vatpercentage of the file item.	
VatId	Guid	ID of the vat of the file item.	
VatInclusive	Boolean	Indicates if the price is saved incl. or excl. VAT	
VatAmount	Decimal	VAT amount that was paid for the file item.	
LineAmount	Decimal	Net amount that was paid for the file item.	
Amount	Decimal	The total amount of the file item.LineAmount + VatAmount	
StockItem	Boolean	Indicates whether the file item is a stock item.	
ItemStockPlaceId	Guid	ID of the stockplace of the file item.	
Comment	String	Comment of the file item.	
DiscountPercentage	Decimal	Discount percentage of the file item.	
DiscountAmount	Decimal	Discount amount of the file item.	

The **FileItemCalculationType** object is an enumeration with the following possible values:
PxH = 0,
F = 1,

PxQ = 2,
PxQxH =3
The **SubFileWorkType** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	ID of the subfile worktype.	
SubFileId	Guid	ID of the subfile to which the subfile worktype belongs.	
WorkTypeItemId	Guid	ID of the article of the subfile worktype.	
WorkTypeItem	Article	WorkTypeItem of the subfile worktype.	
CustomerPriceGroupId	Guid	ID of the customer pricegroup of the subfile worktype.	
CustomerPriceGroup	CustomerPriceGroup	CustomerPriceGroup of the subfile worktype.	
Hours	DateTime	The period of time that the subfile worktype is used.	
CalculationType	SubFileWorkTypeCalculationType	The calculationtype of the subfile worktype.	
UnitPrice	Decimal	The unitprice of the subfile worktype.	
UnitPriceInc	Decimal	The unitprice inclusive vat of the subfile worktype.	
VatPercentage	Decimal	The vatpercentage of the subfile worktype.	
VatId	Guid	ID of the vat of the subfile worktype.	
VatInclusive	Boolean	Indicates if the price is saved incl. or excl. VAT	
VatAmount	Decimal	VAT amount that was paid for the subfile worktype.	
LineAmount	Decimal	Net amount that was paid for the subfile worktype.	
Amount	Decimal	The total amount of the subfile worktype.LineAmount + VatAmount	
Comment	String	Comment of the subfile worktype.	
LanguageId	Guid	LanguageId of the subfile worktype.	
PerGroup	Boolean	Indicates if the subfile worktype is made Per group.	
EmployeeTaskId	Guid	EmployeeTaskId of the subfile worktype.	
EmployeeId	Guid	EmployeeId of the subfile worktype.	
FileActivityWorkTypeId	Guid	FileActivityWorkTypeId of the subfile worktype.	
DiscountPercentage	Decimal	Discount percentage of the subfile worktype.	
DiscountAmount	Decimal	Discount amount of the subfile worktype.	
PayToEmployee	Boolean	Indicates if the amount has to be payed to the employee of the subfile worktype.	
Posted	Boolean	Indicates whether the subfile worktype is posted.	
FixedDuration	Boolean	Indicates if the duration of the subfile worktype is fixed.	
FromDateTime	DateTime	Start of the subfile worktype.	
TillDateTime	DateTime	End of the subfile worktype.	

The **SubFileWorkTypeCalculationType** object is an enumeration with the following possible values:
PxH = 0,
F = 1,
PxQ = 2,
PxQxH =3

FindFiletypes

This method is used to visualise the FileTypes.

Request

The request consists of a service context and a FileTypeSearchCriteria object.
The properties of a **FileTypeSearchCriteria** look as follows:

FileTypeSearchCriteria

Name	Type	Description	Remarks
FileTypeId	GUID?	ID of the file type.	
NamePattern	String	A search pattern on name.	
FileAgeGroupId	GUID?	The ID of the file age group.	
Paging	PagingCriteria	Paging criteria.	
Includes	FileTypeIncludes	Which data has to be included.	

FileTypeIncludes

Name	Type	Description	Remarks
FileTypeParts	Boolean	Include the filetypeparts of the filetype.	

Response

The response will contain a list of FileType objects that meet the criteria.
The **FileType** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the file type.	
Code	String	The shortname of the file type.	
Name	String	The name of the file type.	
Description	String	The description of the file type.	

The **FileTypePart** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	ID of the filetypepart.	
FileTypeId	Guid	ID of the filetype to which the filetypepart belongs.	
FileType	FileType	Filetype of the filetypepart.	
Sequence	Int	Sequence number of the filetypepart.	
Description	String	Description of the filetypepart.	
Fixed	Boolean	Shows if the filetypepart is fixed.	
FileActivityId	Guid	ID of the file activity of the filetypepart.	
FileActivity	FileActivity	FileActivity of the filetypepart.	

ListFileActivityCategories

Calling up this Webservice will visualise the FileActivityCategories.

Request

The request only contains a service context.

Response

The response will contain a list of FileActivityCategory objects.
The **FileActivityCategory** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the file activity category.	
Code	String	The shortname of the file activity category.	
Name	String	The name of the file activity category.	
Description	String	The description of the file activity category.	

ListFileActivityPeriods

Each FileActivity has a number of FileActivityPeriods. This method is used to visualise the FileActivityPeriods.

Request

The request consists of a service context and a FileActivityPeriodSearchCriteria object.
The properties of a **FileActivityPeriodSearchCriteria** look as follows:

FileActivityPeriodSearchCriteria

Name	Type	Description	Remarks
FileActivityId	GUID	The unique identifier of the file activity of which the periods are called up.	Obligatory.
From	DateTime	If filled in, only the file activity periods that start after this date will be shown.	
Until	DateTime	If filled in, only the file activity periods that end before this date will be shown.	
Paging	PagingCriteria	Paging Criteria.	

Response

The response will contain a list of FileActivityPeriod objects that meet the criteria.
The **FileActivityPeriod** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the file activity period.	
FileActivityID	GUID	ID of the file activity.	
From	DateTime	Start time of the FileActivityPeriod.	
Until	DateTime	End time of the FileActivityPeriod.	
Maximum	Int	Max. number of persons for this period.	
CapacityCheck	CapacityCheck	The CapacityCheck that is used to determine the availability.	
CapacityType	CapacityType	The CapacityType thas is used to determine the availability.	
Available	Int	The number of places/groups that are still available.	

The **CapacityCheck** object is an enumeration with the following possible values:
None = 0,
Places = 1,
Groups = 2,
Overlay = 3
The **CapacityType** object is an enumeration with the following possible values:
Simultaneous = 0,
ByPeriod = 1

ListFileAgegroups

Calling up this Webservice will visualise the FileAgeGroups.

Request

The request only contains a service context.

Response

The response will contain a list of FileAgeGroup objects.
The **FileAgeGroup** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the file age group.	
Code	String	The shortname of the file age group.	
Name	String	The name of the file age group.	
Description	String	The description of the file age group.	

FindSubFiles

Request to return a SubFile based on it’s ID.

Request

The request consists of a service context and a SubFileSearchCriteria object.
The properties of a Sub**FileSearchCriteria** look as follows:

SubFileSearchCriteria

FileId	Guid?	The unique identifier of the Subfile.	

Response

The response will contain a list of SubFile objects that meet the criteria.
The **SubFile** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	ID of the subfile.	
FileId	Guid	ID of the file to which the subfile belongs.	
Sequence	Int	Sequence number of the subfile.	
FileActivityId	Guid	ID of the file activity of the subfile.	
FileActivity	FileActivity	FileActivity of the subfile.	
Description	String	Description of the subfile.	
InvoiceAddressId	Guid	ID of the invoice person of the subfile.	
Date	DateTime	Startdate of the subfile.	
StartTime	TimeSpan	Starttime of the subfile.	
EndDate	DateTime	Enddate of the subfile.	
EndTime	TimeSpan	Endtime of the subfile.	
NumberOfPeople	Int	Number of people of the subfile.	
NumberOfPlaces	Int	Number of places of the subfile.	
NumberOfGroups	Int	Number of groups of the subfile.	
Comment	String	Comment of the subfile.	
Invoiced	Boolean	Indicates whether the subfile is invoiced.	
FileAgeGroupId	Guid	ID of the file age group of the subfile.	
FileItems	List<FileItem >	List of file items that belong to the subfile.	
SubFileWorkTypes	List<SubFileWorkType>	List of subfile worktypes that belong to the subfile.	
Location	AddressLocation	The location of the subfile.	
LocationAddressID	Guid	The ID of the location of the subfile.	
CustomerName	string	The name of the invoice customer	
CustomerFirstname	string	the firstname of the invoice customer	

22 Document service

Activity Order Document

The activity order document bundles all registrations within one order document. Different fields can be used in this template that can be listed using a Help service:
http://\{server}/WebshopDocumentService/Help/ActivityOrder/{ShopId}/{Language}

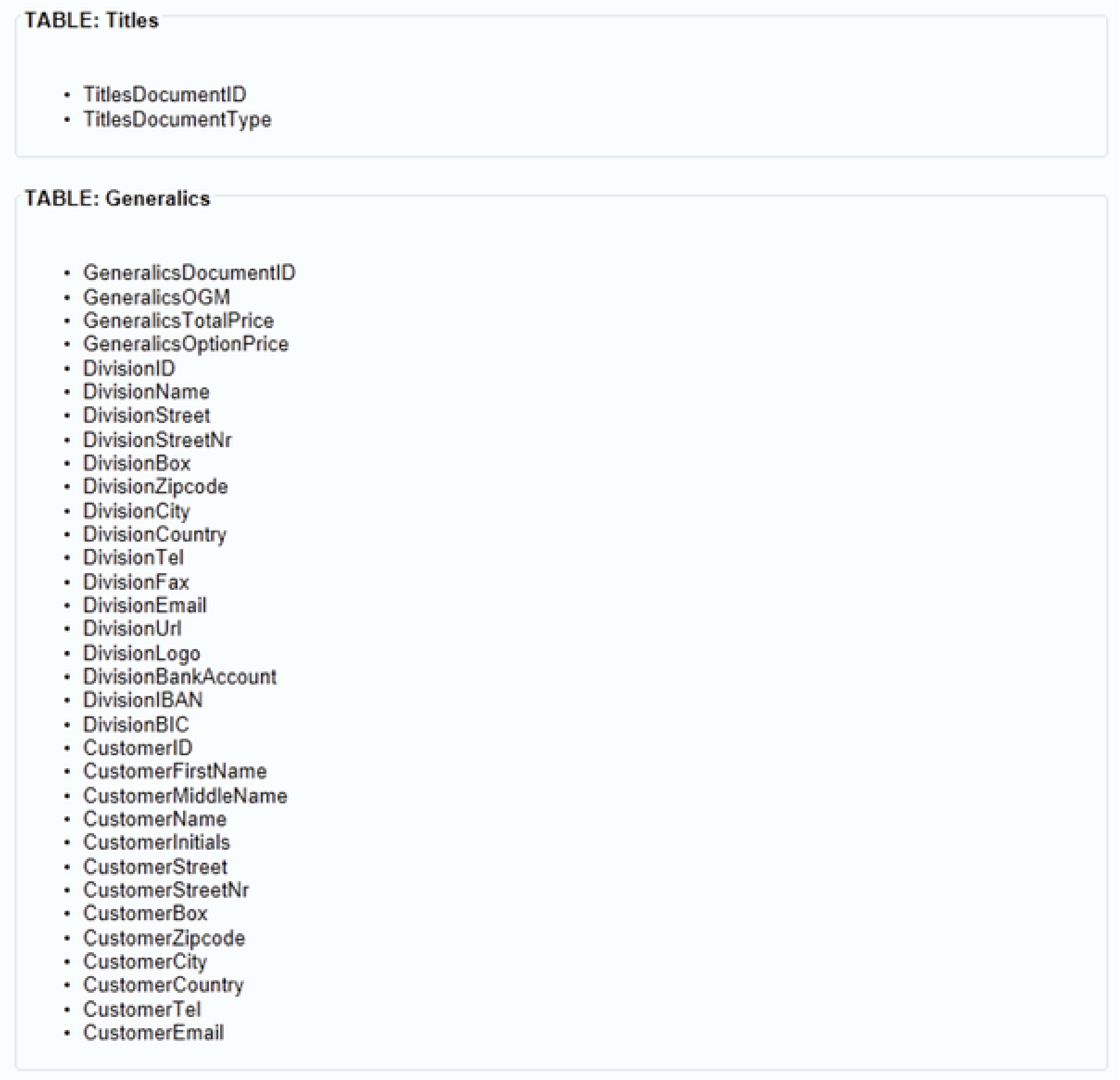


TABLE: Titles

- TitlesDocumentID
- TitlesDocumentType

TABLE: Generalics

- GeneralicsDocumentID
- GeneralicsOGM
- GeneralicsTotalPrice
- GeneralicsOptionPrice
- DivisionID
- DivisionName
- DivisionStreet
- DivisionStreetNr
- DivisionBox
- DivisionZipcode
- DivisionCity
- DivisionCountry
- DivisionTel
- DivisionFax
- DivisionEmail
- DivisionUrl
- DivisionLogo
- DivisionBankAccount
- DivisionIBAN
- DivisionBIC
- CustomerID
- CustomerFirstName
- CustomerMiddleName
- CustomerName
- CustomerInitials
- CustomerStreet
- CustomerStreetNr
- CustomerBox
- CustomerZipcode
- CustomerCity
- CustomerCountry
- CustomerTel
- CustomerEmail

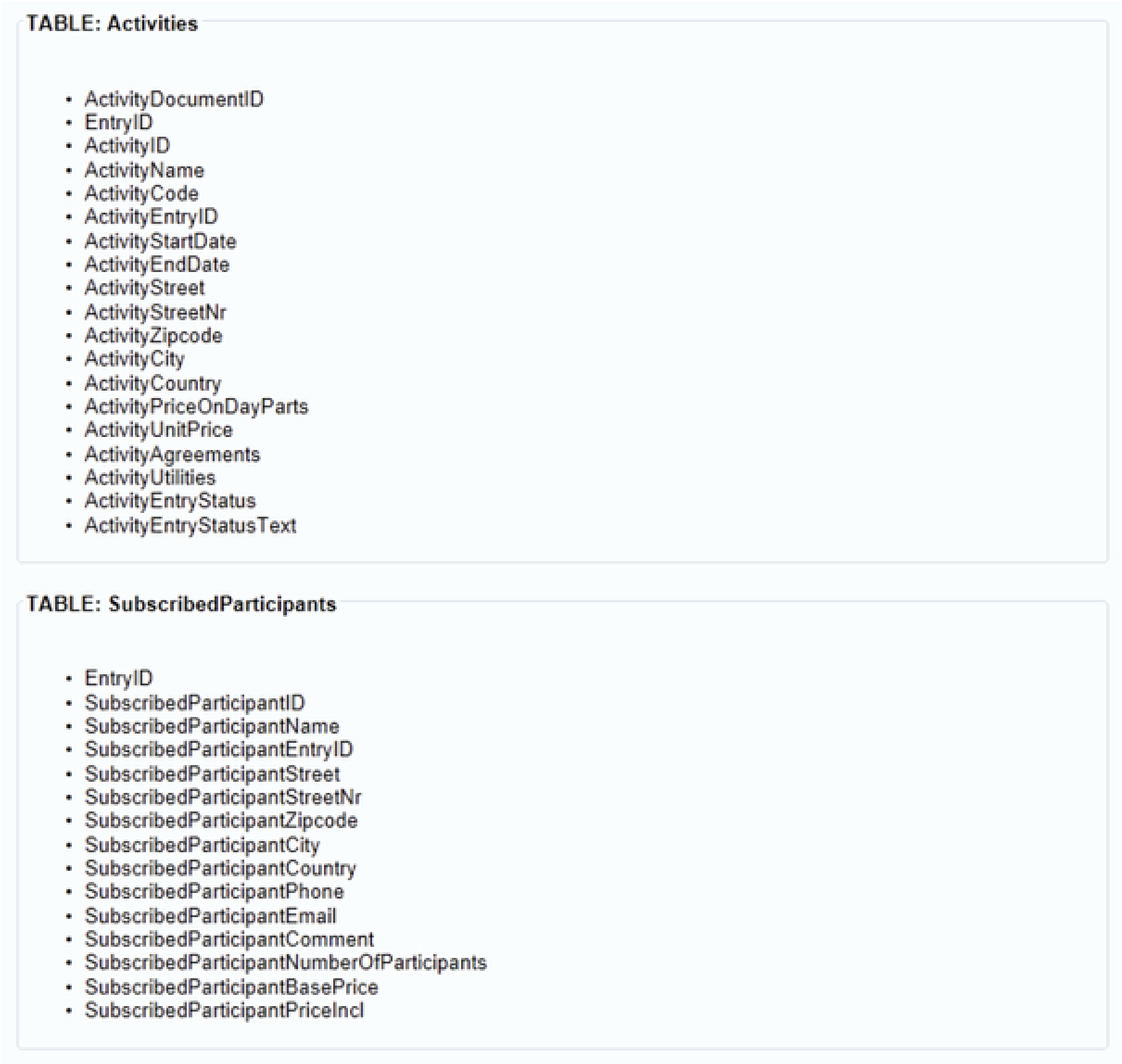


TABLE: Activities

- ActivityDocumentID
- EntryID
- ActivityID
- ActivityName
- ActivityCode
- ActivityEntryID
- ActivityStartDate
- ActivityEndDate
- ActivityStreet
- ActivityStreetNr
- ActivityZipcode
- ActivityCity
- ActivityCountry
- ActivityPriceOnDayParts
- ActivityUnitPrice
- ActivityAgreements
- ActivityUtilities
- ActivityEntryStatus
- ActivityEntryStatusText

TABLE: SubscribedParticipants

- EntryID
- SubscribedParticipantID
- SubscribedParticipantName
- SubscribedParticipantEntryID
- SubscribedParticipantStreet
- SubscribedParticipantStreetNr
- SubscribedParticipantZipcode
- SubscribedParticipantCity
- SubscribedParticipantCountry
- SubscribedParticipantPhone
- SubscribedParticipantEmail
- SubscribedParticipantComment
- SubscribedParticipantNumberOfParticipants
- SubscribedParticipantBasePrice
- SubscribedParticipantPriceIncl

Format of the URL:
http://\{server}/WebshopDocumentService/ActivityOrder/{ShopId}/{Language}/{OrderId}
The OrderID is available withint each ActivityReservationSalesItem object that is returned when using the CheckoutBasket() method.

Barcodes / QR codes

It is possible to add barcodes (code 128) and QR codes within the PDF documents.
To do this, a specific mergefield needs to be added in the WORD template.

«barcode:{symbology}:{field}(:{type})»

{symbology} = **128** or **qr**
{field} = name of the field which needs to be formatted as barcode
{type} = type of generation: **1** or **2** (optional)

Symbology

Barcode

For barcodes, following syntax should be used for the mergefield:

«barcode:128:{field}(:{type})»

{field} = name of the field which needs to be formatted as barcode

Ex:

- *barcode:128:cardnumber*
- *barcode:128:cardnumber:2*

QR code

For QR codes, following syntax should be used for the mergefield:

«barcode:qr:{field}(:{type})»

{field} = name of the field which needs to be formatted as QR code

Ex:

- *barcode:qr:cardnumber*
- *barcode:qr:cardnumber:2*

Type

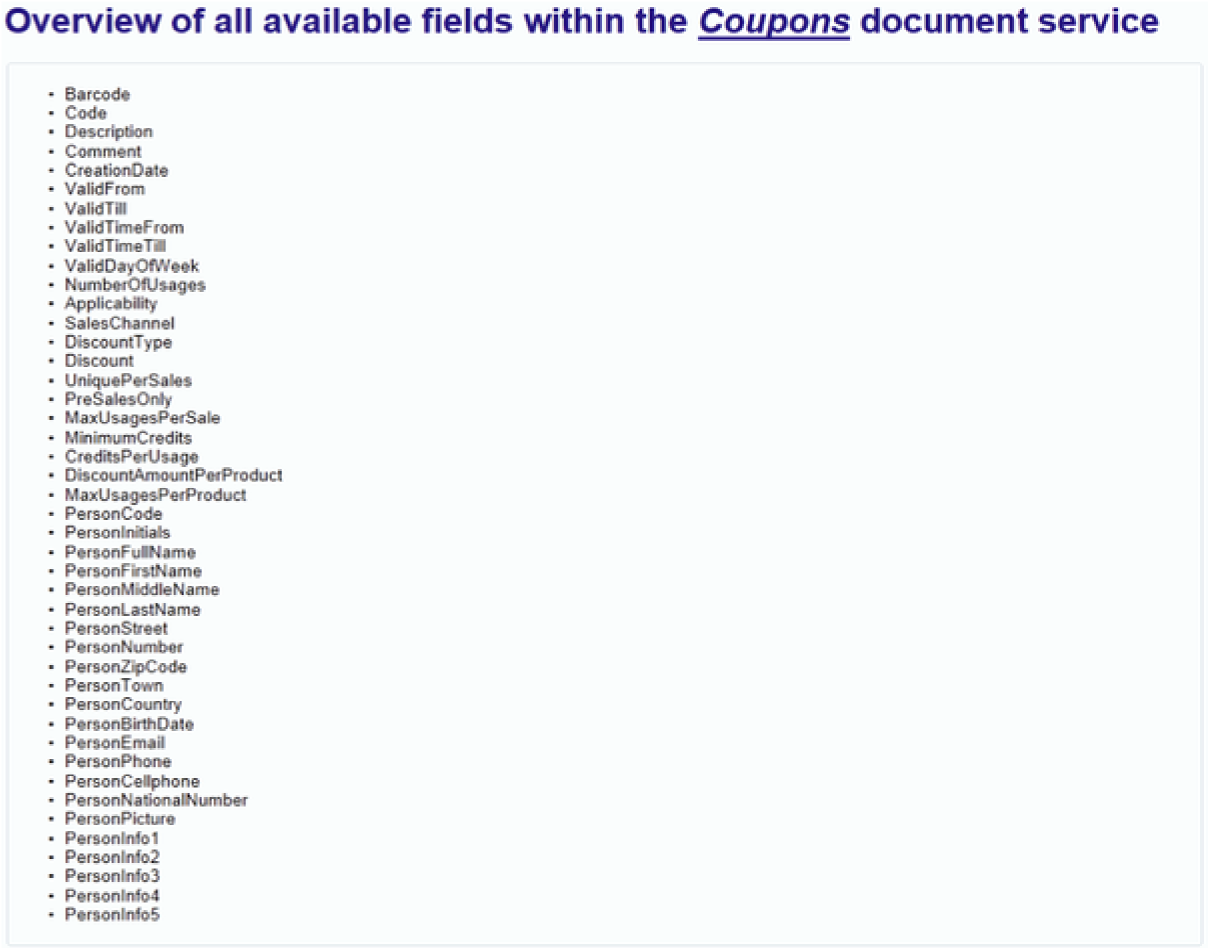
There are 2 ways for generating the barcodes.

1 (=default)	2
⊖ Limited to a fixed width of 325px	⊕ No size limits
⊖ Displays the value below the barcode	⊕ Doesn't display the value below the barcode

- Type 1 is used as default. When the generation of the barcode does not succeed, the system will automatically try type 2.
- The best way to manually change the size of the barcode is to include in a table with 1 row and 1 column. The barcode will then automatically size within that cell.

Coupon tickets

The coupon document service prints all coupons that were created within a specific sale. Different fields can be used in this template that can be listed using a Help service:



Overview of all available fields within the *Coupons* document service

- Barcode
- Code
- Description
- Comment
- CreationDate
- ValidFrom
- ValidTill
- ValidTimeFrom
- ValidTimeTill
- ValidDayOfWeek
- NumberOfUsages
- Applicability
- SalesChannel
- DiscountType
- Discount
- UniquePerSales
- PreSalesOnly
- MaxUsagesPerSale
- MinimumCredits
- CreditsPerUsage
- DiscountAmountPerProduct
- MaxUsagesPerProduct
- PersonCode
- PersonInitials
- PersonFullName
- PersonFirstName
- PersonMiddleName
- PersonLastName
- PersonStreet
- PersonNumber
- PersonZipCode
- PersonTown
- PersonCountry
- PersonBirthDate
- PersonEmail
- PersonPhone
- PersonCellphone
- PersonNationalNumber
- PersonPicture
- PersonInfo1
- PersonInfo2
- PersonInfo3
- PersonInfo4
- PersonInfo5

Format of the URL:
http://\{server}/WebshopDocumentService/Coupons/{ShopId}/{Language}/{SalesLineID}
The SalesLineID is the ID that is generated out of the CheckOutBasketResult.SalesItem [...] ID after CheckOutBasket.

CultureEvent Tickets

If you want to find the tickets that have been created during a reservation for a culture event, you can use the WebShopDocumentService.
After the creation of a reservation, the CheckOutBasket will have given you a SalesItem ID for all CultureEventReservations (this will be the same ID for all events). This ID allows you to call up the tickets of the reservation.
Suppose the SalesItem ID is '*27B769BB-1D51-4904-8BC8-037B55E4B0C4*', this ID will allow you to find the required tickets by surfing to the following address.

Format:
http://\{server}/WebshopDocumentService/CultureEvents/{ShopId}/{Language}/{ReservationId}
In our example, this will be:
http://localhost/WebshopDocumentService/CultureEvents/00000000-0000-0000-0000-000000000001/NL/27B769BB-1D51-4904-8BC8-037B55E4B0C4'
Moreover, it is also possible to call up a series of tickets for a customer and CultureEvent.
In this case, the following format is used:
http://\{server}/WebshopDocumentService/CultureEvents/{ShopId}/{Language}/{CustomerId}/{CultureEventID}/{From}/{Until}
CustomerID: the identifier of the customer
CultureEventID: the identifier of the event
From, Until: the period from-till in which the date format is yyyymmdd.

Direct Debit documents

Two direct debit documents can be used when a direct debit subscription was bought.

- *DirectDebitContractSetUp*
- *DirectDebitContractPayment*

The direct debit contract set-up document can be used to show all direct debit contract information.
The direct debit contract payment document can be used to show all direct debit contract information as well as the payment schedule.

Help pages can be found at following URLs:

- *DirectDebitContractSetUp: http://\{server}/WebshopDocumentService/Help/DirectDebitContractSetUp/{ShopId}/{Language}*
- *DirectDebitContractPayment: http://\{server}/WebshopDocumentService/Help/DirectDebitContractPayment/{ShopId}/{Language}*

Both documents use the same underlying dataset containing following fields:

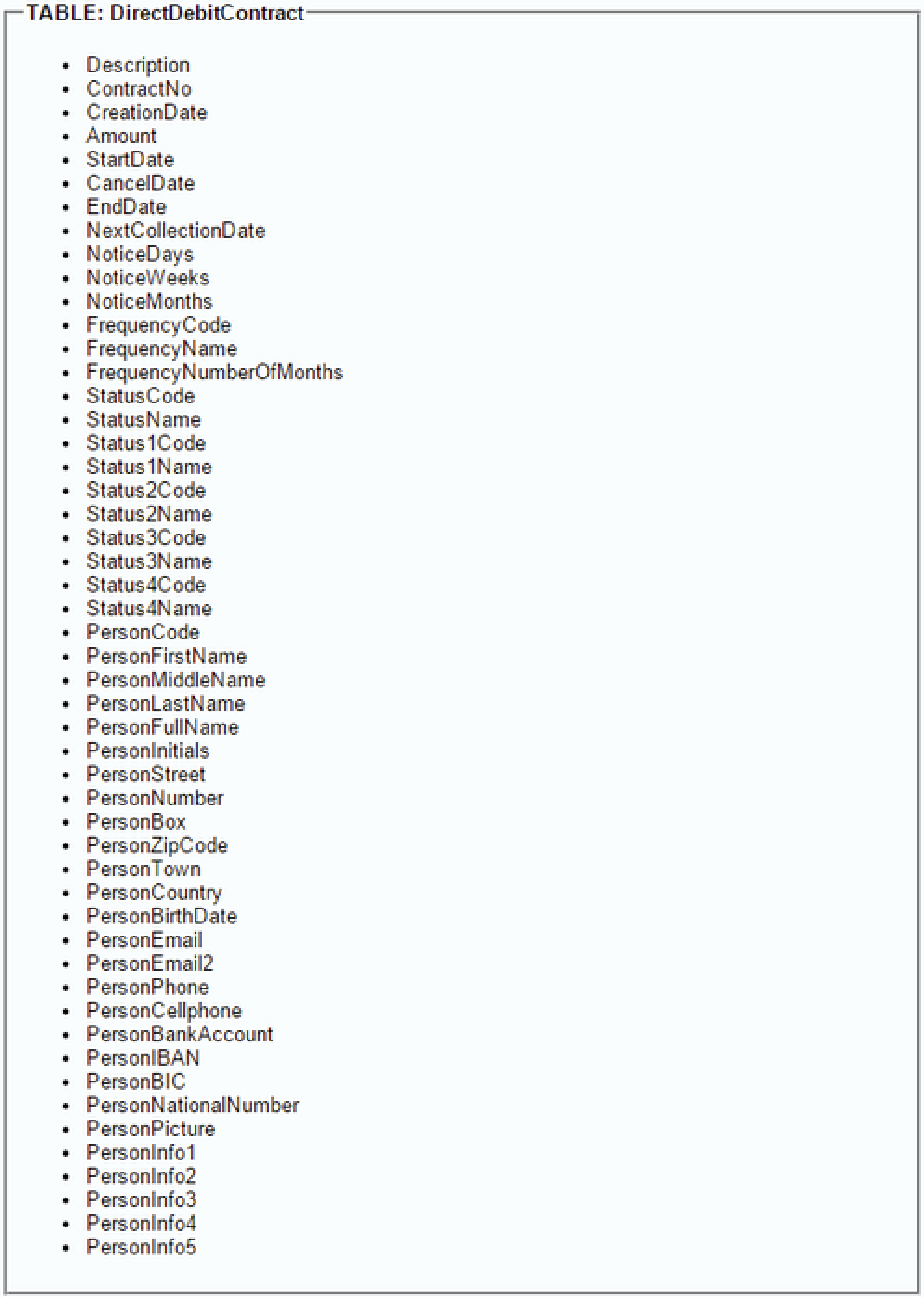


TABLE: DirectDebitContract

- Description
- ContractNo
- CreationDate
- Amount
- StartDate
- CancelDate
- EndDate
- NextCollectionDate
- NoticeDays
- NoticeWeeks
- NoticeMonths
- FrequencyCode
- FrequencyName
- FrequencyNumberOfMonths
- StatusCode
- StatusName
- Status1Code
- Status1Name
- Status2Code
- Status2Name
- Status3Code
- Status3Name
- Status4Code
- Status4Name
- PersonCode
- PersonFirstName
- PersonMiddleName
- PersonLastName
- PersonFullName
- PersonInitials
- PersonStreet
- PersonNumber
- PersonBox
- PersonZipCode
- PersonTown
- PersonCountry
- PersonBirthDate
- PersonEmail
- PersonEmail2
- PersonPhone
- PersonCellphone
- PersonBankAccount
- PersonIBAN
- PersonBIC
- PersonNationalNumber
- PersonPicture
- PersonInfo1
- PersonInfo2
- PersonInfo3
- PersonInfo4
- PersonInfo5

Format of the URLs:

- *DirectDebitContractSetUp: http://\{server}/WebshopDocumentService/DirectDebitContractSetUp/{ShopId}/{Language}/{SalesLineID}*
- *DirectDebitContractPayment: http://\{server}/WebshopDocumentService/DirectDebitContractPayment/{ShopId}/{Language}/{SalesLineID}*

The SalesLineID is the ID that is generated out of the CheckOutBasketResult.SalesItem [...] ID after CheckOutBasket.

Document service – Brief overview

Using the Webservices it is possible to show a document on a Webpage using a link. For this purpose you can use the following Webservice (WebShopDocumentService.svc) operations.

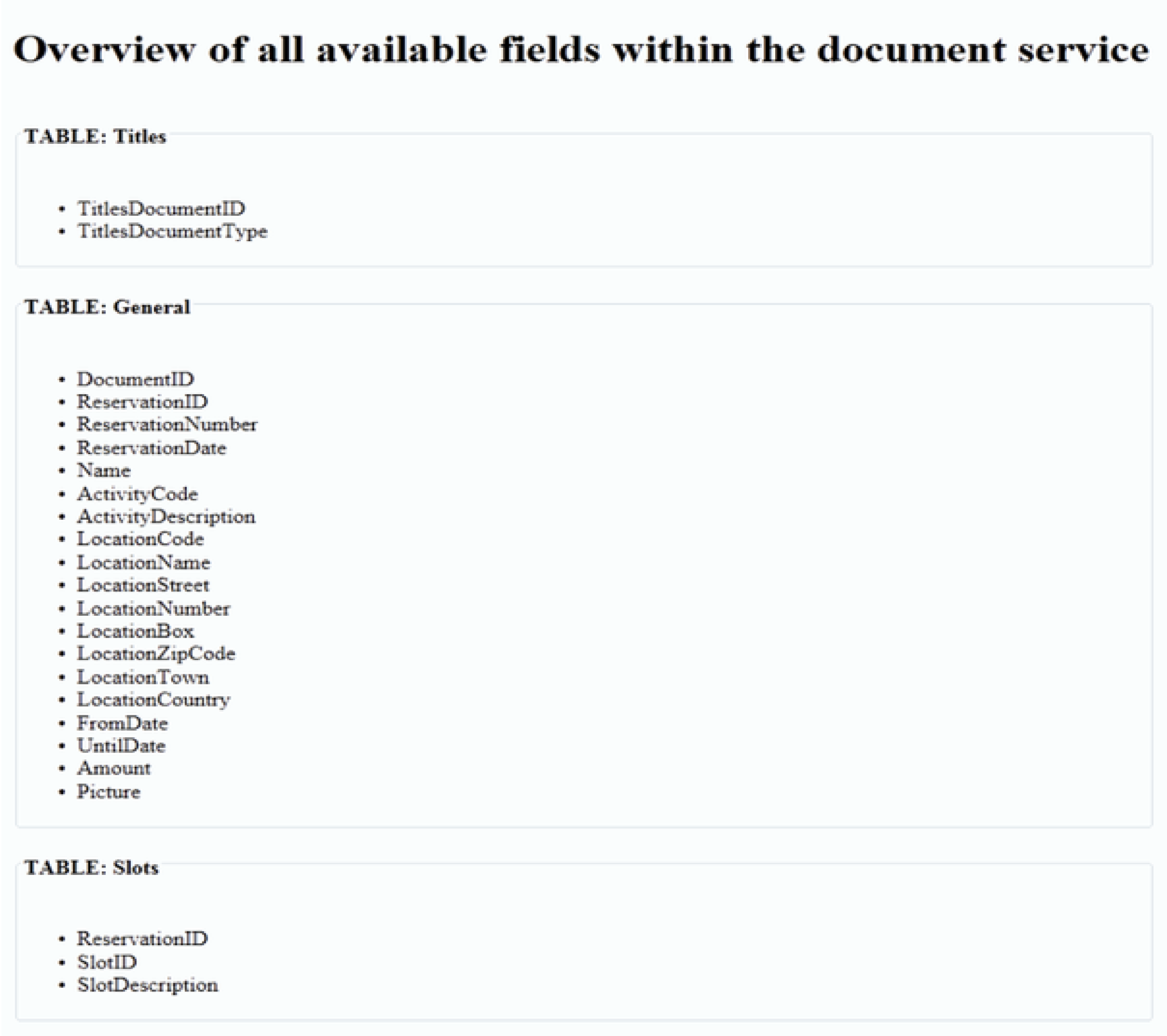
Url	Met hod	Description
ActivitySubscriptions/{ShopID}/{Language}/{SubscriptionId}	GET	Returns the Document for (Activity Subscription).
Childcare/{ShopId}/{Language}/{ChildCareCentreEntryReservationId}	GET	Returns the Documents for a Childcare reservation
CultureEvents/{ShopID}/{Language}/{CustomerId}/{CultureEventID}/{From}/{Until}	GET	Returns the Document for (Culture Event).
CultureEvents/{ShopID}/{Language}/{ReservationId}	GET	Returns the Document for (Culture Event).
ETicketBySubscription/{ShopId}/{Language}/{SubscriptionID}	GET	Returns the Document for a specific E-Ticket
ETickets/{ShopID}/{Language}/{SalesLineID}	GET	Returns the Documents for (E-Ticket)
Files/{ShopId}/{Language}/{FileId}	GET	Returns the Documents for a file
Fitness/{ShopID}/{Language}/{FitnessReservationID}	GET	Returns the Documents for (Fitness Reservation).
Help/ActivitySubscriptions/{ShopId}/{Language}	GET	Help page for GetActivitySubscriptionDocument service
Help/Childcare/{ShopId}/{Language}	GET	Help page for GetChildcareConfirmationDocument service
Help/Files/{ShopId}/{Language}	GET	Help page for GetFileConfirmationDocument service
Help/Fitness/{ShopID}/{Language}	GET	Help page for GetFitnessDocument service.
Help/OrganisedVisits/{ShopId}/{Language}	GET	Help page for GetOrganisedVisitDocument service
Help/Rental/{ShopID}/{Language}	GET	Help page for GetRentalDocument service.
Help/ReservationProof/{ShopId}/{Language}	GET	Help page for GetReservationDocument service - Single reservation
Help/SerieReservationProof/{ShopId}/{Language}	GET	Help page for GetReservationDocument service - Serie reservation
OrganisedVisits/{ShopId}/{Language}/{OrganisedVisitId}	GET	Returns the Documents for (Organised visit)
Rental/{ShopID}/{Language}/{RentalHeaderID}	GET	Returns the Documents for (Rental Order).
ReservationProof/{ShopId}/{Language}/{SalesLineId}/{ReservationId}/{RecurringReservationId}	GET	Returns the Document for (Place Reservations) by Salesline ID, Reservation ID or Recurring Reservation (Serie) ID
Subscriptions/{ShopID}/{Language}/{SubscriptionId}	GET	Returns the subscription document.
ActivityOrder/{ShopID}/{Language}/{OrderId}	GET	Returns the activity order document.
Help/ActivityOrder/{ShopID}/{Language}	GET	Help page for GetActivityOrder service
ScenarioSession/{ShopId}/{Language}/{ScenarioSessionID}	GET	Returns the scenario session document.
Help/ScenarioSession/{ShopId}/{Language}	GET	Help page for the ScenarioSession service.

ETickets

For expositions, amongst others, an access ticket can be called up per sale, on the condition that an access has been bought and that a template is linked to this article in ReCreateX.
The principle is analogous to the previous document services.
Format of the URL:
http://\{server}/WebshopDocumentService/ETickets/{ShopId}/{Language}/{SalesLineID}
The SalesLineID is the ID that is generated out of the CheckOutBasketResult.SalesItem [...] ID after CheckOutBasket.

Fitness tickets

If you reserve fitness activities, you can generate a ticket per reservation on condition that a template is linked in the general fitness parameters. Different fields can be used in this template that can be listed using a Help service:
[**Fout! De hyperlinkverwijzing is ongeldig.**



Overview of all available fields within the document service

TABLE: Titles

- TitlesDocumentID
- TitlesDocumentType

TABLE: General

- DocumentID
- ReservationID
- ReservationNumber
- ReservationDate
- Name
- ActivityCode
- ActivityDescription
- LocationCode
- LocationName
- LocationStreet
- LocationNumber
- LocationBox
- LocationZipCode
- LocationTown
- LocationCountry
- FromDate
- UntilDate
- Amount
- Picture

TABLE: Slots

- ReservationID
- SlotID
- SlotDescription

The principle for finding the fitness tickets is analogous to previous document services.
Format of the URL:
http://\{server}/WebshopDocumentService/Fitness/{ShopId}/{Language}/{FitnessReservationID}
The FitnessReservationID is the ID that is generated out of the CheckOutBasketResult.SalesItem [...] ID after CheckOutBasket.

General

The document service can be used to find and get PDF documents for basket sales.

The following method can be used to generate a PDF document from a URL. The following parameters will be given:

- The URL

e.g.: *http://services.recreatex.be/WebShopDocumentService.svc/CultureEvents/9843ADF4*-4D8A-9D5Q-D4D6-D88S5C4DPVG8/NL/58S1D4D8-VSQ5-6D1C-1D44-M158443F8SD1

- the absolute path of the document to be generated

e.g.: *C:\Temp\Event_58S1D4D8-VSQ5-6D1C-1D44-M158443F8SD1.pdf*

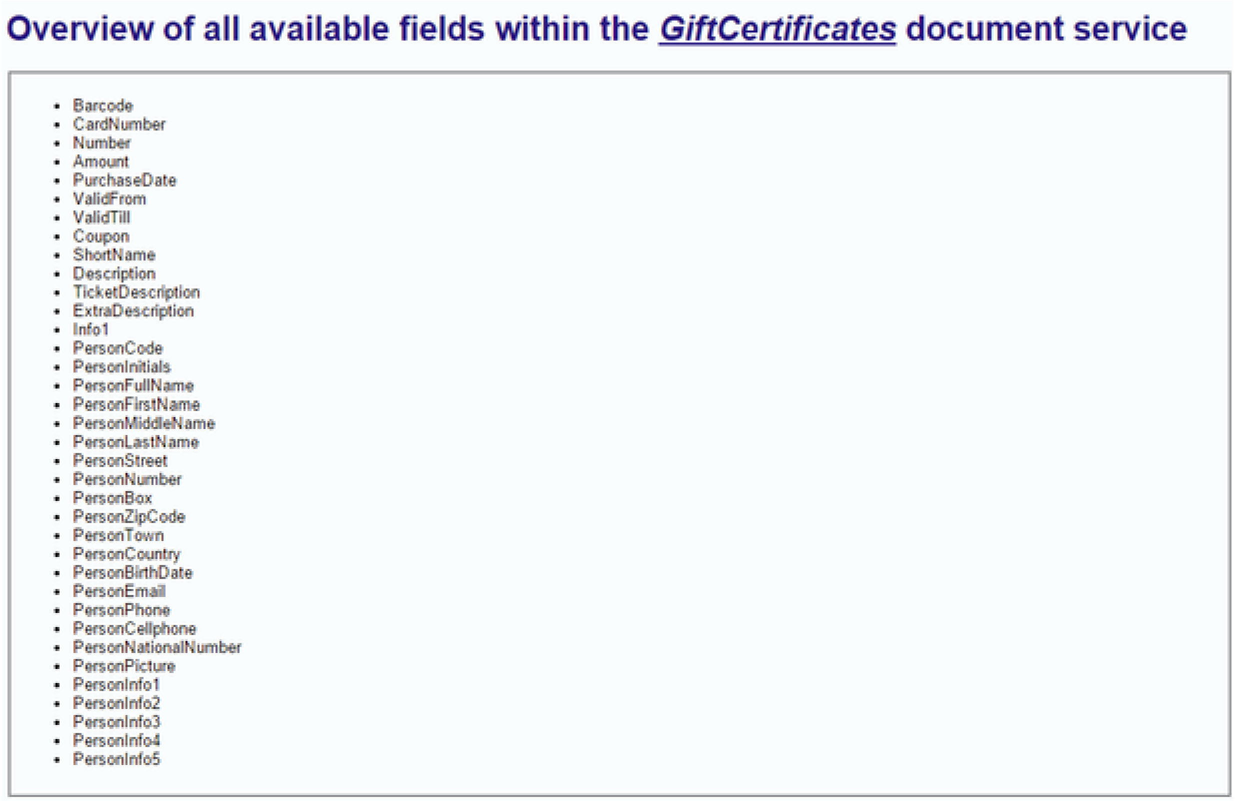
- Time out during execution of the transaction (in seconds)

e.g.: *10*

```
private static void SavePdfFileFromURL(string url, string destinationFileName, int timeoutInSeconds)
{
// Create a web request to the URL
var MyRequest = (HttpWebRequest)WebRequest.Create(url);
MyRequest.Timeout = timeoutInSeconds * 1000;
try
{
// Get the web response
var MyResponse = (HttpWebResponse)MyRequest.GetResponse();
// Make sure the response is valid
if (HttpStatusCode.OK == MyResponse.StatusCode)
{
// Open the response stream
using (Stream MyResponseStream = MyResponse.GetResponseStream())
{
// Open the destination file
using (var MyFileStream = new FileStream(destinationFileName, FileMode.OpenOrCreate, FileAccess.Write))
{
var MyBuffer = new byte[4096];
int BytesRead;
// Read the chunk of the web response into the buffer
while (0 < (BytesRead = MyResponseStream.Read(MyBuffer, 0, MyBuffer.Length)))
{
// Write the chunk from the buffer to the file
MyFileStream.Write(MyBuffer, 0, BytesRead);
}
MyFileStream.Flush();
MyFileStream.Close();
}
}
}
}
catch (Exception err)
{
throw new Exception('Error saving file from URL:' + err.Message, err);
}
return;
}
```

Gift certificates

The gift certificate document service prints all gift certificates that were created within a specific sale. Different fields can be used in this template that can be listed using a Help service:



Overview of all available fields within the *GiftCertificates* document service

- Barcode
- CardNumber
- Number
- Amount
- PurchaseDate
- ValidFrom
- ValidTill
- Coupon
- ShortName
- Description
- TicketDescription
- ExtraDescription
- Info1
- PersonCode
- PersonInitials
- PersonFullName
- PersonFirstName
- PersonMiddleName
- PersonLastName
- PersonStreet
- PersonNumber
- PersonBox
- PersonZipCode
- PersonTown
- PersonCountry
- PersonBirthDate
- PersonEmail
- PersonPhone
- PersonCellphone
- PersonNationalNumber
- PersonPicture
- PersonInfo1
- PersonInfo2
- PersonInfo3
- PersonInfo4
- PersonInfo5

Format of the URL:
http://\{server}/WebshopDocumentService/GiftCertificates/{ShopId}/{Language}/{SalesLineID}
The SalesLineID is the ID that is generated out of the CheckOutBasketResult.SalesItem [...] ID after CheckOutBasket.

Invoices

File	Modified
FACTUUR.docx Example document	Oct 16, 2015 by Former user

Drag and drop to upload or browse for files

Merge fields

Introduction

MS Office Word "merge fields" are used to include specific data from ReCreateX into the PDF document e-template.

The document service dynamically replaces the merge fields by the correct data when loading the PDF document.

Create merge fields

Merge fields can be added manually to you Word template from within the "Mailings" section, button "Insert Merge Field".

For more information, see

- WORD 2007: https://support.office.com/en-us/article/Insert-and-format-field-codes-in-Word-2007-1C27736E-9624-45DA-946C-2B3050A9491F
- WORD 2010: https://support.office.com/en-us/article/Insert-and-format-field-codes-in-Word-2010-7e9ea3b4-83ec-4203-9e66-4efc027f2cf3
- WORD 2013: https://support.office.com/en-us/article/Insert-mail-merge-fields-9a1ab5e3-2d7a-420d-8d7e-7cc26f26acff
- Field codes: https://support.office.com/en-us/article/Field-codes-in-Word-1ad6d91a-55a7-4a8d-b535-cf7888659a51

Example

Formatting

Additional formatting can be applied to merge fields. Formatting can be applied to numeric, date, time, ... values.

For more information, see

- http://iits.haverford.edu/documents/mail-merges-formatting-numbers-in-a-merge-field/
- https://knowledgecenter.zuora.com/CB_Billing/IA_Invoices/Creating_a_Custom_Invoice_Template/B_Define_the_Format_for_Date_and_Number_Fields

Example

Conditional formatting

Merge fields allow conditional formatting.
This means that the value which will be visualized can depend on the value of another field.

For more information, see

- https://support.office.com/en-us/article/Field-codes-IF-field-9f79e82f-e53b-4ff5-9d2c-ae3b22b7eb5e

- http://word.mvps.org/faqs/mailmerge/mmergeiffields.htm

Example

Advanced formatting

- Grouping
- Merging tables
- Calculating table totals

Samples: Catalogue Mailmerge.zip

Reseller documents

Coupons (Reseller)

The fields which can be used within reseller PDF documents are the same as the original one.
See Coupon tickets for more information.

However, the format of the URL is different:
http://\{server}/WebshopDocumentService/TraderCoupons/{ShopId}/{Language}/{TraderId}/{SalesLineID}

The SalesLineID is the ID that is generated out of the CheckOutBasketResult.SalesItem [...] ID after CheckOutBasket.

ETickets (Reseller)

The fields which can be used within reseller PDF documents are the same as the original one.
See ETickets for more information.

However, the format of the URL is different:
http://\{server}/WebshopDocumentService/TraderETickets/{ShopId}/{Language}/{TraderId}/{OrganisedVisitId}

The SalesLineID is the ID that is generated out of the CheckOutBasketResult.SalesItem [...] ID after CheckOutBasket.

Gift certificates (Reseller)

The fields which can be used within reseller PDF documents are the same as the original one.
See Gift certificates for more information.

However, the format of the URL is different:
http://\{server}/WebshopDocumentService/TraderGiftCertificates/{ShopId}/{Language}/{TraderId}/{OrganisedVisitId}

The SalesLineID is the ID that is generated out of the CheckOutBasketResult.SalesItem [...] ID after CheckOutBasket.

Tickets for exposition visits (Reseller)

The fields which can be used within reseller PDF documents are the same as the original one.
See Tickets for exposition visits for more information

However, the format of the URL is different:
http://\{server}/WebshopDocumentService/TraderOrganisedVisits/{ShopId}/{Language}/{TraderId}/{OrganisedVisitId}

After check-out the OrganisedVisitId can be found in the ExpositionSalesItem object.

Scenario session document

The scenario session document generates an overview of all fields as defined within the scenario wizard.
Following fields can be used in this template that can be listed using a Help service:
Format of the URL:
*http://\{server}/WebshopDocumentService/ScenarioSession/{ShopId}/{Language}/{*ScenarioSession*ID}*
The ScenarioSessionID is the ID of the scenario session as defined in the ScenarioValidationResult object when saving the scenario session.

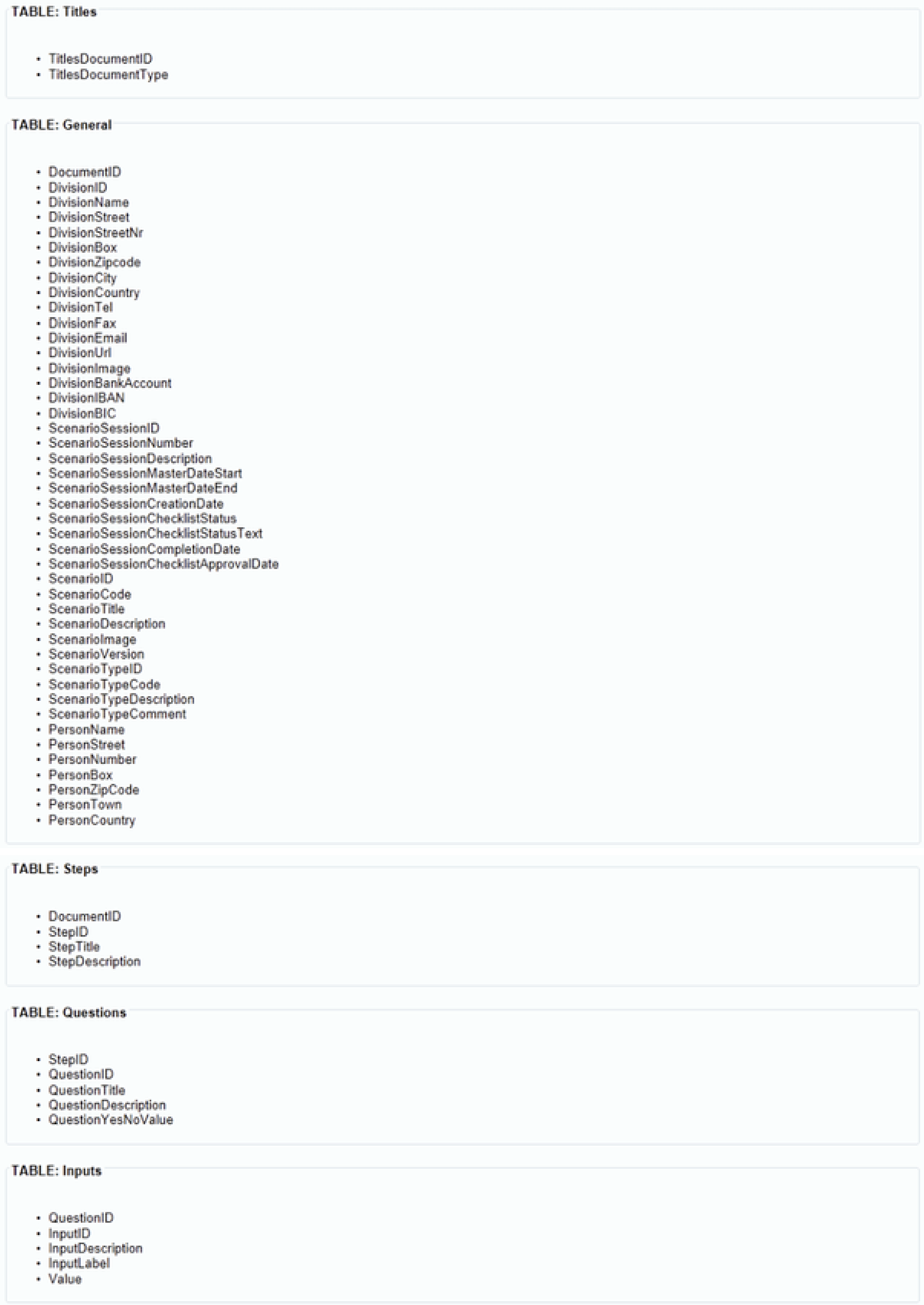


TABLE: Titles

- TitlesDocumentID
- TitlesDocumentType

TABLE: General

- DocumentID
- DivisionID
- DivisionName
- DivisionStreet
- DivisionStreetNr
- DivisionBox
- DivisionZipcode
- DivisionCity
- DivisionCountry
- DivisionTel
- DivisionFax
- DivisionEmail
- DivisionUrl
- DivisionImage
- DivisionBankAccount
- DivisionIBAN
- DivisionBIC
- ScenarioSessionID
- ScenarioSessionNumber
- ScenarioSessionDescription
- ScenarioSessionMasterDateStart
- ScenarioSessionMasterDateEnd
- ScenarioSessionCreationDate
- ScenarioSessionChecklistStatus
- ScenarioSessionChecklistStatusText
- ScenarioSessionCompletionDate
- ScenarioSessionChecklistApprovalDate
- ScenarioID
- ScenarioCode
- ScenarioTitle
- ScenarioDescription
- ScenarioImage
- ScenarioVersion
- ScenarioTypeID
- ScenarioTypeCode
- ScenarioTypeDescription
- ScenarioTypeComment
- PersonName
- PersonStreet
- PersonNumber
- PersonBox
- PersonZipCode
- PersonTown
- PersonCountry

TABLE: Steps

- DocumentID
- StepID
- StepTitle
- StepDescription

TABLE: Questions

- StepID
- QuestionID
- QuestionTitle
- QuestionDescription
- QuestionYesNoValue

TABLE: Inputs

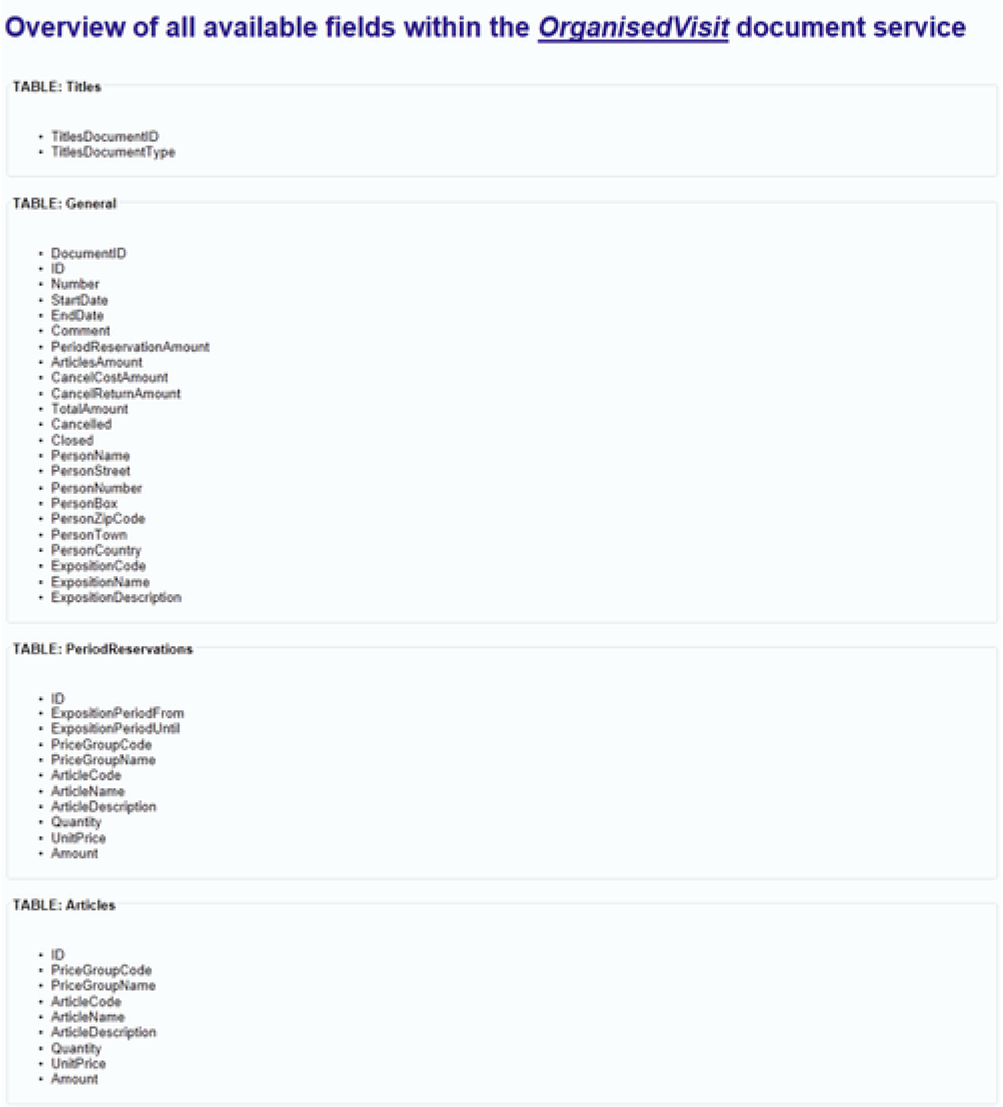
- QuestionID
- InputID
- InputDescription
- InputLabel
- Value

Specific ETickets

The document service above prints all tickets of a specific sale. It is also possible to generate the document for 1 specific e-Ticket. For this purpose, you can use the document service below:
Format of the URL:
http://\{server}/WebshopDocumentService/ETicketBySubscription/{ShopId}/{Language}/{SubscriptionID}

Tickets for exposition visits

If you register an exposition visit, you can ask for a ticket on condition that a template is linked to the exposition (or the exposition type) in ReCreateX. In this template different fields can be used that can be listed using a Help service:
http://\{server}/WebshopDocumentService/Help/OrganisedVisits/{ShopId}/{Language}



Overview of all available fields within the *OrganisedVisit* document service

TABLE: Titles

- TitlesDocumentID
- TitlesDocumentType

TABLE: General

- DocumentID
- ID
- Number
- StartDate
- EndDate
- Comment
- PeriodReservationAmount
- ArticlesAmount
- CancelCostAmount
- CancelReturnAmount
- TotalAmount
- Cancelled
- Closed
- PersonName
- PersonStreet
- PersonNumber
- PersonBox
- PersonZipCode
- PersonTown
- PersonCountry
- ExpositionCode
- ExpositionName
- ExpositionDescription

TABLE: PeriodReservations

- ID
- ExpositionPeriodFrom
- ExpositionPeriodUntil
- PriceGroupCode
- PriceGroupName
- ArticleCode
- ArticleName
- ArticleDescription
- Quantity
- UnitPrice
- Amount

TABLE: Articles

- ID
- PriceGroupCode
- PriceGroupName
- ArticleCode
- ArticleName
- ArticleDescription
- Quantity
- UnitPrice
- Amount

The principle for finding the tickets for registrations for activities is analogous to previous document services.
Format of the URL:
http://\{server}/WebshopDocumentService/OrganisedVisits/{ShopId}/{Language}/{OrganisedVisitId}
After check-out the OrganisedVisitId can be found in the ExpositionSalesItem object.

Tickets for Files

If you order file articles, you can generate a ticket per file on condition that a template is linked in the file type of the file. Different fields can be used in this template that can be listed using a Help service:
[**Fout! De hyperlinkverwijzing is ongeldig.**

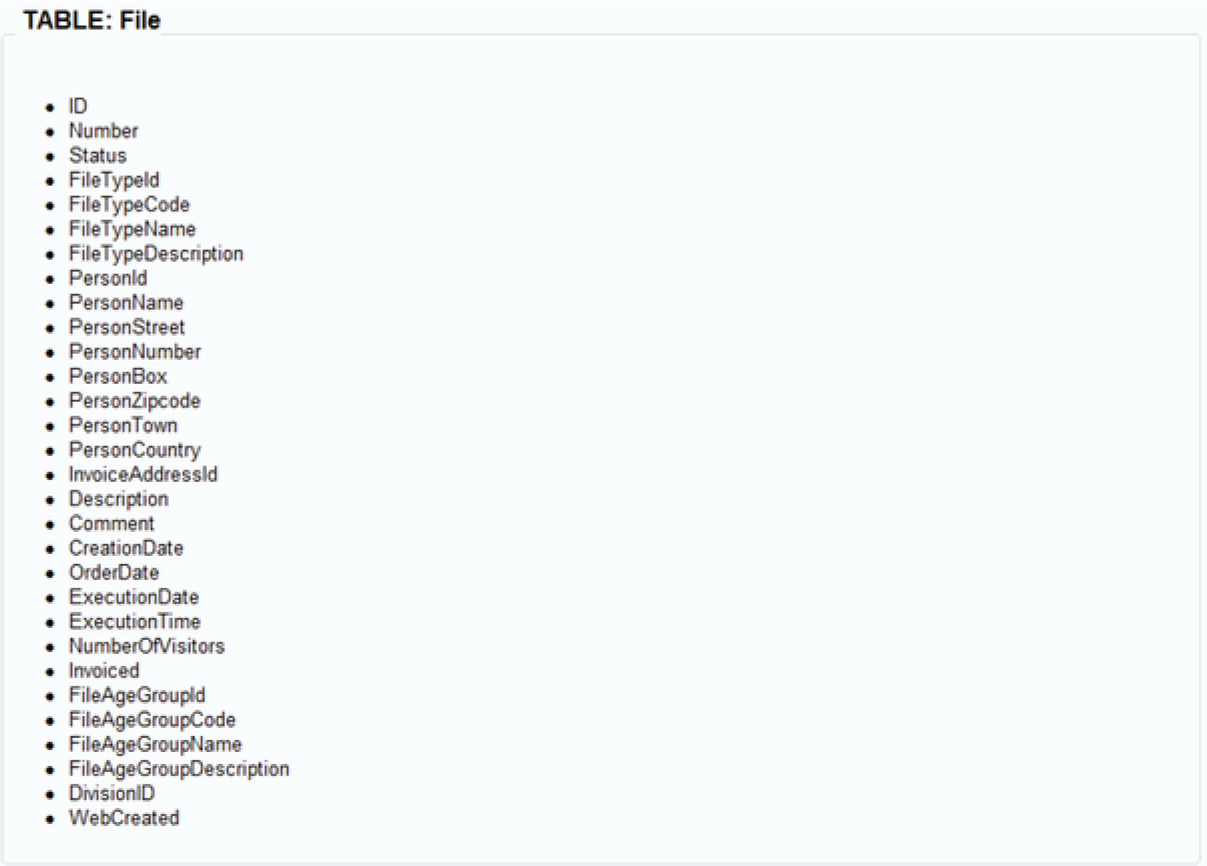


TABLE: File

- ID
- Number
- Status
- FileTypeId
- FileTypeCode
- FileTypeName
- FileTypeDescription
- PersonId
- PersonName
- PersonStreet
- PersonNumber
- PersonBox
- PersonZipcode
- PersonTown
- PersonCountry
- InvoiceAddressId
- Description
- Comment
- CreationDate
- OrderDate
- ExecutionDate
- ExecutionTime
- NumberOfVisitors
- Invoiced
- FileAgeGroupId
- FileAgeGroupCode
- FileAgeGroupName
- FileAgeGroupDescription
- DivisionID
- WebCreated

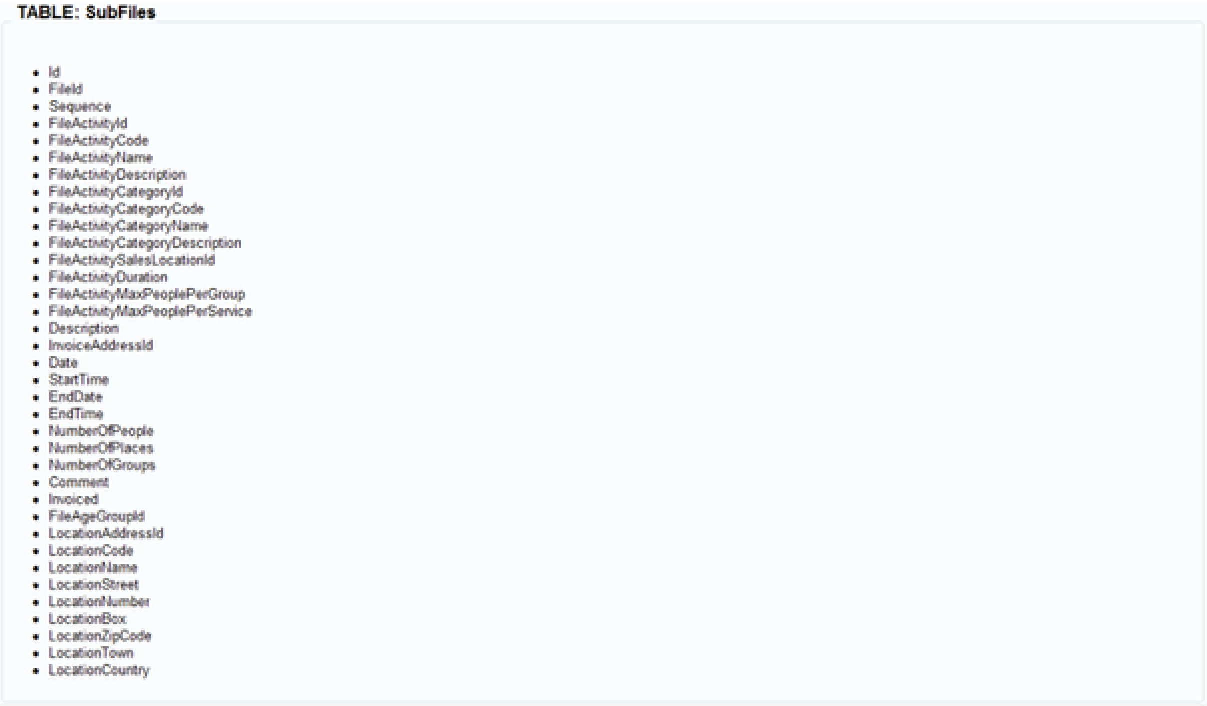


TABLE: SubFiles

- Id
- FileId
- Sequence
- FileActivityId
- FileActivityCode
- FileActivityName
- FileActivityDescription
- FileActivityCategoryId
- FileActivityCategoryCode
- FileActivityCategoryName
- FileActivityCategoryDescription
- FileActivitySalesLocationId
- FileActivityDuration
- FileActivityMaxPeoplePerGroup
- FileActivityMaxPeoplePerService
- Description
- InvoiceAddressId
- Date
- StartTime
- EndDate
- EndTime
- NumberOfPeople
- NumberOfPlaces
- NumberOfGroups
- Comment
- Invoiced
- FileAgeGroupId
- LocationAddressId
- LocationCode
- LocationName
- LocationStreet
- LocationNumber
- LocationBox
- LocationZipCode
- LocationTown
- LocationCountry

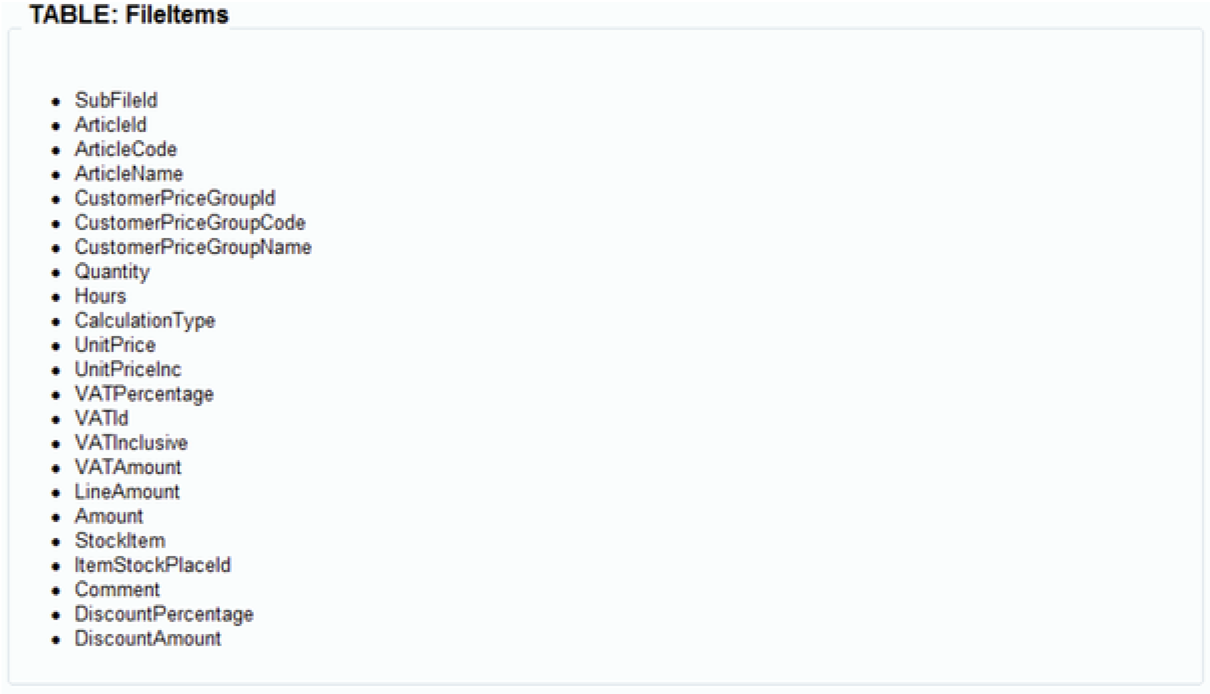


TABLE: FileItems

- SubFileId
- ArticleId
- ArticleCode
- ArticleName
- CustomerPriceGroupId
- CustomerPriceGroupCode
- CustomerPriceGroupName
- Quantity
- Hours
- CalculationType
- UnitPrice
- UnitPriceInc
- VATPercentage
- VATId
- VATInclusive
- VATAmount
- LineAmount
- Amount
- StockItem
- ItemStockPlaceId
- Comment
- DiscountPercentage
- DiscountAmount

The principle for finding the file tickets is analogous to previous document services.
Format of the URL:
http://\{server}/WebshopDocumentService/File/{ShopId}/{Language}/{FileID}
The FileID is the ID that is generated out of the CheckOutBasketResult.SalesItem [...] ID after CheckOutBasket.

Tickets for rental orders

If you order rental articles, you can generate a ticket per order on condition that a template is linked in the general rental parameters. Different fields can be used in this template that can be listed using a Help service:
http://\{server}/WebshopDocumentService/Help/Rental/{ShopId}/{Language}

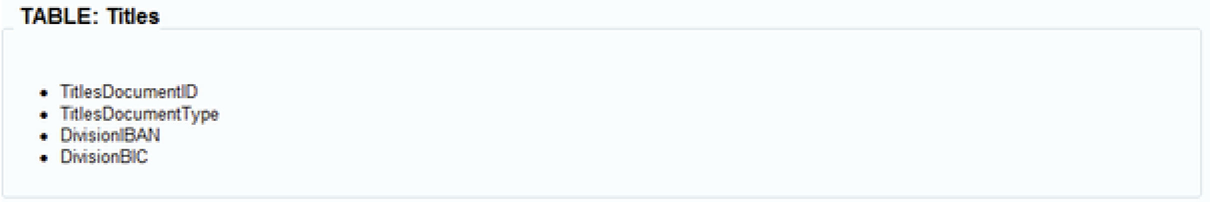


TABLE: Titles

- TitlesDocumentID
- TitlesDocumentType
- DivisionIBAN
- DivisionBIC

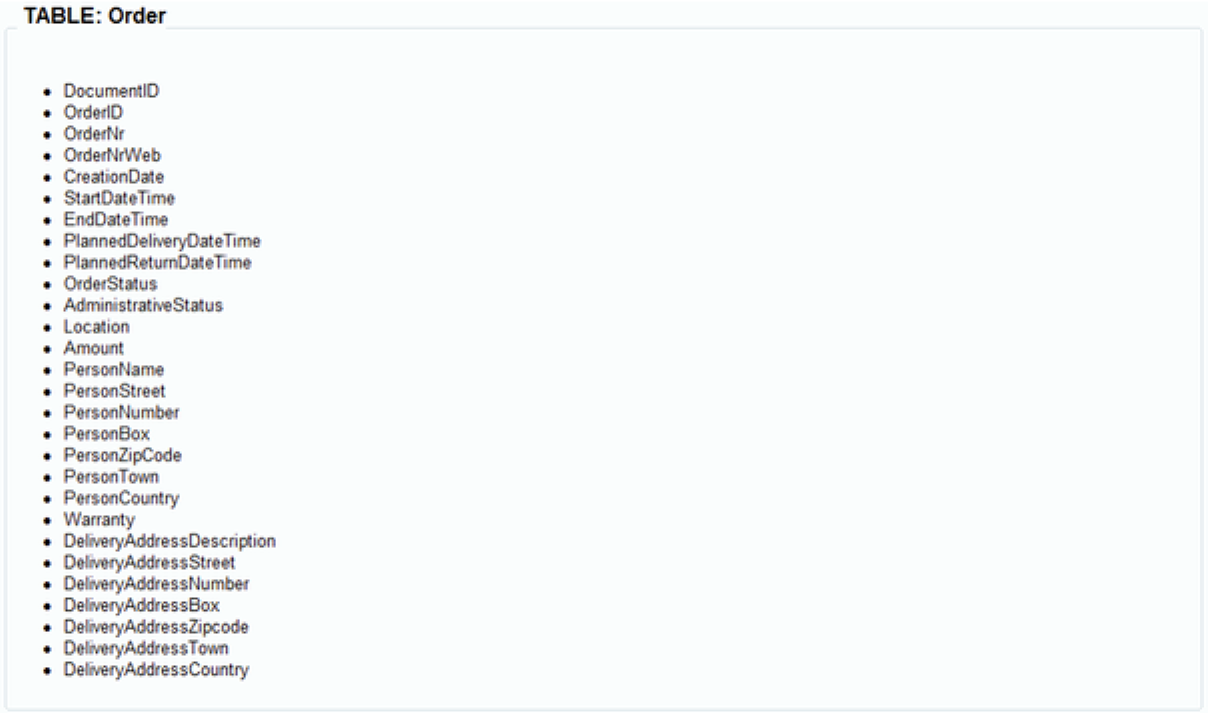


TABLE: Order

- DocumentID
- OrderID
- OrderNr
- OrderNrWeb
- CreationDate
- StartDateTime
- EndDateTime
- PlannedDeliveryDateTime
- PlannedReturnDateTime
- OrderStatus
- AdministrativeStatus
- Location
- Amount
- PersonName
- PersonStreet
- PersonNumber
- PersonBox
- PersonZipCode
- PersonTown
- PersonCountry
- Warranty
- DeliveryAddressDescription
- DeliveryAddressStreet
- DeliveryAddressNumber
- DeliveryAddressBox
- DeliveryAddressZipcode
- DeliveryAddressTown
- DeliveryAddressCountry

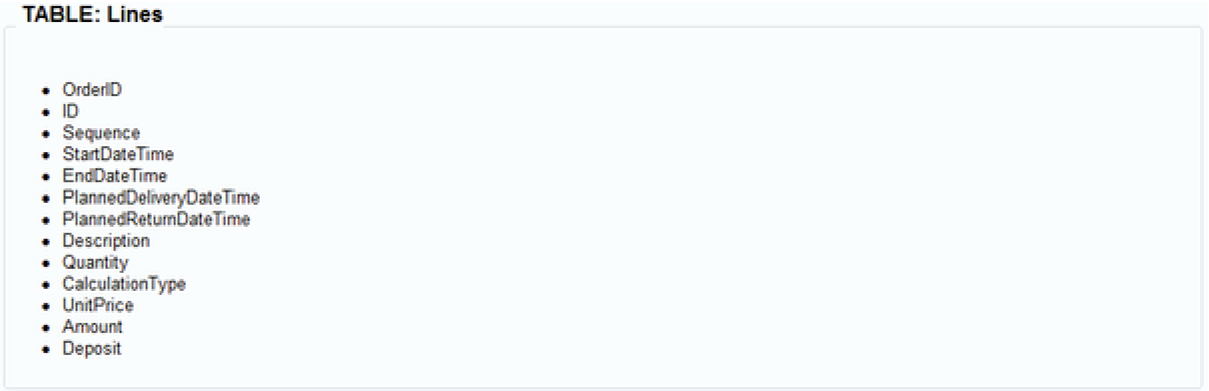


TABLE: Lines

- OrderID
- ID
- Sequence
- StartDateTime
- EndDateTime
- PlannedDeliveryDateTime
- PlannedReturnDateTime
- Description
- Quantity
- CalculationType
- UnitPrice
- Amount
- Deposit

The principle for finding the rental tickets is analogous to previous document services.
Format of the URL:
http://\{server}/WebshopDocumentService/Rental/{ShopId}/{Language}/{RentalHeaderID}
The RentalHeaderID is the ID that is generated out of the CheckOutBasketResult.SalesItem [...] ID after CheckOutBasket.

Tickets of registrations

If you register for activities, you can generate a ticket per registration on condition that a template is linked to this activity in ReCreateX. Different fields can be used in this template that can be listed using a Help service:
http://\{server}/WebshopDocumentService/Help/ActivitySubscription/{ShopId}/{Language}

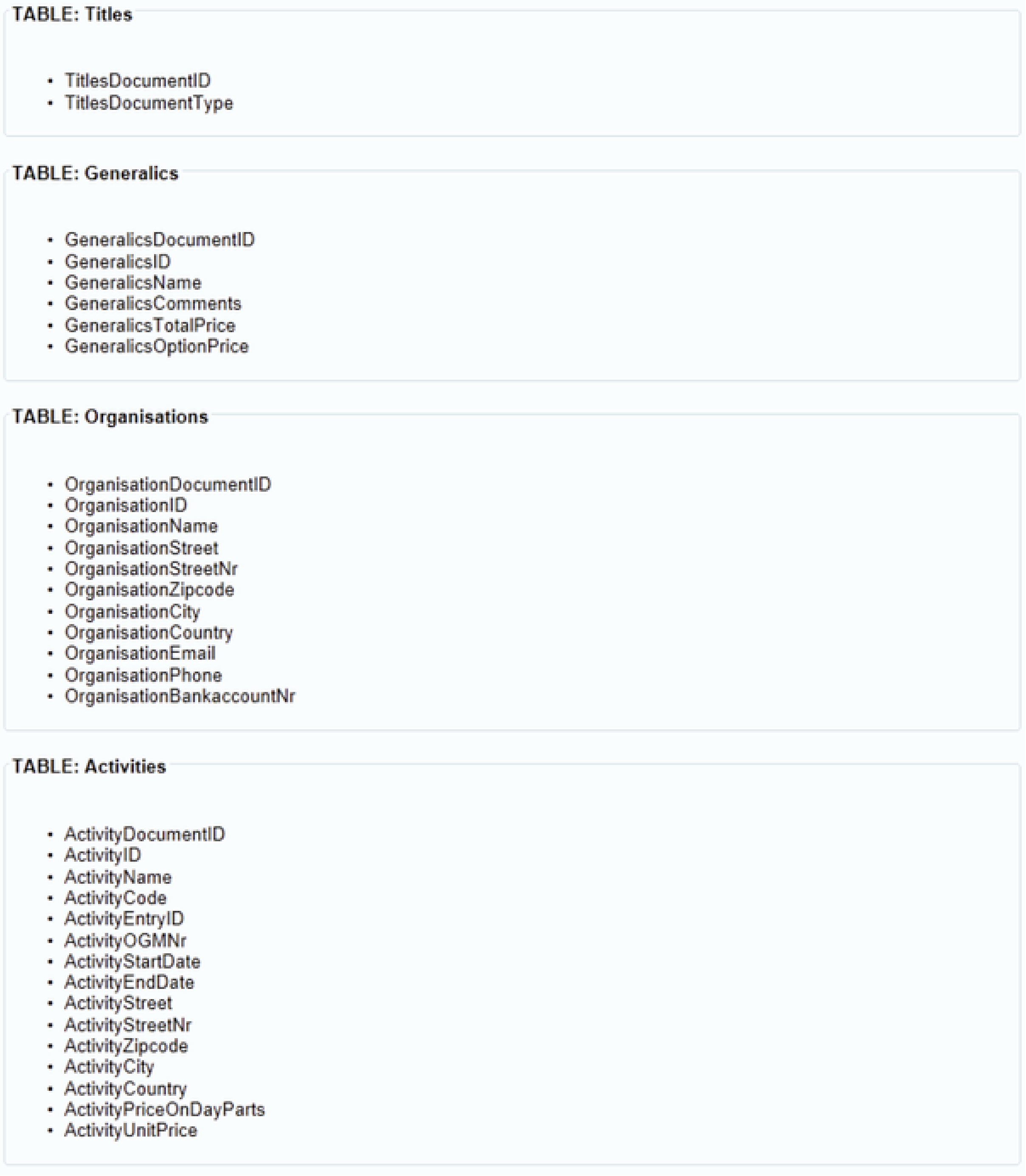


TABLE: Titles

- TitlesDocumentID
- TitlesDocumentType

TABLE: Generalics

- GeneralicsDocumentID
- GeneralicsID
- GeneralicsName
- GeneralicsComments
- GeneralicsTotalPrice
- GeneralicsOptionPrice

TABLE: Organisations

- OrganisationDocumentID
- OrganisationID
- OrganisationName
- OrganisationStreet
- OrganisationStreetNr
- OrganisationZipcode
- OrganisationCity
- OrganisationCountry
- OrganisationEmail
- OrganisationPhone
- OrganisationBankaccountNr

TABLE: Activities

- ActivityDocumentID
- ActivityID
- ActivityName
- ActivityCode
- ActivityEntryID
- ActivityOGMNr
- ActivityStartDate
- ActivityEndDate
- ActivityStreet
- ActivityStreetNr
- ActivityZipcode
- ActivityCity
- ActivityCountry
- ActivityPriceOnDayParts
- ActivityUnitPrice

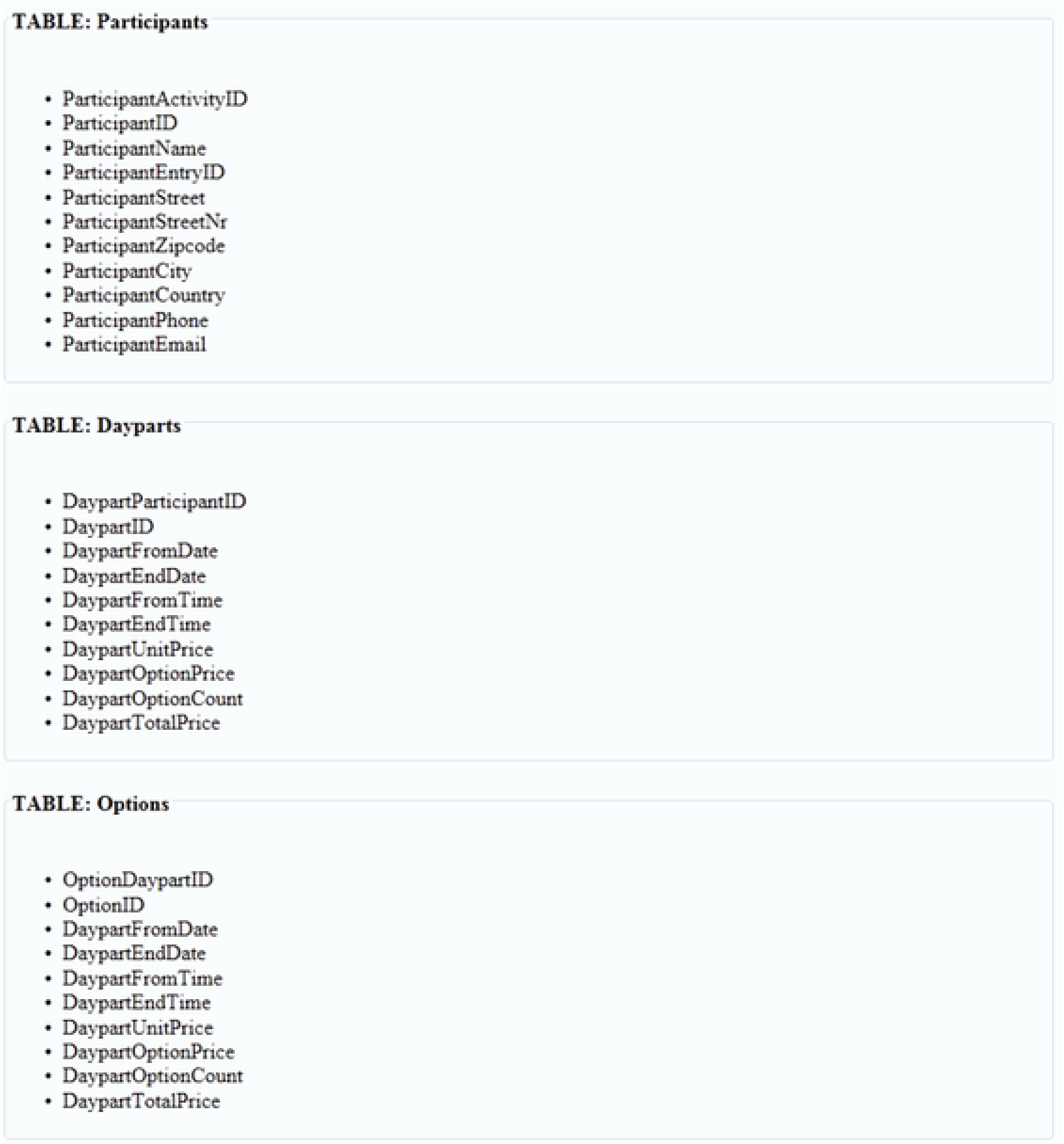


TABLE: Participants

- ParticipantActivityID
- ParticipantID
- ParticipantName
- ParticipantEntryID
- ParticipantStreet
- ParticipantStreetNr
- ParticipantZipcode
- ParticipantCity
- ParticipantCountry
- ParticipantPhone
- ParticipantEmail

TABLE: Dayparts

- DaypartParticipantID
- DaypartID
- DaypartFromDate
- DaypartEndDate
- DaypartFromTime
- DaypartEndTime
- DaypartUnitPrice
- DaypartOptionPrice
- DaypartOptionCount
- DaypartTotalPrice

TABLE: Options

- OptionDaypartID
- OptionID
- DaypartFromDate
- DaypartEndDate
- DaypartFromTime
- DaypartEndTime
- DaypartUnitPrice
- DaypartOptionPrice
- DaypartOptionCount
- DaypartTotalPrice

The principle for finding the tickets for registrations for activities is analogous to previous document services.
Format of the URL:
http://\{server}/WebshopDocumentService/ ActivitySubscriptions/{ShopId}/{Language}/{ SubscriptionId}
The SubscriptionId is the ID that is generated out of the CheckOutBasketResult.SalesItem [...] ID after CheckOutBasket.

23 Vending machine

FindVendingMachineBenefits

Calling up this Webservice operation will visualise the benefits for the given person card id.

Request

The request consists of the service context and the VendingMachineBenefitsSearchCriteria.

VendingMachineBenefitsSearchCriteria

Name	Type	Description	Remarks
PersonCardId	Guid	The person card id of the user	

Response

The response consists a list of VendingMachineBenefit

VendingMachineBenefit

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the benefit	
Description	String	The description of the benefit	
Translations	List<VendingMachineTranslations>	The translations from RCX	
TicketLayout	int	Ticket layout linked to the benefit	
BenefitType	VendingMachineBenefitType	Type of the benefit	
IsActive	bool	Determines if the benefit is available	
Details	VendingMachineBenefitDetails	Deatils of the benefit	Only filled in if the type is a coupon or a voucher

BenefitType

Name	Type	Description	Remarks
Incentive	0	Enumerator position 0	
Coupon	1	Enumerator position 1	
Voucher	2	Enumerator position 2	

VendingMachineBenefitDetails

Name	Type	Description	Remarks
ValidFrom	DateTime	The start date when the benefit is valid	
ValidTill	DateTime	The end date when the benefit is valid	
ValidTimeFrom	DateTime	The start time when the benefit is valid	
ValidTimeFrom	DateTime	The end time when the benefit is valid	
ValidDayOfWeek	WeekDay	Determines when the benefit is valid	
NumberOfUsages	int	Determines how much the benefit can be used	
Applicability	int	Determines how the benefit can be used	
SalesChannel	int	How has the coupon been sold	
DiscountType	int	Determines the discount type	
Discount	decimal	Determines the discount	
UniquePerSales	bool	Determines if the benefit can be used per sale	
PreSalesOnly	bool	Determines if the benefit is pre sale only	
MaxUsagesPerSale	int	Determines the maximum usages per sale	
MinimumCredits	int	Determines the minimum credits	
CreditsPerUsage	int	Determines the credits per usage	
DiscountAmountPerProduct	bool	Determines if the discount is per prodcut	
MaxUsagesPerProduct	int	Determines the max usages per product	
CouponNumberUsed	int	Determines how much the benefit has been used	

FindVendingMachineCultureEvents

Calling up this Webservice operation will visualise the desired CultureEvents on the basis of the linked parameters.

Request

The FindVendingMachineCultureEvents consists of the service context and an VendingMachineCultureEventSearchCriteria object.

VendingMachineCultureEventSearchCriteria

Name	Type	Description	Remarks
Includes	VendingMachineCultureEventsIncludes	Makes clear how images are retrieved	

VendingMachineCultureEventsIncludes

This object will determine which type of images of the CultureEvents will be called up.

Name	Type	Description	Remarks
Image	Boolean	Request the image as a complete image	
ImageUri	Boolean	Request the image as a uri	

Response

The response is a list of VendingMachineCultureEvents. A VendingMachineCultureEvent is the same as a CultureEvent but more compact.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the CultureEvent.	
Description	String	The description of the CultureEvent.	
From	DateTime	The start date of the CultureEvent	
Until	DateTime	The end date of the CultureEvent	
Hall	String	The name of the Hall	
Image	Picture	The picture of the CultureEvent	
ImageUri	String	The Url on which the image can be found	
RemainingSeats	Int	The number of available seats (= maximum available seats – reserved seats – blocked seats – seats in option)	
SoldOut	Bool	Indicates if the CultureEvent is sold out	
Translations	List<VendingMachineTranslation>	The translations available from RCX	
Prices	List<VendingMachineCulturePrice>	All the prices on a CultureEvent	
TicketLayout	Int	What ticket type will be printed	
AllowOverbooking	Bool	Is Overbooking allowed on the CultureEvent	
RemainingSeatsTotal	Int	The number of total available seats (= all the seats – reserved seats – blocked seats – seats in option)	

VendingMachineTranslations

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the translation.	
LanguageCode	String	The language code to determine the language	
Description	String	The actual translation in the language designated	

VendingMachineCulturePrice

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the CulturePrice.	
Description	String	The price group	
Price	Decimal	The actual price for the price group	

FindVendingMachineIncentives

Calling up this Webservice operation will visualise the Incentive tickets on the basis of the linked parameters.

Request

The FindVendingMachineCultureIncentives consists of the service context and an VendingMachineIncentivesSearchCriteria object.

VendingMachineIncentivesSearchCriteria

Name	Type	Description	Remarks
PersonCardId	Guid	The unique identifier of the Card.	
SalesLineID	Guid	The unique identifier of the SalesLine	

Response

The response is a list of VendingMachineCultureEvents.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the Incentive.	
Description	String	The description of the Incentive.	
Translations	List<VendingMachineTranslation>	The translations from RCX	
TicketLayout	Int	What ticket type will be printed	

FindVendingMachinePersonSubscriptions

Calling up this Webservice operation will visualise the Subscriptions tickets on the basis of the linked parameters.

Request

The FindVendingMachineSubscriptions consists of the service context and an VendingMachineSubscriptionsSearchCriteria object.

VendingMachineSubscriptionsSearchCriteria

Name	Type	Description	Remarks
PersonCardId	Guid	The unique identifier of the Card.	
UseHeadOfFamily	Bool	Return subscriptions of the head of the family	
OnlyCarrierSwitchActive	Bool	Return only subscriptions where CarrierSwitch is active	7.5.1

Response

The response is a list of VendingMachineSubscriptions.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the Subscription.	
Description	String	The description of the Subscription.	
Translations	List<VendingMachineTranslation>	The translations from RCX	
CreationDate	DateTime	When is the subscription created	
EndDate	DateTime	When expires the subscription	
LastProlongation	DateTime	Last date when the Subscription is prolonged	
ArticleId	Guid	What's the article linked to the subscription	
MultipleProlongAllowed	Bool	Is the subscription prolongable multiple times	
Price	Decimal	The price of the subscription	
TicketLayout	Int	What ticket type will be used	
NumberOfTurns	Int		
StartDate	DateTime	When starts the subscription	
CheckNumberOfTurns	Bool		
CheckPeriod	Bool		
UseActivationHoursMinutes	Bool		
Active	Bool		

ValidProlongMinDaysBeforeExpiration	Bool		
CanProlongAutoCash	Bool		
CarrierSwitchTicketLayout	Int	What ticket type will be used for carrier switch	7.5.1
CarrierSwitchMaxSwitchesReached	Bool	Max switches reached	7.5.1

FindVendingMachineSubscriptions

Calling up this Webservice operation will visualise the Subscriptions tickets on the basis of the linked parameters.

Request

The FindVendingMachineSubscriptions consists of the service context and an VendingMachineSubscriptionsSearchCriteria object.

VendingMachineSubscriptionsSearchCriteria

Name	Type	Description	Remarks
PersonCardId	Guid	The unique identifier of the Card.	
ExcludeComponents	bool	Excludes components if their main item is a Subscription when set to true	
OnlyCarrierSwitchActive	bool	Returns only Subscriptions with CarrierSwitchActive when set to true	

Response

The response is a list of VendingMachineSubscriptions.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the Subscription.	
Description	String	The description of the Subscription.	
Translations	List<VendingMachineTranslation>	The translations from RCX	
CreationDate	DateTime	When is the Subscription created	
EndDate	DateTime	When expires the Subscription	
LastProlongation	DateTime	Last date when the Subscription is prolonged	
ArticleId	Guid	What's the article linked to the subscription	
MultipleProlongAllowed	Bool	Is the subscription prolongable multiple times	
Price	Decimal	The price of the subscription	
TicketLayout	Int	What ticket type will be printed	

GetBebarmaticInfo

This will return all the information linked to a Bebarmatic parking ticket.

Request

GetBebarmaticInfoCriteria

Name	Type	Description	Remarks
TicketId	String	The identifier of the Bebarmatic ticket	

Response

GetBebarmaticInfoResult

Name	Type	Description	Remarks
TicketId	String	The identifier of the Bebarmatic ticket	
EntryTime	DateTime	EntryTime retrieved from Bebarmatic linked to the passed TicketId	

ExitTime	DateTime	ExitTime retrieved from Bebarmatic linked to the passed TicketId	
PaymentTime	DateTime	PaymentTime retrieved from Bebarmatic linked to the passed TicketId	
CurrentTime	DateTime	CurrentTime retrieved from Bebarmatic linked to the passed TicketId	
ValidUntil	DateTime	ValidUntil retrieved from Bebarmatic linked to the passed TicketId	
RepayTime	DateTime	RepayTime retrieved from Bebarmatic linked to the passed TicketId	
FreeTimeAfterEntry	String	FreeTimeAfterEntry retrieved from Bebarmatic linked to the passed TicketId	
FreeTimeAfterPayment	String	FreeTimeAfterPayment retrieved from Bebarmatic linked to the passed TicketId	
Status	String	Status retrieved from Bebarmatic linked to the passed TicketId	
Paid	Bool	Paid retrieved from Bebarmatic linked to the passed TicketId	
VisitorItem	VendingMachineArticle	The configured item used for Visitor registration configured in the CounterPlugin Bebarmatic Settings	
ExternalItem	VendingMachineArticle	The configured item used for External registration configured in the CounterPlugin Bebarmatic Settings	
CustomerId	Guid	The default customer configured in the CounterPlugin Bebarmatic Settings	
Hours	TimeSpan	The amount of hours calculated that the user should pay for	
Price	Decimal	The price that the user should pay for	

GetVendingMachineNextTransactionID

Calling up this Webservice operation will visualise the TransactionId for the payment terminal.

Request

The GetVendingMachineCrystalReportsDatasource consists of the service context

Response

The response is a transactionId, the transactionId is from the type ulong.

GetVendingMachineSettings

Calling up this Webservice operation will visualise the settings for the vendingmachine.

Request

The GetVendingMachineCrystalReportsDatasource consists of the service context.

Response

The response is a object of VendingMachineSettings.

Name	Type	Description	Remarks
Items	Bool	Determine if there will be an items section on the vending machine	
InfoSubscriptions	Bool	Determine if there will be an subscription information section on the vending machine	
ProlongSubscriptions	Bool	Determine if there will be an subscription prolongation section on the vending machine	
ShowInItemsScreen	Bool	Determine if there will be a login button by the articles section on the vending machine	
Tickets	Bool	Determine if there will be an Ticket section on the vending machine	
IncentiveTickets	Bool	Determine if there will be an IncentiveTicket section on the vending machine	
ErrorEmailAddresses	Bool	The email address where the vendingmachine sends emails to	
PaymentMethods	VendingMachinePaymentMethods	All available paymentmethods	
MuseumCardEnterItem	VendingMachineArticle	Museumcard item that will be sold	
Bebarmatic	Bool	Determines if the Bebarmatic module should be active on the vending machine	

PrintEnvisoTicketsByOrderNumber	Bool	Determines if Enviso tickets should also be allowed to be printed in the Print ticket by order number flow	
MinimumAgeToRegister	Int	Determines the minimum age to register	
AddNewMembershipsToExistingCard	Bool	Determine if users are allowed to buy new memberships and put them on an existing card.	

VendingMachinePaymentMethods

Name	Type	Description	Remarks
Proton	Bool	Is Proton available	
Bancontact	Bool	Is Bancontact available	
Maestro	Bool	Is Maestro available	
Mastercard	Bool	Is Mastercard available	
Visa	Bool	Is Visa available	
Pin	Bool	Is Pin available	
Chip	Bool	Is Chip available	

ListVendingMachineArticles

Calling up this Webservice operation will visualise the Articles on the basis of the linked parameters.

Request

The ListVendingMachineArticles consists of the service context. The point of sales used here is the same of in the setting from the services.

Response

The response is a list of VendingMachineSubscriptions.

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the Article.	
Description	String	The description of the Subscription.	
Translations	List<VendingMachineTranslation>	The translations from RCX	
Price	Decimal	The price of the article	
Types	VendingMachineArticleTypes	Types of the article	
ExtraSettings	VendingMachineArticleExtraSettings	Extra settings of the article	
RequiredSubscriptionArticleIds	List<Guid>	All the subscription Ids to buy the article	
TicketLayout	Int	What ticket type will be printed	
CombineTickets	Bool	Print tickets together or separated	
ActiveDays	ArticleActiveDay	Active periods for the article	

VendingMachineArticleTypes

Name	Type	Description	Remarks
SubscriptionItem	Bool	Is the article an subscription item	
ConsumptionCoupon	Bool	Is the article an consumption coupon	
ChipKnipItem	Bool	Is the article an ChipKnip item	
NewMuseumCard	Bool	Is the article an museumcard	

VendingMachineArticleExtraSettings

Name	Type	Description	Remarks
AutoCardGeneration	Bool	Automaticly generate a card	

ArticleActivityDay

Name	Type	Description	Remarks
Id	GUID	The unique identifier of the ActivityDay.	
StartDate	DateTime	The start date when the article is available	
EndDate	DateTime	The end date when the article is available	

PayBebarmatic

This will notify Bebarmatic that the passed Bebarmatic ticket is paid.

Request

PayBebarmaticCriteria

Name	Type	Description	Remarks
TicketId	String	The identifier of the Bebarmatic ticket	

Response

PayBebarmaticResult

Name	Type	Description	Remarks
Paid	Bool	Indicates that the Bebarmatic ticket is paid for	

SetVendingMachineIncentivesTicketPrinted

Calling up this Webservice operation will set the incentive ticket as printed.

Request

The SetVendingMachineIncentivesTicketPrinted consists of the service context and an Guid that represents the Id of the IncentiveTicket.

Vending machine – Brief overview

Following methods are specifically used within the vending machine. These methods are mostly needed to use the correct back office pricing instead of the web pricing.

VendingMachineCarrierSwitch

Calling up this Webservice operation will link the carrier to an existing subscription.

Request

The VendingMachineCarrierSwitch consists of the service context and an VendingMachineCarrierSwitchCriteria object.

VendingMachineCarrierSwitchCriteria

Name	Type	Description	Remarks
PersonCardId	Guid	The unique identifier of the person card.	
SubscriptionId	Guid	The unique identifier of the subscription linked to the person card	
CarrierSwitchNumber	String	Card number of the carrier switch	

Response

The response is a validation result

BokenRuleName	Description	Remarks
SubscriptionNotFound	The given subscription is not found	
SubscriptionCarrierSwitchNotActive	Carrier Switch not active on given subscription	

PersonCardNotFound	The given person card is not found	
PersonCardSubscriptionNotFound	The given person card is not linked to the subscription	
CarrierSwitchCardNotFound	The carrier switch card is not found	
CarrierSwitchPersonCardLinked	There is already a person card linked to the carrier switch card	
CarrierSwitchPersonCardSubscriptionNotSaved	Could not link carrier switch card to subscription	

GetVendingMachineCrystalReportsDatasource

Calling up this Webservice operation will visualise the datasource for crystal reports on the basis of the linked parameters. This operation is used to call Stored Procedures on the SQL database and returns data Crystal Reports.

Request

The GetVendingMachineCrystalReportsDatasource consists of the service context and an VendingMachineCrystalReportDatasourceCriteria object.

VendingMachineCrystalReportDatasourceCriteria

Name	Type	Description	Remarks
Request	List<VendingMachineDictionaryObject>	Paramaters that define what Stored Procedure will be executed and the parameters for the Stored Procedure	

Response

The response is a object of VendingMachineCrystalReportDatasource.

Name	Type	Description	Remarks
Xml	String	The output of the request Stored Procedure	
XmlSchema	String	The schema of the outputted Xml	

Usage

ReportEntryTicketSubs

Key	Type	Value
ReportEntryTicketSubs	String	Empty string
SalesLineID	String	The Id of a SalesLine
SubscriptionID	String	The Id of a Subscription

ReportEntryTicket

Key	Type	Value
ReportEntryTicket	String	Empty string
SalesLineID	String	The Id of a SalesLine

ReportCultureTickets

Key	Type	Value
ReportCultureTickets	String	Empty string
CultureEventID	String	The Id of the culture event
PaymentMethodID	String	Empty string
StartReservation	String	Name of item
EndReservation	String	Name of item
OnlyPayed	Bool	False

OnlyOptions	Int	False
Printed	Bool	False
StartDate	DateTime	Min DateTime
EndDate	DateTime	Max DateTime
UniqueKey	String	Empty string
WebOnly	Int	2
CustomerID	String	Empty string
SeatCustomerID	String	Empty string

SaleslinesListSubscriptionTicketsBySalesSeries

Key	Type	Value
SalesLinesListSubscriptionTicketsBySalesSeries	String	Empty string
UniqueKey	String	A random uniqueKey
SubscriptionItems	List of SubscriptionProlongationSalesItems	All the subscription items

24 Waiterpad

Entities

DivisionConfiguration

Name	Type	Description
CashierCollection	Array Of Cashier	Contains the list of cashiers allowed for taking sales on the selected division.
PointOfSaleCollection	Array Of PointOfSale	Contains the list of sales points which are allowed on the selected division.
SalesZoneCollection	Array Of SalesZone	Contains the list of the Sales Zones available in the selected division.
DefaultCustomerId	Guid	Determines the default customer for mobile sales.
PaymentMethods	Array Of BasketPaymentMethod	Contains the list of all the possible payment methods.
MobilePaymentMethods	Array Of MobilePaymentMethod	Contains the list of the mobile payment methods which should be used for mobile sales.

Cashier

Name	Type	Description
Id	Guid	The unique Id of the cashier / waiter.
Name	String	The name of the cashier / waiter.

PointOfSale

Name	Type	Description
Id	Guid	The unique identifier of point of sales.
Name	String	The name of the point of sales.
UsesStartingBalance	Bool	Determines if a starting balance should be provided when starting a new shift.
SalesZoneId	Guid?	The unique identifier of the linked sales zone.
AllowedCashiers	IList<Guid>	A list of the unique identifiers of the cashiers which are allowed at the point of sale.

FdmActive	Bool	Defines whether the FDM (for GKS) is enabled or not.
PrintingAgentUri	String	The URI that will be used for printing.
ExpositionCustomerMandatory	Bool	Indicates if the customer is mandatory for this point of sale.
MustPrintCashReport	Bool	Indiciates if a cash report must be printed when the shift is ended.
FixedPaymentAmount	Bool	Indiciates if you have to settle the paymnet method amounts or not

SalesZone

Name	Type	Description
Id	Guid	The unique identifier of the Article.
Name	Sstring	The name of the sales zone
MultipleIntermediateTicketPrint	Bool	Indicates if it is allowed to print the intermediate ticket multiple times

PosConfiguration

Name	Type	Description
EmployeeConfiguration	EmployeeConfiguration	The configuration data for 1 employee
MenuConfiguration	PointOfSaleConfiguration	The configuration data for 1 Point of sales.

EmployeeConfiguration

Name	Type	Description
Id	Guid	The unique identifier of the employee.
Name	String	The name of the employee
IsOrderIntakeAllowed	Bool	True if the waiter is allowed to take new orders.
IsPaymentAllowed	Bool	True if the waiter is allowed to collect payments for orders.
CanAssignOrder	Bool	Helper: True if the waiter is allowed to assign orders to other employees. (can be determined with the original RCX parameters too)
EmployeeType	EmployeeType	Helper: An enumerator describing the type of the employee (can be determined with the original RCX parameters too).
CurrentEmployeeShiftConfiguration	EmployeeShiftConfiguration	The configuration of shifts for the employee.
CanChangeOrder	Bool	Helper: True if the waiter is allowed to change his own orders. (can be determined with the original RCX parameters too)
EmployeeAllowAllTableSales	Bool	Original RCX parameter (May settle all attributed tables)Sales can be added on all tables and all tables can be settled. Sales allready present from another employee CANNOT be modified!
EmployeeAllowTableChanges	Bool	Original RCX parameter (May modify and settle all attributed tables)Note: this can only be true if EmployeeAllowAllTableSales is also true. When true, sales can be added on all atrributed tables and all tables can be settled Sales allready present from another employee can also be MODIFIED/DELETED!
ParametersAllowTableChanges	Bool	Original RCX parameter (May modify and settle all attributed tables) When true, all employees can settle and modify all tables, this however requires that the employee also has EmployeeAllowAllTableSales enabled.
EmployeeAllowViewShiftInfo	Bool	Original RCX parameter (May view shift information) When true, the employee can view the information of previous and current shift.
AllowDiscounts	Bool	Is it allowed to give discount on the selected POS?

EmployeeAllowDiscounts	Bool	Is the employee allowed to give discount?
CanDeleteArticles	Bool	Is the employee allowed to delete articles?

PointOfSaleConfiguration

Name	Type	Description
Id	Guid	The id of the Point of Sale
Name	String	The name of the point of sale
PosMenuCollection	Array Of PosMenu	Contains the list of main menus avaible on this point of sale.
DefaultCustomerId	Guid	The default pos customer of the point of sale

EmployeeShiftConfiguration

Name	Type	Description
IsCashBalanceRequired	Bool	True if the employee needs to specify the amount of cash in his wallet when opening the shift.
OpeningBalance	decimal	The amount of cash that was in the wallet of the employee when the shift was started.
CashOnHand	decimal	The amount of cash currently in the wallet of the employee.
ShiftType	ShiftType	An enumerator describing the status of the current shift.
EmployeeShiftCollection	Array Of EmployeeShift	Waiter shift details like shift name, shift time, collection during shift etc. The collection will be of last 7 days' history of the employee's shifts for the employee to keep track on the CSP app.
EndBalanceCard	decimal	The amount of money paid with card in the virtual-wallet of the employee
EndBalanceCredit	decimal	The amount of money paid with credit in the virtual-wallet of the employee
EndBalanceInvoice	decimal	The amount of money paid with invoice in the virtual-wallet of the employee

EmployeeShift

Name	Type	Description
ShiftStartTimestamp	dateTime	Shift start timestamp
ShiftCloseTimestamp	dateTime	Shift end timestamp
ShiftStartingBalance	decimal	Starting balance of the shift
ShiftClosingBalance	decimal	Closing balance of the shift
ShiftCashCollection	decimal	Total amount (cash) collected by waiter during the shift
Description	String	Shift summary, if any, entered by the waiter
ShiftClosingBalanceCard	decimal	Total amount (card) collected by waiter during the shift
ShiftClosingBalanceCredit	decimal	Total amount (credit) collected by waiter during the shift
ShiftClosingBalanceInvoice	decimal	Total amount (invoice) collected by waiter during the shift

PosMenu

A pos menu is a collection of buttons that are on the menu.

Name	Type	Description
Id	Guid	The unique identifier of the pos menu.
Name	String	The name of the pos menu
MenuButtons	Array Of MenuButton	The list of MenuButtons in this menu.
SubMenus	Array Of PosMenu	The list of child menu's in this menu. This will only be availbale in the main menu, because menu's can be recursive.

MenuButton

MenuButton is a base clase inherited by MenuItem and MenuLink.
A menu button is either an item the customer can order, or a link to a submenu.

Name	Type	Description
Id	Guid	The unique Id of the button within the menu
Name	String	The text that appears on the menu for this item or submenu.
X	Int	X coordinate of the button inside the PosMenu
Y	Int	Y coordinate of the button inside the PosMenu
Color	Color	Backgroundcolor of the button
IsDigitalWalletItem	bool	Indicates whether the item is a digital wallet item or not

MenuItem

Name	Type	Description
Id	Guid	The unique Id of the button within the menu
Name	String	The text that appears on the menu for this item or submenu.
X	Int	X coordinate of the button inside the PosMenu
Y	Int	Y coordinate of the button inside the PosMenu
Color	Color	Backgroundcolor of the button
IsDigitalWalletItem	bool	Indicates whether the item is a digital wallet item or not
ArticleGroup	String	The article group which the menu item belongs to.
ArticleModifierCollection	Array Of ArticleModifier	A collection of modifiers which can be applied to an article.
UnitPrice	decimal	The unit price of the item.
ArticleId	Guid	The unique id of the article
ArticleDescription	String	The description of the article
CustomerMandatory	MandatoryType	Defines whether or not the customer is mandatory for this article
AllowCustomPrice	bool	Indiciates if a custom price is allowed
CombineSales	bool	Indiciates if the sales should be combined

ExpositionMenuItem

Name	Type	Description
Id	Guid	The unique Id of the button within the menu

Name	string	The text that appears on the menu for this item or submenu.
X	Int	X coordinate of the button inside the PosMenu
Y	Int	Y coordinate of the button inside the PosMenu
Color	Color	Backgroundcolor of the button
IsDigitalWalletItem	bool	Indicates whether the item is a digital wallet item or not
ExpositionId	Guid	The unique id of the exposition.

ActivityMenuItem

Name	Type	Description
Id	Guid	The unique Id of the button within the menu
Name	string	The text that appears on the menu for this item or submenu.
X	Int	X coordinate of the button inside the PosMenu
Y	Int	Y coordinate of the button inside the PosMenu
Color	Color	Backgroundcolor of the button
IsDigitalWalletItem	bool	Indicates whether the item is a digital wallet item or not
ActivityId	Guid	The unique id of the activity.

CultureEventMenuItem

Name	Type	Description
Id	Guid	The unique Id of the button within the menu
Name	string	The text that appears on the menu for this item or submenu.
X	Int	X coordinate of the button inside the PosMenu
Y	Int	Y coordinate of the button inside the PosMenu
Color	Color	Backgroundcolor of the button
IsDigitalWalletItem	bool	Indicates whether the item is a digital wallet item or not
CultureEventId	Guid	The unique id of the culture event.

CultureSubscriptionMenuItem

Name	Type	Description
Id	Guid	The unique Id of the button within the menu
Name	string	The text that appears on the menu for this item or submenu.
X	Int	X coordinate of the button inside the PosMenu
Y	Int	Y coordinate of the button inside the PosMenu
Color	Color	Backgroundcolor of the button
IsDigitalWalletItem	bool	Indicates whether the item is a digital wallet item or not
CultureSubscriptionId	Guid	The unique id of the culture subscription.

ReservationViewMenuItem

Name	Type	Description
Id	Guid	The unique Id of the button within the menu
Name	string	The text that appears on the menu for this item or submenu.
X	Int	X coordinate of the button inside the PosMenu
Y	Int	Y coordinate of the button inside the PosMenu
Color	Color	Backgroundcolor of the button
IsDigitalWalletItem	bool	Indicates whether the item is a digital wallet item or not
ReservationViewId	Guid	The unique id of the reservation view.

RentalMenuItem

Name	Type	Description
Id	Guid	The unique Id of the button within the menu
Name	string	The text that appears on the menu for this item or submenu.
X	Int	X coordinate of the button inside the PosMenu
Y	Int	Y coordinate of the button inside the PosMenu
Color	Color	Backgroundcolor of the button
IsDigitalWalletItem	bool	Indicates whether the item is a digital wallet item or not
ItemId	Guid	The unique id of the item to rent.

FitnessActivityMenuItem

Name	Type	Description
Id	Guid	The unique Id of the button within the menu
Name	string	The text that appears on the menu for this item or submenu.
X	Int	X coordinate of the button inside the PosMenu
Y	Int	Y coordinate of the button inside the PosMenu
Color	Color	Backgroundcolor of the button
IsDigitalWalletItem	bool	Indicates whether the item is a digital wallet item or not
FitnessActivityId	Guid	The unique id of the fitness activity.

MenuLink

Name	Type	Description
Id	Guid	The unique Id of the button within the menu
Name	string	The text that appears on the menu for this item or submenu.
X	Int	X coordinate of the button inside the PosMenu
Y	Int	Y coordinate of the button inside the PosMenu
Color	Color	Backgroundcolor of the button
IsDigitalWalletItem	bool	Indicates whether the item is a digital wallet item or not
MenuId	Guid	The unique identifier of the sub menu this buttons links to. The submenu should be found in the SubMenus collection of the main PosMenu.

ArticleModifier

Article modifiers are a collection of option values from which the user can choose one.
Option values are mutually exclusive.
For example :
For meat the ArticleModifier will indicate an option "Cooking". Possible ArticleModifierValues are : rare, medium, well cooked.
There could be a second option for sauces : Pepper sauce, Mushroom sauce, …

Name	Type	Description
Id	Guid	The unique identifier of the ArticleModifier.
Name	String	The description of the article modifier.
ArticleModifierValueCollection	Array Of ArticleModifierValue	A collection of possible values from which the user can choose one.

ArticleModifierValue

Modifier values are a list of options from which the user can choose one value. The choosen value is an option for a menu-item. These options are mostley cooking, veageatables, sauces, …

Name	Type	Description
Id	Guid	The unique identifier of the Article modifier value.
Name	String	The text description of the modifier value.
ArticleId	Guid (optional)	Sometimes a modifier value acts as an extra article added to the list of orders. In case of the modifier "cooking", this will be empty and a text line will be added to the order with the description of the choosen coocking. In case of "Sauce" an extra item with the choosen sauce will be added, because adding a sauce to a dish will increment the price of the dish. Note : Adding an extra line to the order is done by the API, the webservice consumer doesn't have to do this.
UnitPrice	Decimal (optional)	When a modifier value is linked to an ArticleId, there can be an extra cost for the option. When there is an extra cost, the field UnitPrice indicates the amount of the extra cost.

SalesZoneConfiguration

Name	Type	Description
Id	Guid	The unique identifier of the sales zone.
Name	string	The name of the sales zone.
TableCollection	Array Of Table	The collection of tables present in the sales zone.

Table

Name	Type	Description
Id	Guid	The unique identifier of the table.
Name	string	The name of the table
TableType	TableType	The kind of table : Ordinal, Virtual, …
SalesZoneId	Guid	The id of the sales zone which the table belongs to.
EmployeeId	Guid	The id of the employee which the table is assigned to.
TableStatus	TableStatus	The current status of the table

TableSaleCollection	Array Of TableSale	The collection of order items for the table
PrintedTicket	Bool	True if the ticket has been printed
Location	Location	Location data for the table
PageName	string	Name for the page the table is on
PageUID	Guid	Unique id for the page the table is on

TableSale

A TableSale is one orderline. This object inherits from BasketItem.

Name	Type	Description
DivisionId	Guid	The division which this order belongs to. This should not be specified by the webservice consumer, it is filled in by the webservice logic when the order has been sent to the POS system.
Id	Guid	The unique id of the order line. This should not be specified by the webservice consumer, it is filled in by the webservice logic when the order has been sent to the POS system.
Quantity	Int	The quantity of a specific item ordered by a user. When the same item has been ordered more then once, these items should be combined in one order line (TableSale) because the customer could get a discount based on the quantity ordered. There is one exception : When an item has an article modifier, these order lines cannot be combined, even if the modifier values are the same. The API will check this when writing order lines to the database, and if this constraint is violated, it will split this orderliness in multiple orderliness.
RuleNamesToIgnore	Array Of string	The list of validation rules that should be ignored for table sales. Not used at his moment.
UnitPrice	Decimal	The unit price of the order line. This can be set by the webservice consumer, but will be overwritten by the API once the order has been sent to the POS system.
AdvancementPrice	Decimal	Not used by table sales.
Sequence	Int	Each order line has a sequence number which should be unique within the order. As a rule, the first sequence starts at 10.000 and should be incremented by 10.000 for each consecutive order line.
CreateTimestamp	DateTime	The timestamp on which the orderline was created.
TableSaleType	TableSaleType	Enum which indicates if the orderline contains text or an item.
ArticleId	Guid	If the TableSaleType is an item, the ArticleId should contain the Id of the ordered item. This should be empty for text lines.
Description	String	The description of the ordered item when TableSaleType is "Item", or free text when the TableSaleType is "text".
ExtraDescription	String	A custom value which can be specified by the webservice consumer to provide additional information. Ex. "customer is allergic to cucumber".
Modifiers	Array Of ChosenArticleModifier	Collection of Chosen article modifiers.
EmployeeId	Guid	The id of the employee who registered this order line.
OrganisedVisitId	Guid	The id of the linked organised visit (if applicable).
CultureEventReservationId	Guid	The id of the linked ticketing reservation (if applicable).
ReservationId	Guid	The id of the linked place reservation (if applicable).
RentalHeaderId	Guid	The id of the linked rental reservation (if applicable).
RentalLineId	Guid	The id of the linked rental detail line (if applicable).

ActivityEntryId	Guid	The id of the linked activity reservation (if applicable).
FitnessActivityDayReservationId	Guid	The id of the linked fitness reservation (if applicable).
Amount	Decimal	The price amount of the order line.
CustomerID	Guid?	The id of the customer which is linked to the order line.

ChosenArticleModifier

A chosen article modifier contains the information about selected options for an ordered item.

Name	Type	Description
ArticleModifierDescription	string	The description of the article modifier.
ArticleModifierValueDescription	string	The description of the article modifier value.
ArticleId	Guid	The item associated to the article modifier value.
Amount	decimal	The total amount for the selected option.

MoveToTableAction

A chosen article modifier contains the information about selected options for an ordered item.

Name	Type	Description
TableSaleId	Guid	The id of the tablesale you want to move.
TableId	Guid	The id of the table you want to move the sale to.
Quantity	Decimal	The amount to move to the other table.

TicketMessage

Describes the status of a table

Name	Type	Description
Id	Guid	Id of the message
ShortName	String	A short name for the message
Message	String	The message that will be printed

MobilePaymentMethod

Name	Type	Description
Id	Guid	Id of the BasketPaymentMethod
Type	MobilePaymentType	The type of the mobile payment: • Cash • CreditCard • Credit • Invoice

PrintDatum

Name	Type	Description	Remarks
PrinterType	PrinterTypes	The printer type	
PrinterNr	Int?	The ticket printer number	

Data	Byte[]	The report/ticket as a byte array.	

TicketPrinter

Name	Type	Description	Remarks
PrinterNr	int	The ticket printer number	
PrinterType	TicketPrinterType	The type of the ticket printer	

Enumerations

EmployeeType

EmployeeType describes the type of employee.

Name	Description
Waiter	The employee is a regular waiter.
HeadWaiter	The employee is a waiter with added privileges : This waiter can assign orders to other tables and can alter orders of tables assigned to other employees.
Manager	Not used
Cashier	Not used

ShiftType

Describes the shift status for an employee

Name	Description
NoShift	There is no open shift for this employee
ExistingShift	There is an existing,open, shift for this employee.
NewShift	A new shift has been openened for this employee.

TableType

Describes the kind of table

Name	Description
Ordinal	A regular table
Grouped	The table is a group of other tables
Virtual	The table is virtual, it isn't mapped to an existing table

TableStatus

Describes the status of a table

Name	Description
Available	The table is available to customers
Occupied	The table is already occupied by one or more customers

TableSaleType

Describes the kind of orderline

Name	Description

Item	The unique identifier of the Article.
Text	

TicketPrinterType

Name	Description
TM_88	Refers to the TM_88III EPSON printer
TM_U220	Refers to the TM_U220 EPSON printer
TM_P60	Refers to the TM_P60II EPSON printer
TM_P80	Refers to the TM_P80 EPSON printer

Methods

An overview of all methods which can be used within the waiterpad implementation

ConfigureDevice

The method ConfigureDevice is used to get basic information about the available SalesZones, PointOfSales and employees.

Parameters

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MAC Address ID of the device to be registered with the web service. This MAC ID will be used by the web service for registering the device for subsequent communication.
EmployeeId	Guid	Not needed in this webservice call.

Response

Name	Type	Description
Data	DivisionConfiguration	The DivisionConfiguration data contains the brief description of the configuration data available for the employee.
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an error code when there was an error processing the request.

CreateVirtualTable

CreateVirtualTableRequest

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the Point Of Sales on which the user is currently logged in.
EmployeeId	Guid	The Id of the employee which is logged in in the application.
SalesZ	Guid	The Id of the SalesZone to which this table belongs.

oneId		
Description	String	The description that will be given to the new table.

CreateVirtualTableResponse

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an error code when there was an error processing the request.
Table	Table	The new virtual table that was created.

Following error codes are possible:

Error code	Description
INVALID_EMPLOYEE	The provided EmployeeId was not found in ReCreateX back office
INVALID_SALES_ZONE_ID	The provided SalesZoneId was not found in ReCreateX back office
MISSING_DESCRIPTION	The description for the new virtual table was not provided

DeleteTable

DeleteTableRequest

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the Point Of Sales on which the user is currently logged in.
EmployeeId	Guid	The Id of the employee which is logged in in the application.
SalesZoneId	Guid	The Id of the SalesZone to which this table belongs.
Table	Table	The table you want to delete.

DeleteTableResponse

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an error code when there was an error processing the request.

Following error codes are possible:

Error code	Description
INVALID_EMPLOYEE	The provided EmployeeId was not found in ReCreateX back office
INVALID_TABLE_ID	The provided TableId was not found in ReCreateX back office
ORDINAL_TABLES_CANNOT_BE_DELETED	A normal table can not be deleted, only fictional (=virtual) tables can be deleted
TABLES_WITH_ORDERS_CANNOT_BE_DELETED	A table with open order(s) can not be deleted

EMPLOYEE_TABLE_CHANGE_NOT_ALLOWED	The provided employee does not have the rights to change/split tables

GetPointOfSaleConfiguration

Request

The request consists of the service context and an GetPointOfSaleConfigurationRequest object.

GetPointOfSaleConfigurationRequest

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the Point Of Sales on which the user is currently logged in.
EmployeeId	Guid	The Id of the employee which is logged in in the application.
SalesZoneId	Guid	The Id of the SalesZone to which this table belongs.

Response

The response contains the configuration details of the point of sale:

GetPointOfSaleConfigurationResponse

Name	Type	Description
Result	boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an errorcode if an error occurred.
PointOfSaleConfiguration	PointOfSaleConfiguration	Returns the PointOfSaleConfiguration if no error occured.

PointOfSaleConfiguration

Name	Type	Description	Remarks
Id	Guid	The unique identifier of the point of sale	
Name	String	The name of the point of sale	
PosMenuCollection	IList<PosMenu>	The list of menus which are available in the point of sale	

GetSalesZoneOverview

Request

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the point of sales for which the overview should be obtained.
EmployeeId	Guid	The Id of the current employee.
SalesZoneId	Guid	The Id of the sales zone for which the overview should be obtained.

PageId	Guid?	The Id of the sales zone page for which the overview should be obtained. • Pass no value to retrieve the full sales zone overview • Pass empty Guid to retrieve the overview of all virtual tables • Pass a Page Guid to retrieve the overview of a certain page

Response

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an error code when there was an error processing the request.
TableOverviewCollection	IList<TableOverview>	The list of TableOverview objects which contain the high-level table overview.

TableOverview

The TableOverview object contains all high-level (dynamic) table information.

Name	Type	Description
Id	Guid	The unique identifier of the table.
TableType	TableType	The kind of table : Ordinal, Virtual, …
EmployeeId	Guid	The id of the employee which the table is assigned to.
SplitCode	String	The code which is given to the split table.
MainTableId	Guid	The id of the table to which this table is linked (split tables)
TicketPrinted	Bool	True if the ticket has been printed
Amount	Decimal	The total price that is linked to the open sale (if any)
TableNo	int	The number of the table
Description	string	The description of the table

Following error codes are possible:

Error code	Description
INVALID_SALESZONE	The provided SalesZoneId was not found in ReCreateX back office

GetSalesZoneStatus

Request

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the point of sales for which the status should be obtained.
EmployeeId	Guid	The Id of the current employee.
SalesZoneId	Guid	The Id of the sales zone for which the status should be obtained.
TableId	Guid?	The id of the sales table for which the status should be obtained. If empty, all tables for the sales zone will be obtained.

Response

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an error code when there was an error processing the request.
Data	SalesZoneConfiguration	The configuration of the requested Sales Zone.

Following error codes are possible:

Error code	Description
INVALID_SALESZONE	The provided SalesZoneId was not found in ReCreateX back office

GetSalesZoneTables

Request

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the point of sales for which the status should be obtained.
EmployeeId	Guid	The Id of the current employee.
SalesZoneId	Guid	The Id of the sales zone for which the status should be obtained.
TableId	Guid?	The Id of a table

Response

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an error code when there was an error processing the request.
TableBaseCollection	IList<TableBase>	The list of TableBase objects which contain the static table information.

TableBase

The TableBase object contains all static table information.
This information should only be retrieved once since it does not change a lot.

Name	Type	Description
Id	Guid	The unique identifier of the table.
Name	String	The name or number of the table.
Description	String	The description of the table.
SalesZoneId	Guid	The id of the sales zone which the table belongs to.
Location	Location	Location data for the table (physical location).
PageUID	Guid	Unique id for the page the table is on.
PageName	String	Name for the page the table is on.

Following error codes are possible:

Error code	Description
INVALID_SALESZONE	The provided SalesZoneId was not found in ReCreateX back office

GetTableSalesWithoutTable

Request

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the point of sales for which the status should be obtained.
EmployeeId	Guid	The Id of the current employee.
SalesZoneId	Guid	*Leave this parameter empty.*

Response

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an error code when there was an error processing the request.
TableSales	Array Of TableSale	The sales you are currently doing.

GetTicketMessages

Request

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the Point Of Sales on which the user is currently logged in.
EmployeeId	Guid	The Id of the employee which is logged in in the application.
SalesZoneId	Guid	The Id of the SalesZone to which this table belongs.

Response

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an errorcode if an error occurred.
TicketMessages	Array of TicketMessage	Returns an array of ticketmessages

LoadGuiConfiguration

LoadGuiConfiguration allows for loading custom parameters per WebShop. These are device independant, they are the same for all users who use the same webshop.

Request

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.

Response

The GuiConfigurationValidationResult inherits from ValidationResult where the base validation result indicates whether the request failed or successed.

Name	Type	Description
GuiConfiguration	String	The GuiConfiguration is a string which contains XML text. API users can consume this XML and add their own nodes. *Note : Only XML Elements known by the application should be modified, added or removed. Xml Elements not known by the API Consumer should remain in the original GuiConfiguration.*

LoginCashier

The LoginCashier method is used to validate the EmployeePIN, which authorizes an employee to use the application.

Request

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
Employee Pin	Int	The pin code of the employee, needed for validating the login of the employee.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the Point Of Sale to which the user logs on. This Id can be obtained from the PointOfSaleCollection which is in the DivisionConfiguration.
EmployeeId	Guid	The Id of the Employee who tries to login.
SalesZone Id	Guid	The Id of the SalesZone on which the employee logs in. This can be obtained from the SalesZoneCollection which is in the DivisionConfiguration.
RFID	String	The employee RFID, for example from a wristband. When this is provided, no EmployeePin/EmployeeId is required.
CardNo	String	The card number of the employee. When this is provided, no EmployeePin/EmployeeId is required.
IncludeConfiguration	Bool?	If set true, the Employee- and PointOfSaleConfiguration is included

Response

Name	Type	Description
Data	PosConfiguration	Contains the entire configuration for the employee and the selected point of sales.
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an error code when there was an error processing the request.

Following error codes are possible:

Error code	Description
NOT_ALL_PARAMETERS_PROVIDED	One ore more mandatory parameters were not provided (EmployeeId / PointOfSaleId)
LOGIN_FAILED	Login failed for the provided EmployeeId & EmployeePin
FDM_ERROR	Error during FDM module

LogoutCashier

Request

Name	Type	Description	Comments
Context	ServiceContext	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.	
CloseShift	Boolean	True if the current shift for the selected employee should be closed. If the shift should be close on logout, then a closingbalance can be specified.	
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.	
PointOfSaleId	Guid	The Point of sale from which the user want to log out.	
EmployeeId	Guid	The id of the employee who wants to log out.	
SalesZoneId	Guid	The Id of the Sales Zone from which the employee wants to log out.	
ClosingBalance	Decimal	The amount the employee currently has in its wallet.	Only to be used for CASH payment
ShiftReceipts	IList<ShiftReceipt>	List of remaining balances for multiple payment methods	When filled in, the ClosingBalance parameter is overruled and will not be taken into account

ShiftReceipt

Name	Type	Description
PaymentMethodId	Guid	The payment method for which a remaning balance needs to be entered
Amount	Decimal	The remaining balance/amount

Response

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an error code when there was an error processing the request.

Following error codes are possible:

Error code	Description
NOT_ALL_PARAMETERS_PROVIDED	One ore more mandatory parameters were not provided (EmployeeId)
INVALID_EMPLOYEE	The provided EmployeeId was not found in ReCreateX back office
CASH_PAYMENT_METHOD_MISSING	The default payment method for cash payments is not configured correctly in ReCreateX back office
NO_OPEN_SHIFT	There is not open shift for the provided employee and point of sale
BLOCKED_OPEN_TABLES	There are still open tables for the provided EmployeeId

MoveTableSalesToTable

Request

Name	Type	Description
Context	ServiceContext	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the Point Of Sales on which the user is currently logged in.
EmployeeId	Guid	The Id of the employee which is logged in in the application.
SalesZoneId	Guid	The Id of the SalesZone to which this table belongs.
Moves	Array of MoveToTableAction	A list of all the move actions

Response

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an errorcode if an error occurred.

Following error codes are possible:

Error code	Description
NOT_ALL_PARAMETERS_PROVIDED	One ore more mandatory parameters were not provided (EmployeeId / Moves / PointOfSaleId)
INVALID_EMPLOYEE	The provided EmployeeId was not found in ReCreateX back office
INVALID_POS	The provided PointOfSaleId was not found in ReCreateX back office
NO_OPEN_SHIFT	There is not open shift for the provided employee and point of sale
INVALID_TABLE_SALE	The provided TableSaleId was not found in ReCreateX back office
INVALID_QUANTITY	The provided Quantity is not correct or exceeds the quantity of the original sale
CANNOT_MOVE_TO_SAME_TABLE	The provided TableId can not be the same as the original TableId
INVALID_TABLE	The provided TableId was not found in ReCreateX back office
RIGHTS_CHANGE_OTHER_EMPLOYEE_ORDERS	The provided employee does not have the rights to change orders
RIGHTS_CHANGE_OTHER_EMPLOYEE_TABLE	The table was started by another employee and the provided employee does not have the rights to change tables

PointOfSaleResetZeroLines

Request

The request consists of the service context and an PointOfSaleResetZeroLinesRequest object.

PointOfSaleResetZeroLinesRequest

Name	Type	Description
PointOfSaleId	Guid	Unique identifier of the point of sale

Response

The response contains a bool returning whether or not the POS sales lines were reset:

PointOfSaleResetZeroLinesResponse

Name	Type	Description
Result	boolean	Returns true if the service request could be processed, otherwise false.

SalesZoneTableResetZeroLines

Request

The request consists of the service context and an SalesZoneTableResetZeroLinesRequest object.

SalesZoneTableResetZeroLinesRequest

Name	Type	Description
SalesZoneTableId	Guid	The unique identifier of the sales zone table

Response

The response contains a bool returning whether or not the POS sales lines were reset:

SalesZoneTableResetZeroLinesResponse

Name	Type	Description
Result	boolean	Returns true if the service request could be processed, otherwise false.

SaveGuiConfiguration

SaveGuiConfiguration allows for saving custom parameters per WebShop. These are device independant, they are the same for all users who use the same webshop.

Parameters

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
GuiConfiguration	String	The GuiConfiguration is a string which contains XML text. API users can consume this XML and add their own nodes. *Note : Only XML Elements known by the application should be modified, added or removed. Xml Elements not known by the API Consumer should remain in the original GuiConfiguration.*

Response

The GuiConfigurationValidationResult inherits from ValidationResult where the base validation result indicates whether the request failed or successed.

Name	Type	Description
GuiConfiguration	String	The GuiConfiguration is a string which contains XML text. API users can consume this XML and add their own nodes. *Note : Only XML Elements known by the application should be modified, added or removed. Xml Elements not known by the API Consumer should remain in the original GuiConfiguration.*

SendPOSOrder

Request

Name	Type	Description

Context	ServiceContext	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the Point Of Sales on which the user is currently logged in.
TableSaleCollection	Array Of TableSale	An order is a collection of order lines op the TableSale type **.**
EmployeeId	Guid	The Id of the employee which is logged in in the application.
SalesZoneId	Guid	The Id of the SalesZone to which this table belongs.
TableId	Guid?	The Id of the order to which this order belongs.

Response

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.
TableSaleCollection	Array Of TableSale	Returns the collection of order lines for the requested table.Note that the returned list of order lines can deffer from the list specified in the parameters. When adding new TableSales to the list, the Id's of the TableSale should be empty, because the are assigned by the webservices. So the Id's will only be known in the request. Other case in which the returned list can deffer from the original list is when the more then 1 employee is assigned to the same table. The result list will contains sales from all employees for the requested table. If the table order list is cached on the device using the webserices, then after successfully using this method, the returned list should be used in further requests.
ErrorCode	String	Returns an error code when there was an error processing the request.
TableId	Guid	The Id of the table for which the order has been sent.

Following error codes are possible:

Error code	Description
NOT_ALL_PARAMETERS_PROVIDED	One ore more mandatory parameters were not provided (EmployeeId / TableSaleCollection / PointOfSaleId)
INVALID_EMPLOYEE	The provided EmployeeId was not found in ReCreateX back office
INVALID_TABLE	The provided TableId was not found in ReCreateX back office
INVALID_SALESZONE_FOR_TABLE	The provided SalesZoneId was not found in ReCreateX back office
INVALID_POS	The provided PointOfSaleId was not found in ReCreateX back office
NO_OPEN_SHIFT	There is not open shift for the provided employee and point of sale
RIGHTS_CHANGE_OTHER_EMPLOYEE_TABLE	The table was started by another employee and the provided employee does not have the rights to change tables
INVALID_CUSTOMER	The provided CustomerId was not found in ReCreateX back office
MULTIPLE_CUSTOMERS_ON_SAME_TABLE	Multiple CustomerIds were provided for the same table where all sales should be on the same (or no) customer
INVALID_SALE_ID	A new sale should always have an empty Guid for the Id parameter
INVALID_ARTICLEID	The provided ArticleId was not found in ReCreateX back office
ONLY_RECREATEX	The provided ArticleId may only be used in ReCreateX back office (depending on the type of article: MuseumCard / Fitness / Loyalty / LessonGroup / Subsidy / BlackList / AskPriceGroup types are not supported)

INVALID_QUANTITY	The Quantity parameter was not provided correctly
NEGATIVE_SALES_NOT_ALLOWED	The provided employee does not have the rights to allow negative sales
INVALID_MODIFIER_VALUE	The provided ArticleModifierValueDescription was not found in ReCreateX back office
RIGHTS_CHANGE_OTHER_EMPLOYEE_ORDERS	The provided employee does not have the rights to change orders
CANNOT_CHANGE_ARTICLEID	The provided ArticleId does not correspond to the initial order
CANNOT_CHANGE_PROLONG_SUBSCRIPTION	It is not possible to change a subscription prolongation
CANNOT_CHANGE_FDM_ACTIVE	It is not possible to change an order when the FDM module is activated on the provided PointOfSaleId
VAT_PERCENTAGE_NOT_ZERO_ON_ARRANGEMENT	The VAT percentage should be configured as 0 on arrangement articles
MAX_SALES_QUANTITY_EXCEEDED	The desired quantity exceeds the maximum allowed quantity
ITEM_NOT_IN_MENU	The provided ArticleId is not on the menu and the provided PointOfSaleId does not allow to sell articles that are not on the menu
AUTOCARD_PERIPHERALSTRING_MUST_BE_SPECIFIED	No default peripheral is configured in ReCreateX back office
DONATION_ITEM_MUST_BE_SPECIFIED	No donation item is configured in ReCreateX back office
SUBSIDY_SETTINGS_MUST_BE_SPECIFIED	Not all subsidy settings are configured correctly in ReCreateX back office
INACTIVE_ITEM	The article is not active and thus can not be sold
STOCK_ITEM_MISCONFIGURED	The unit of measure is not correctly configured for the provided ArticleId
ITEM_LEDGER_NEGATIVE	The stock is dropping below zero and the provided PointOfSaleId is configured to alert this
ITEM_LEDGER_UNDER_MINIMUM	The stock is dropping below the minimum and the provided PointOfSaleId is configured to alert this

SplitTable

SplitTableRequest

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the Point Of Sales on which the user is currently logged in.
EmployeeId	Guid	The Id of the employee which is logged in in the application.
SalesZoneId	Guid	The Id of the SalesZone to which this table belongs.
Table	Table	The table that needs to be split
Description	String	The description that will be given to the new table

SplitTableResponse

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.

ErrorCode	String	Returns an error code when there was an error processing the request.
Table	Table	The new virtual table that was created

Following error codes are possible:

Error code	Description
INVALID_EMPLOYEE	The provided EmployeeId was not found in ReCreateX back office
INVALID_TABLE_ID	The provided TableId was not found in ReCreateX back office
FICTIONAL_TABLE_CANNOT_BE_SPLIT	A fictional (or virtual) table can not be split
INVALID_POS	The provided PointOfSaleId was not found in ReCreateX back office
EMPLOYEE_TABLE_CHANGE_NOT_ALLOWED	The provided employee does not have the rights to change/split tables

StartNewShift

Request

Name	Type	Description
Context	Service Context	The service context contains the required configuration elements needed by the webservices to connect to the correct database. The ShopId is very important and should be known by the webservice consumer.
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing.
PointOfSaleId	Guid	The Id of the point of sale on which a new shift should be started.
Starting Balance	Decimal	The amount of cash an employee has in its wallet when the shift is about to be started.
EmployeeId	Guid	The Id of the employee who starts a new shift.
SalesZoneId	Guid	The Id of the sales zone on which a new shift is started.

Response

Name	Type	Description
Result	Boolean	Returns true if the service request could be processed, otherwise false.
ErrorCode	String	Returns an error code when there was an error processing the request.
SalesShiftId	Guid	The ID of the new sales shift.

Following error codes are possible:

Error code	Description
NOT_ALL_PARAMETERS_PROVIDED	One ore more mandatory parameters were not provided (EmployeeId)
INVALID_POS	The provided PointOfSaleId was not found in ReCreateX back office
POS_REQUIRES_STARTING_BALANCE	The PointOfSaleId requires a starting balance which was not provided
POS_DOES_NOT_USE_STARTING_BALANCE	The PointOfSaleId does not use starting balances while it was provided
INVALID_EMPLOYEE	The provided EmployeeId was not found in ReCreateX back office
OPEN_SHIFT	There is an open shift for the provided employee and point of sale

Print methods

An overview of all print methods which can be used within the waiterpad implementation

GetCashReport

The cash report method can be used to generate a cash report of a SalesShift.

Request

The request contain a ServiceContext object and a GetCashReportCriteria criteria object.

GetCashReportCriteria

Name	Type	Description	Remarks
SalesShiftId	Guid	The id of the SalesShift	
PrinterType	TicketPrinterType?	The ticket printer type	Optional
Printer	TicketPrinter	The ticket printer	Optional, when filled this has priority on the PrinterType parameter

Response

The response contains a list of PrintDatum objects.

GetIntermediateSalesTicketPrintData

The method can be used to generate an intermediate sales ticket for a certain table.

Request

The request contain a ServiceContext object and a GetIntermediateSalesTicketCriteria criteria object.

GetIntermediateSalesTicketCriteria

Name	Type	Description	Remarks
SalesZoneTableId	Guid	The id of the SalesZoneTable	
EmployeeId	Guid	The id of the Employee	
PrinterType	TicketPrinterType?	The ticket printer type	Optional
PointOfSaleId	Guid	The id of the PointOfSale	
MacId	String	The MacId of the device, used for validating the number of concurrent users in terms of licensing	
Printer	TicketPrinter	The ticket printer	Optional, when filled this has priority on the PrinterType parameter

Response

The response contains a list of PrintDatum objects.

GetKitchenBarTicketPrintData

The method can be used to generate a kitchen or bar ticket for a certain table or sale.

Request

The request contain a ServiceContext object and a GetKitchenBarTicketCriteria criteria object.

GetKitchenBarTicketCriteria

The GetKitchenBarTicketCriteria object can be specified for a certain table or for a certain sale.
Following (derived) objects can be used for this.

GetKitchenBarTicketCriteriaForTable

Name	Type	Description	Remarks
SalesZoneTableId	Guid	The id of the SalesZoneTable	

EmployeeId	Guid	The id of the Employee	
PrinterType	TicketPrinterType?	The ticket printer type	Optional
PointOfSaleId	Guid	The id of the PointOfSale	
KitchenPrinters	List<TicketPrinter>	The ticket printers to be used for the kitchen	Optional, when filled this has priority on the PrinterType parameter
Barprinters	List<TicketPrinter>	The ticket printers to be used for the bar	Optional, when filled this has priority on the PrinterType parameter

GetKitchenBarTicketCriteriaWithoutTable

Name	Type	Description	Remarks
SalesSeriesId	Guid	The id of the SalesSeries	
LastHeaderOnly	Bool		
PrinterType	TicketPrinterType?	The ticket printer type	Optional
KitchenPrinters	List<TicketPrinter>	The ticket printers to be used for the kitchen	Optional, when filled this has priority on the PrinterType parameter
Barprinters	List<TicketPrinter>	The ticket printers to be used for the bar	Optional, when filled this has priority on the PrinterType parameter

Response

The response contains a list of PrintDatum objects.

GetKitchenMessageTicketPrintData

This method can be used to generate a kitchen message report for a certain table.

Request

The request contain a ServiceContext object and a GetInterTicketMessageCriteria object.

GetInterTicketMessageCriteria

Name	Type	Description	Remarks
PointOfSaleId	Guid	The id of the PointOfSale	
SalesZoneTableId	Guid	The id of the SalesZoneTable	
MessageId	Guid	The id of the Message	
EmployeeId	Guid	The id of the Employee	
PrinterType	TicketPrinterType?	The ticket printer type	Optional
Printers	List<TicketPrinter>	The list of ticket printers to be used	Optional, when filled this has priority on the PrinterType parameter

Response

The response contains a list of PrintDatum objects.

GetPosReport

The POS report method can be used to generate a POS report for a certain point of sale.

Request

The request contain a ServiceContext object and a GetPosReportCriteria criteria object.

GetPosReportCriteria

Name	Type	Description	Remarks
PointOfSaleId	Guid	The id of the PointOfSale	
IsFinancialReport	Bool		
IsFinalReport	Bool		
PrinterType	TicketPrinterType?	The ticket printer type	Optional

Printer	TicketPrinter	The ticket printer	Optional, when filled this has priority on the PrinterType parameter

Response

The response contains a list of PrintDatum objects.

GetSalesHeaderAccessTicketsPrintData

This method can be used to generate the access tickets for a certain sale.

Request

The request contain a ServiceContext object and a GetSalesHeaderAccessTicketsCriteria criteria object.

GetSalesHeaderReceiptCriteria

Name	Type	Description	Remarks
SalesSeriesId	Guid	The id of the SalesSeries	Mandatory
PrinterType	TicketPrinterType?	The ticket printer type	Optional
PaperSizeWidth	Decimal	Determines the size of the paper	Mandatory
Printer	TicketPrinter	The ticket printer	Optional, when filled this has priority on the PrinterType parameter

Response

The response contains a list of PrintDatum objects.

GetSalesHeaderReceiptPrintData

This method can be used to generate the receipt for a certain sale.

Request

The request contain a ServiceContext object and a GetSalesHeaderReceiptCriteria criteria object.

GetSalesHeaderReceiptCriteria

Name	Type	Description	Remarks
SalesSeriesId	Guid	The id of the SalesSeries	Mandatory
PrinterType	TicketPrinterType?	The ticket printer type	Optional
PaperSizeWidth	Decimal	Determines the size of the paper	Mandatory
PrintAccessTickets	Bool	Determines if the access tickets should be printed or not	Optional
Printer	TicketPrinter	The ticket printer	Optional, when filled this has priority on the PrinterType parameter

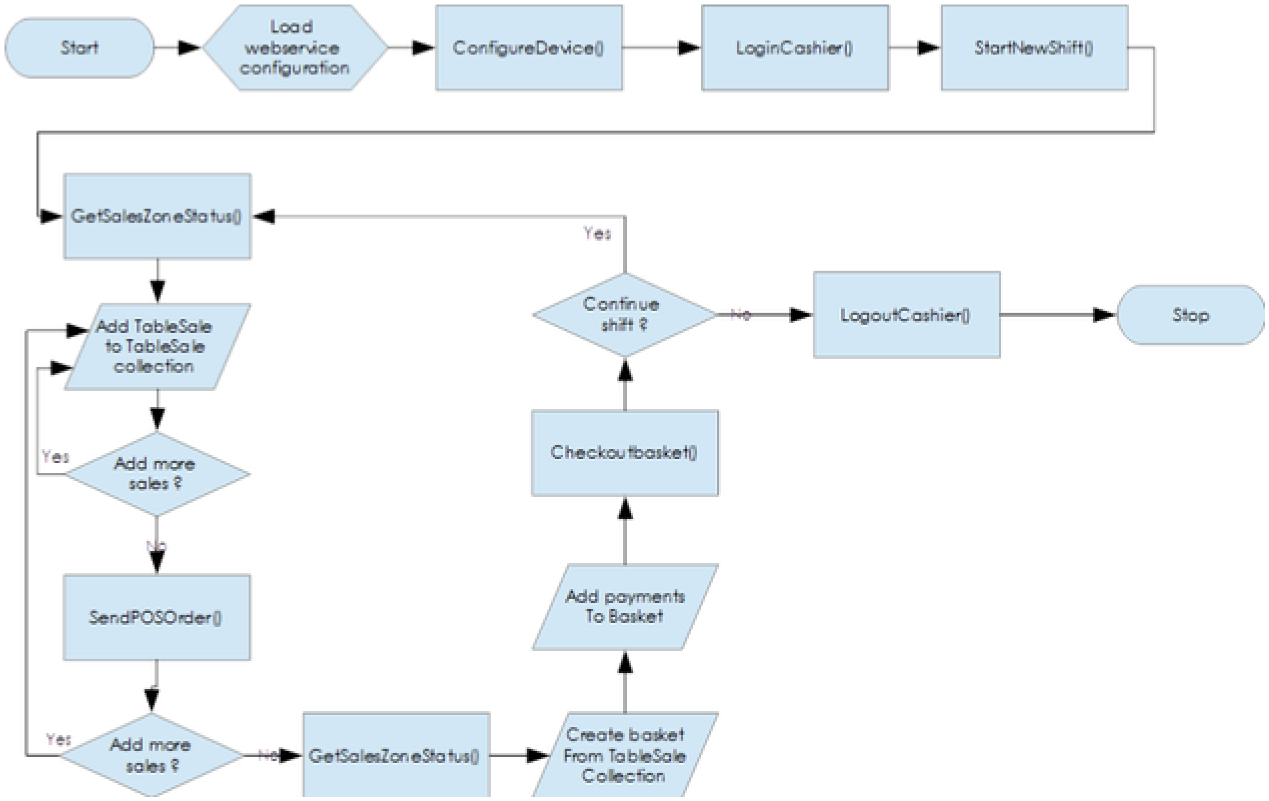
Response

The response contains a list of PrintDatum objects.

Waiterpad – Brief overview

The WaiterPad services are being used as an API for the mobile POS (mPOS) to integrate with the existing POS functionalities in ReCreateX.

Workflow



Startup

When the application starts up, two paramets are required in order to start with the API.

- WebService URL
- ShopID : This is a required property of the ServiceContext to be able to connect to the correct database.

These should be communicated when the application is installed.
Custom settings can be loaded and saved using the methods LoadGuiConfiguration / SaveGuiConfiguration.

Loading settings

With the ConfigureDevice method, the user can load the list of available employees (waiters / cashiers), PointOfSales and SalesZones.
After selecting one employee and a point of sales, the user can login by using his PIN code by using the LoginCashier request. This request returns advanced settings for the employee and the point of sales.
The PointOfSales configuration contains the entire configuration of the menu structure used by the POS.Both the ReCreateX menu button structure and item categories are in this structure, which enables the consumer of the API to present either the menu structure as it has been defined in ReCreateX, or it can represent a tree structure in which items are grouped by their item category.
Finally, the GetSalesZoneStatus method can be used to get the current status of a selected SalesZone.The response of this request returns the collection of tables present in the SalesZone. For each table, both the status and the order details are present in the result set.

Starting and ending shifts

When the waiter logs in, he receives a POSConfiguration, which contains the EmployeeConfiguration. The EmployeeConfiguration object contains the shift information for this employee.
If there is no current shift, the employee should start a new shift, using the StartNewShift method. Optionally, a starting cash balance can be specified.
At the end of the day, the user should close his shift, by using the methd LogoutCashier. In this request, the user specify whether or not he whishes to close his shift.

Adding order lines to a table

When the waiter takes orders on his device, each order line should be added to a local collection of TableSales. When the customer confirms his order, these orderlines can be sent to the POS by using the method SendPosOrder. Amongst other properties, this method accepts a collection of TableSale, so the local list of order line can be passed to the webservice request.
When creating new order lines, the Id of a TableSale should not be specified by the consumer of the API. This is the responsibility of the webservice logic. Alle lines with empty Id's are considered new order lines.

The response of SendPOSOrder contains the updated list of TableSale, where all Id fields will be completed by the webservice logic. Also some other fields (like unitprice) can be altered by the webservice logic.
If any tablesales is locally cashed on the device, it should allways be the collection returned in the response, and not the original collection passed in the request.
If multiple waiters are selling on the same table, the tablesales should be regularly refreshed. This can be done by using the method GetSalesZoneStatus, which returns the details of each table in the sales zone.

Checking out orders

Checking out (payment) of order is handled by the CheckoutBasket method of the API.
A basket contains a list of basktitems and a list payments.
These steps are required to complete the checkout process :

- Create a new basket
- Add all tablesales (which inherit from BaskteItem) to the items collection of the basketTo be sure that all table sales of the table is present in the basket, it would be preffered to call the GetSalesZoneStatus a last time.
- Perform a recalculate basket to get the total value of the Basket
- Use the ListPaymentmethods method to get the list of accepted payment methods
- Add the selected payment(s) to the list op payments in the basketFor regular payments (cash, visa, …) a BasketPayment object is required. When putting the amount on the to-go account, a PersonCardPayment should be provided, with the id of the person's card.
- Use the ValidateBasket method to validate if the basket can be processed
- Use the CheckoutBasket to book the payment of the bill.

Printing tickets

Printing of tickets is not handled by this API, because the API itself can be hosted on another location.
Instead, the printing functionality has been integrated in the ReCreateX backoffice software as an embeddd SOAP Service.
This printing service can be configured in ReCreateX under System > Settings > Application Settings.
Within the application settings window, there should be a key called "Embedded Webservice Settings" and within this key, there should be a child key "SalesDeviceService".
Within the child key, there should be an int value with the name "Port".
If these don't exist, they can be created by the user.
When the port setting exists, and is not zero, the SalesDeviceService is started when the ReCreateX application is started. The WSDL can be accessed under
http://localhost:<port>/SalesDeviceService
Note : <port> should be replaced by the actual value assigned to the port setting in recreatex.
The following tickets can be printed :

1. Order ticket for new orders

Sends a request to print a ticket for all items which have not been printed yet.
Requirement to print : The table should have at least one non printed item
SoapAction : PrintInterTicket
Arguments :

-
  - Guid salesZoneTableID : Id of the table
  - Guid employeeID : Id of the employee

1. Predefined message

Sends a request to print a predefined message for a given table
Requirement to print : the table should have at least one order line.
SoapAction : PrintInterTicketMessage
Arguments :

-
  - Guid gPointOfSaleID : Id of the current point of sales
  - Guid gTableID : Id of the table
  - Guid gEmployeeID : id of the current employee
  - Guid gMessageID : id of the message to print

1. Intermediate bill

Sends a request to print an intermediate bill for a given table
SoapAction : PrintIntermediateSalesTicket
Arguments :

-
    - Guid salesZoneTableID : Id of the table
    - Guid employeeID : Id of the current employee

1. Bill

Sends a request to print the bill for a payed order
SoapAction : PrintSalesHeader
Arguments :

-
    - Guid salesSeriesID : Id of the SalesSerie to print Should be provided by the checkout method ??

25 Find images using Webservices

Image service - Brief overview

Using the Webservices it is possible to show an image on a Webpage using a link. For this purpose you can use the following Webservice (WebShopImageService.svc) operations.

The Webservice allows us to call up the link of the specific article or exhibition, ...
This will return a link that has the following structure **TYPE/{ShopID}/{id}**

Using this link an image will be shown by calling up the ImageWebService with this link.

The following URL's are available:

Url	Method	Description
Activity/{ShopID}/{id}	**GET**	Returns the image for an activity.
Article/{ShopID}/{id}	**GET**	Returns the image for an article (Rental or Sale).
Article/{ShopID}/{id}/{number}	**GET**	Returns the image for an article (Rental or Sale).
ArticleAttachment/{ShopID}/{id}	**GET**	Returns the attachment for an article.
ArticleGroup/{ShopID}/{id}	**GET**	Returns the image for an article group.
Contact/{ShopID}/{id}	**GET**	Returns the image for a contact.
CultureSeries/{ShopId}/{id}	**GET**	Returns the image for a culture series
CultureSubscription/{ShopID}/{id}	**GET**	Returns the image for a culture subscription.
Event/{ShopID}/{id}	**GET**	Returns the image for an event (Culture Event).
Exposition/{ShopID}/{id}	**GET**	Returns the image for an exposition.
FitnessActivity/{ShopID}/{id}	**GET**	Returns the image for a fitness activity.
Hall/{ShopID}/{id}	**GET**	Returns the image for a hall
Person/{ShopID}/{id}	**GET**	Returns the image for a person.
Scenario/{ShopID}/{id}	**GET**	Returns the image for a scenario.
Employee/{ShopID}/{id}	**GET**	Returns the image for an employee.

Image service - Samples

Article

An Article has an ImageUrl that looks as follows.

Article/BD7F0911-D9EB-44F7-86CF-55851040C553/27B769BB-1D51-4904-8BC8-037B55E4B0C4

By calling up the WebShopImagewebservice with this parameter, the image for an Article with ID 27B769BB-1D51-4904-8BC8-037B55E4B0C4 on shop BD7F0911-D9EB-44F7-86CF-55851040C553 will be shown.

In our example, the full link will be: http://localhost/WebShopImageService.svc/Article/BD7F0911-D9EB-44F7-86CF-55851040C553/27B769BB-1D51-4904-8BC8-037B55E4B0C4

Person

A person has an ImageUrl that looks as follows.

Person/BD7F0911-D9EB-44F7-86CF-55851040C553/B6353F48-C20B-42D2-8EE3-E98FD4EF2BEE

By calling up the WebShopImagewebservice with this parameter, the image for a person with ID B6353F48-C20B-42D2-8EE3-E98FD4EF2BEE on shop BD7F0911-D9EB-44F7-86CF-55851040C553 will be shown.

In our example, the full link will be: http://localhost/WebShopImageService.svc/Article/BD7F0911-D9EB-44F7-86CF-55851040C553/27B769BB-1D51-4904-8BC8-037B55E4B0C4

Image service - Sizing

As from RCX 6.1.0, it is possible to provide the size of the image you wish to receive using the image service.
As a result, the client application can decide which width/height he is expecting.

This can improve performance (by reducing IO) as the image will be converted on the server before sending it to the client application.

To use the sizing, the client application can specify the following query parameters:

width	Desired width for the image to retrieve (in pixels)
height	Desired height for the image to retrieve (in pixels)

Example: http://localhost/WebShopImageService.svc/Article/BD7F0911-D9EB-44F7-86CF-55851040C553/27B769BB-1D51-4904-8BC8-037B55E4B0C4?**width=100&height=100**

Info

- It is possible to only provide one query parameter (so for example only the width)
  - Example: http://localhost/WebShopImageService.svc/Article/BD7F0911-D9EB-44F7-86CF-55851040C553/27B769BB-1D51-4904-8BC8-037B55E4B0C4?**width=100**
- The image will be resized to the maximum possible value when width and height are both provided
  - Example: When image is 500 (width) x 200 (height) and you specify width=100&height=100 in the query string, then the image will be resized to 100 (width) x 40 (height)
- The maximum value which can be provided within the query string parameters is **1024**. When providing a value which is bigger than 1024, the image will be returned in size 1024.
- When using the sizing options, the server processing will likely take a little bit longer (due to the server resizing the image), but the IO transfer will be reduced drastically.
- It is not possible to grow pictures.
  - Example: When image is 500 (width) x 200 (height) and you specify width=1000&height=400 in the query string, then the image will not be resized and the original image size will be returned.

26 Counter module (POS integration)

Counter – Brief overview

The module counter is available as of ReCreateX 5.7.0.
The system functions as follows:

- The articles to be sold are prepared (saved to the ReCreateX database) using the SaveCounterSales method.
- The prepared POS sales lines are processed within the ReCreateX POS.

The Webservice methods that have been created for this purpose are the following:

- SaveCounterSales (saves one or more POS sales lines within the ReCreateX database)

GetIntermediateSalesTicket

This method is used to receive the intermediate sales ticket in Epson format.

Request

Within the GetIntermediateSalesTicketRequest only a GetIntermediateSalesTicketCriteria object is expected in addition to the service context object. The GetIntermediateSalesTicketCriteria object is described below:

Name	Type	Description	Remarks
SalesZoneTableId	Guid	The unique identifier of the Sales zone table	
EmployeeId	Guid	The unique identifier of the employee	

Response

The GetIntermediateSalesTicketResponse contains an IntermediateSalesTicket. The IntermediateSalesTicket object is described below:

Name	Type	Description	Remarks
Id	Guid	The unique Id from the IntermediateSalesTicket	Not used for the moment
BinaryData	Byte[]	Byte array with the Epson codes ready to print	

GetInterTicket

This method is used to receive the inter ticket in Epson format.

Request

Within the GetInterTicketRequest only a GetInterTicketCriteria object is expected in addition to the service context object. The GetInterTicketCriteria object is described below:

Name	Type	Description	Remarks
SalesZoneTableId	Guid	The unique identifier of the Sales zone table	
EmployeeId	Guid	The unique identifier of the Employee	

Response

The GetInterTicketResponse contains an InterTicket. The InterTicket object is described below:

Name	Type	Description	Remarks
Id	Guid	The unique Id from the SalesReceipt	Not used for the moment
KitchenTicketBinaryData	Byte[]	Byte array with the Epson codes ready to print	
BarTicketBinaryData	Byte[]	Byte array with the Epson codes ready to print	

GetInterTicketMessage

This method is used to receive the Inter ticket message in Epson format.

Request

Within the GetInterTicketMessageRequest only a GetInterTicketMessageCriteria object is expected in addition to the service context object. The GetInterTicketMessageCriteria object is described below:

Name	Type	Description	Remarks
PointOfSaleId	Guid	The unique identifier of the Point Of Sale	
SalesZoneTableId	Guid	The unique identiefier of the Sales zone table	
MessageId	Guid	The unique identifier of the Message	
EmployeeId	Guid	The unique identiefier of the Employee	

Response

The GetInterTicketMessageResponse contains an InterTicketMessage. The InterTicketMessage object is described below:

Name	Type	Description	Remarks
Id	Guid	The unique Id from the SalesReceipt	Not used for the moment
BinaryData	Byte[]	Byte array with the Epson codes ready to print	

GetSalesHeaderKitchenBarTicket

This method is used to receive the sales header kitchen and bar ticket in Epson format.

Request

Within the GetSalesHeaderKitchenBarTicketRequest only a GetSalesHeaderKitchenBarTicketCriteria object is expected in addition to the service context object. The GGetSalesHeaderKitchenBarTicketCriteria object is described below:

Name	Type	Description	Remarks
SalesSerieId	Guid	The unique identifier of the Sales serie	Same id as the sessionId from the service context

Response

The GetSalesHeaderKitchenBarTicketResponse contains an KitchenBarTicket. The KitchenBarTicket object is described below:

Name	Type	Description	Remarks
Id	Guid	The unique Id from the SalesReceipt	Not used for the moment
KitchenTicketBinaryData	Byte[]	Byte array with the Epson codes ready to print	
BarTicketBinaryData	Byte[]	Byte array with the Epson codes ready to print	

GetSalesHeaderReceipt

This method is used to receive the receipt ticket in Epson format.

Request

Within the GetSalesHeaderReceiptRequest only a GetSalesHeaderReceiptCriteria object is expected in addition to the service context object. The GetSalesHeaderReceiptCriteria object is described below:

Name	Type	Description	Remarks
SalesSerieId	Guid	The unique identifier of the Sales serie	Same id as the sessionId from the service context
PrinterType	TicketPrinterType?		
PaperSizeWidth	decimal		
PrintAccessTickets	bool	Indiciates if the access tickets should be printed	

Response

The GetSalesHeaderReceiptResponse contains an SalesReceipt. The SalesReceipt object is described below:

Name	Type	Description	Remarks
Id	Guid	The unique Id from the SalesReceipt	Not used for the moment
BinaryData	Byte[]	Byte array with the Epson codes ready to print	

SaveCounterSales

This method is used to validate the credentials of a child care centre if it wants to log in.

Request

Within the SaveCounterSalesRequest only a SaveCounterSalesCriteria object is expected in addition to the service context object. The SaveCounterSalesCriteria object is described below:

Name	Type	Description	Remarks
PointOfSaleID	Guid	The unique identifier of the point of sale	
SaleCustomerID	Guid	The unique identifier of the sale customer	
CounterSales	IList<CounterSale>	List one or more counter sales. This can be of type: • ArticleCounterSale	

The ArticleCounterSale object is described below:

Name	Type	Description	Remarks
Comment	String	Extra information for the sale	
ArticleID	Guid	The unique identifier of the article	
Quantity	Int	Number of articles to be sold	
SubscriptionCustomerID	Guid?	The unique identifier of the customer that needs to be linked to the subscription	Only applicable to subscription articles
SubscriptionStartDate	DateTime?	The subscription start date	
SubscriptionEndDate	DateTime?	The subscription end date	
CustomPrice	Decimal	The custom unit price which should be used	Only applicable when configured in RCX.

Response

The SaveCounterSalesResponse contains an CounterValidationResult that indicates if the transaction succeeded or not.

Name	Type	Description	Remarks
BrokenRuleName	String	The error code.	
Message	String	The standard error.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	
TotalAmount	Decimal	The total amount of the items to be sold.	

Using the mPOS 2.0 webservices

The mPOS 2.0 webservices have been created to allow the mPOS application to sell both horeca and non-horeca items on a mobile device. They integrate with the existing POS in ReCreateX, and allow selling with and without tables.

This is an expansion on the already existing mPOS webservices.

You can sell the following non-horeca items:

- Activity reservation
- Fitness reservation
- Exposition reservation
- Place reservation
- Ticketing reservation
- Rental reservation

Availability
This functionality is available from RCX 5.11.0.x and WSDL 3.7.0.x

- Related articles

Related articles

Using the mPOS 2.0 webservices

ICD Document 2.0.0.0 : Sprint 2

ICD Document 2.0.0.0 : Sprint 3

GetCashReportJson

This method is used to generate the cash report in JSON.

The JSON contains commands with properties.

This can be used for Adyen Print.

ID	Command	Properties	Explanation

1	Open printer	N/A	
2	Close device	N/A	
3	Epson printer type	Value	0 = TM_88III 1 = TM_U220 2 = TM_P60II 3 = TM_P80
4	Cut paper	N/A	
5	Flush	N/A	
6	Sound buzzer	N/A	
7	Print logo	ImageSlot	Integer value, the slot that contains the image that should be printed. [obsolete]
8	Open frame		
9	Print text in frame	Text	
10	Split frame	N/A	
11	Close frame	N/A	
12	Print line	N/A	
13	Print half line	N/A	
14	Set font	Value	0 = F12x35 (large) 1 = F9x17 (small)
15	Set emphasized	Value	true/false
16	Set underlined	Value	true/false
17	Print text	Text	
18	Print code 128	Code	
19	Print code 93	Code	
20	Print EAN 13	Code	
21	Print QR code	Code	

Request

GetCashReportJsonCriteria

Name	Type	Description	Remarks
SalesShiftId	Guid	The unique identifier of the Sales shift	
SmallFontTextWidth	int	The amount of characters that should fit on one line with the small font	Default is 56
LargeFontTextWidth	int	The amount of characters that should fit on one line with the large font	Default is 42

Response

GetCashReportJsonResult

Name	Type	Description	Remarks
JsonPrintData	string	The printdata that was generated	The JSON is base64 encoded

GetSalesHeaderReceiptPrintDataJson

This method is used to receive the receipt ticket in JSON.

The JSON contains commands with properties.

This can be used for Adyen Print.

ID	Command	Properties	Explanation
1	Open printer	N/A	

2	Close device	N/A	
3	Epson printer type	Value	0 = TM_88III 1 = TM_U220 2 = TM_P60II 3 = TM_P80
4	Cut paper	N/A	
5	Flush	N/A	
6	Sound buzzer	N/A	
7	Print logo	ImageSlot	Integer value, the slot that contains the image that should be printed. [obsolete]
8	Open frame		
9	Print text in frame	Text	
10	Split frame	N/A	
11	Close frame	N/A	
12	Print line	N/A	
13	Print half line	N/A	
14	Set font	Value	0 = F12x35 (large) 1 = F9x17 (small)
15	Set emphasized	Value	true/false
16	Set underlined	Value	true/false
17	Print text	Text	
18	Print code 128	Code	
19	Print code 93	Code	
20	Print EAN 13	Code	
21	Print QR code	Code	

Request

GetSalesHeaderReceiptPrintDataJsonCriteria

Name	Type	Description	Remarks
SalesSeriesId	Guid	The unique identifier of the Sales serie	Same id as the sessionId from the service context
SmallFontTextWidth	int	The amount of characters that should fit on one line with the small font	Default is 56
LargeFontTextWidth	int	The amount of characters that should fit on one line with the large font	Default is 42
PrintAccessTickets	bool	Indicates if the access tickets should be printed	
ReprintReceiptsLinkedToDigitalWallet	bool	Indicates if the receipts linked to the digital wallet should be reprinted	

Response

GetSalesHeaderReceiptPrintDataJsonResult

Name	Type	Description	Remarks
JsonPrintData	string	The printdata that was generated	The JSON is base64 encoded

27 Scenario module (as of RCX 5.8.0)

DeleteScenarioSession

This method is used to delete a scenario session.

Request

The request consists of a service context and a scenario session GUID.

Name	Type	Description	Remarks
ScenarioSessionId	GUID	Unique identifier of the scenario session id.	

Response

See 8.8.2.2: ValidationResult

DeleteScenarioSessionAttachment

This method is used to delete a scenario session attachment

Request

The request consists of a service context and a scenario session attachment GUID.

Name	Type	Description	Remarks
ScenarioSessionAttachmentId	GUID	Unique identifier of the scenario session attachment.	

Response

ValidationResult

DeleteScenarioSessionComment

This method is used to delete a scenario session comment.

Request

The request consists of a service context and a scenario session GUID.

Name	Type	Description	Remarks
ScenarioSessionCommentId	GUID	Unique identifier of the scenario session comment id.	

Response

ValidationResult

FindScenarios

This method is used to find the scenarios as defined in ReCreateX.
It will return all the scenarios which are published for web usage.

Request

The request consists of a service context and a ScenarioSearchCriteria object.
The properties of a ScenarioSearchCriteria look as follows:

ScenarioSearchCriteria

Name	Type	Description	Remarks
ScenarioId	GUID?	ID of the scenario.	
Keyword	String	A keyword to search for.	
Paging	PagingCriteria	Paging criteria.	
Includes	ScenarioIncludes	This object determines which characteristics of the scenarios will be returned.	

ScenarioIncludes

Name	Type	Description	Remarks
Image	Bool	Include the image.	
ImageUri	Bool	Include the image URL.	
Steps	Bool	Include the steps.	
Questions	Bool	Include the questions.	
QuestionInputs	Bool	Include the question inputs.	
Conditions	Bool	Include the conditions.	
NonPublished	Bool	Include the non-published scenarios.	

Response

The response will contain a list of Scenario objects that meet the criteria.
The **Scenario** object looks as follows:

Scenario

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the scenario.	
Code	String	The shortname (code) of the scenario.	
Title	String	The title of the scenario.	
Description	String	The description of the scenario.	
Image	Picture	The image of the scenario.	
ImageUri	String	The image URL of the scenario.	
Version	Int	The version of the scenario.	
AskEmailsOnComplete	Bool	Determines whether to ask for additional e-mails when completing a scenario session.	
Steps	IList<ScenarioStep>	The list of steps within the scenario.	
Conditions	IList<ScenarioCondition>	The list of conditions for the scenario.	
IsLinkToFile	bool	Determines whether to ask a file number when starting a scenario session.	

ScenarioStep

Name	Type	Description	Remarks
Id	GUID	ID of the scenario step.	
Title	String	The title of the scenario step.	
Description	String	The description of the scenario step.	
Position	Int	Position (order) of the scenario step.	
InfoHtml	String	Additional HTML info for the scenario step.	
IsAlwaysEditable	Bool	Determines whether the scenario step is always editable (also when scenario is completed)	
Questions	IList<ScenarioQuestion>	The list of questions within the scenario step.	

ScenarioQuestion

Name	Type	Description	Remarks
Id	GUID	ID of the scenario question.	
Title	String	The title of the scenario question.	
Description	String	The description of the scenario question.	
Position	Int	Position (order) of the scenario question.	
IsYesNo	Bool	Determines whether the scenario question is a Yes/No question which needs to be answered.	

InfoHtml	String	Additional HTML info for the scenario question.	
IsAlwaysEditable	Bool	Determines whether the scenario question is always editable (also when scenario is completed)	
CodeYes	String	Code which is applied when the scenario question is answered with "Yes"	Only applicable if IsYesNo = true
CodeNo	String	Code which is applied when the scenario question is answered with "No"	Only applicable if IsYesNo = true
RequiredCodes	String	Codes which are required to visualise the scenario question	Only applicable if IsYesNo = true
Inputs	IList<ScenarioQuestionInput>	The list of inputs within the scenario question.	

ScenarioQuestionInput

The ScenarioQuestionInput object has multiple overrides:

- ScenarioQuestionInputText
- ScenarioQuestionInputInt
- ScenarioQuestionInputDateTime
- ScenarioQuestionInputBool
- ScenarioQuestionInputDropDownList
- ScenarioQuestionInputChoiceList
- ScenarioQuestionInputRadioList
- ScenarioQuestionInputReservation
- ScenarioQuestionInputRental
- ScenarioQuestionInputAttachment
- ScenarioQuestionInputCdb
- ScenarioQuestionInputDecimal
- ScenarioQuestionInputGeo
- ScenarioQuestionInputFileGeneral
- ScenarioQuestionInputFileDetail

Following fields are defined in all overrides:

Name	Type	Description	Remarks
Id	GUID	ID of the scenario question input.	
Description	String	The description of the scenario question input.	
ShowDescription	Bool	Determines whether the description is shown.	
Label	String	The label of the scenario question input.	
ShowLabel	Bool	Determines whether the label is shown.	
LabelPosition	ScenarioQuestionInputLabelPosition	Determines the position of the label compared to the input field	
Position	Int	Position (order) of the scenario question input.	
Enabled	Bool	Determines whether the scenario question input is enabled.	
Mandatory	Bool	Determines whether the scenario question input is mandatory.	
ReadOnly	Bool	Determines whether the scenario question input is read only.	
InfoHtml	String	Additional HTML info for the scenario question input.	
IsAlwaysEditable	Bool	Determines whether the scenario question input is always editable (also when scenario is completed).	

ScenarioQuestionInputText

Name	Type	Description	Remarks
DefaultValue	String	The default value for the text control.	
Multiline	Bool	Determines whether the text control is multiline.	
Width	Int	The width for the text control.	
Height	Int	The height for the text control.	

ScenarioQuestionInputInt

Name	Type	Description	Remarks
DefaultValue	Int	The default value for the numeric control.	
MinValue	Int	The minimum value for the numeric control.	
MaxValue	Int	The maximum value for the numeric control.	
Width	Int	The width for the numeric control.	

ScenarioQuestionInputDateTime

Name	Type	Description	Remarks
DefaultValue	DateTime?	The default value for the date control.	
DateTimeMode	ScenarioQuestionInputDateTimeMode	The mode for the date control: • DateTime • Date • Time	
Width	Int	The width for the date control.	
DefaultMasterDate	Bool	Determines whether the master date is automatically used as default date.	
MinDateTimeRegEx	String	Regular expression which is used for calculating the minimum date value.	
MaxDateTimeRegEx	String	Regular expression which is used for calculating the maximum date value.	

ScenarioQuestionInputBool

Name	Type	Description	Remarks
DefaultValue	Bool	The default value for the checkbox control.	

ScenarioQuestionInputDropDownList

Name	Type	Description	Remarks
DefaultValue	Int	The default value (index) for the dropdown control.	
Values	List<string>	List of possible values for the dropdown control.	
LookupEnabled	Bool	Determines whether the dropdown is filled by a ReCreateX lookup field.	
LookupValues	List<ScenarioLookupValue>	List of possible values (Value / Text) for the dropdown control when using a ReCreateX lookup field.	
Width	Int	The width for the dropdown control	

ScenarioQuestionInputChoiceList

Name	Type	Description	Remarks
DefaultValue	Int	The default value (index) for the list control.	
Values	List<string>	List of possible values for the list control.	
MultipleSelect	Bool	Determines whether multiple values can be selected within the list control.	
Width	Int	The width for the list control.	

ScenarioQuestionInputRadioList

Name	Type	Description	Remarks
DefaultValue	Int	The default value (index) for the radio control.	
Values	List<string>	List of possible values for the radio control.	

Columns	Int	Number of columns for the radio control.	
Orientation	ScenarioQuestionInputListOrientation	Determines the orientation of the radio control: • Horizontal • Vertical	

ScenarioQuestionInputReservation

Name	Type	Description	Remarks
PlaceIds	List<Guid>	List of place Ids for which reservations can be made.	
MultipleRecordsEnabled	Bool	Determines whether multiple reservations can be made.	
DefaultMasterDate	Bool	Determines whether the master date is automatically used as reservation date.	

ScenarioQuestionInputRental

Name	Type	Description	Remarks
RentalArticleIds	List<Guid>	List of rental article Ids for which rental orders can be made.	
MultipleRecordsEnabled	Bool	Determines whether multiple rental orders can be made.	
DefaultMasterDate	Bool	Determines whether the master date is automatically used as rental dates.	

ScenarioQuestionInputAttachment

Name	Type	Description	Remarks
ShowCdbFields	Bool	Determines whether the CDB fields are shown.	
MultipleRecordsEnabled	Bool	Determines whether multiple attachments can be made.	

ScenarioQuestionInputCdb

No additional properties…

ScenarioQuestionInputDecimal

Name	Type	Description	Remarks
DefaultValue	Int	The default value for the decimal control.	
Width	Int	The width for the decimal control.	

ScenarioQuestionInputGeo

Name	Type	Description	Remarks
DescriptionVisible	Bool	Determines whether the description field is visible.	
DescriptionMandatory	Bool	Determines whether the description field is mandatory.	
DescriptionValues	List<string>	List of description values.	
StreetVisible	Bool	Determines whether the street field is visible.	
StreetMandatory	Bool	Determines whether the street field is mandatory.	
NumberVisible	Bool	Determines whether the number field is visible.	
NumberMandatory	Bool	Determines whether the number field is mandatory.	
ZipcodeVisible	Bool	Determines whether the postal code, city and country fields are visible.	
ZipcodeMandatory	Bool	Determines whether the postal code, city and country fields are mandatory.	
CommentVisible	Bool	Determines whether the comment fields is visible.	
CommentMandatory	Bool	Determines whether the comment field is mandatory.	
MultipleRecordsEnabled	Bool	Determines whether multiple geo records can be made.	

DefaultMasterDate	Bool	Determines whether the master date is automatically used as geo dates.	
Zipcodes	List<Zipcode>	List of postal codes for which geo records can be made.	

ScenarioQuestionInputFileGeneral

Name	Type	Description	Remarks
FileGeneralNumberVisible	bool	Determines whether the number is visible.	
FileGeneralStatusVisible	bool	Determines whether the status is visible.	
FileGeneralAgeCategoryVisible	bool	Determines whether the age category is visible.	
FileGeneralDescriptionVisible	bool	Determines whether the description is visible.	
FileGeneralCustomerVisible	bool	Determines whether the customer is visible.	
FileGeneralContactVisible	bool	Determines whether the contact is visible.	
FileGeneralPriceGroupVisible	bool	Determines whether the price group is visible.	
FileGeneralDesiredPaymentMethodVisible	bool	Determines whether the desired payment method is visible.	
FileGeneralInvoiceCustomerVisible	bool	Determines whether the invoice customer is visible.	
FileGeneralFileManagerVisible	bool	Determines whether the file manager is visible.	
FileGeneralSalesManagerVisible	bool	Determines whether the sales manager is visible.	
FileGeneralExecutionDateTimeVisible	bool	Determines whether the execution datetime is visible.	
FileGeneralExpirationDateVisible	bool	Determines whether the expiration date is visible.	
FileGeneralPriceDateVisible	bool	Determines whether the price date is visible.	
FileGeneralCustomerReferenceVisible	bool	Determines whether the customer reference is visible.	
FileGeneralCommentVisible	bool	Determines whether the comment is visible.	
FileGeneralLeader1Visible	bool	Determines whether the first leader is visible.	
FileGeneralLeader2Visible	bool	Determines whether the second leader is visible.	
FileGeneralLeader3Visible	bool	Determines whether the thirth leader is visible.	
FileGeneralLeader4Visible	bool	Determines whether the fourth leader is visible.	
FileGeneralLeader5Visible	bool	Determines whether the fifth leader is visible.	

ScenarioQuestionInputFileDetail

Name	Type	Description	Remarks
FileDetailInvoiceVisible	bool	Determines whether the invoice is visible.	
FileDetailTypeVisible	bool	Determines whether the type is visible.	
FileDetailDescriptionVisible	bool	Determines whether the description is visible.	
FileDetailQuantityVisible	bool	Determines whether the quantity is visible.	
FileDetailTotalHoursVisible	bool	Determines whether the total hours are visible.	
FileDetailDateVisible	bool	Determines whether the date is visible.	
FileDetailCalculationTypeVisible	bool	Determines whether the calculation type is visible.	
FileDetailPriceVisible	bool	Determines whether the price is visible.	
FileDetailPriceGroupVisible	bool	Determines whether the price group is visible.	
FileDetailTotalPriceVisible	bool	Determines whether the total price is visible.	
FileDetailDepartmentVisible	bool	Determines whether the department is visible.	
FileDetailFileTypePartVisible	bool	Determines whether the file type part is visible.	
FileDetailFileActivityVisible	bool	Determines whether the file activity is visible.	
FileDetailPlaceVisible	bool	Determines whether the place is visible.	
FileDetailCommentVisible	bool	Determines whether the comment is visible.	

FileDetailStatusVisible	bool	Determines whether the status is visible.	
FileDetailStatusCommentVisible	bool	Determines whether the status comment is visible.	
FileDetailEmployeeVisible	bool	Determines whether the employee is visible	

ScenarioQuestionInputContact

Name	Type	Description	Remarks
NameVisible	bool	Determines whether the name is visible	
AddressVisible	bool	Determines whether the address is visible	
TelephoneVisible	bool	Determines whether the telephone is visible	
EmailVisible	bool	Determines whether the email is visible	
ExtraDescriptionVisible	bool	Determines whether the extra description is visible	

ScenarioQuestionInputLocation

Name	Type	Description	Remarks
NameVisible	bool	Determines whether the name is visible	
AddressVisible	bool	Determines whether the address is visible	
TelephoneVisible	bool	Determines whether the telephone is visible	
EmailVisible	bool	Determines whether the email is visible	
WebsiteVisible	bool	Determines whether the website is visible	
ExtraDescriptionVisible	bool	Determines whether the extra description is visible	

ScenarioQuestionInputTable

Name	Type	Description	Remarks

Table	String	JSON representation of a dynamic table
		How to read this JSON object:
		The columns of the dynamic table are described in this object as a table, which was filled in in the backoffice.
		This object contains columns and rows. The columns contain the columns as shown to the editor of the scenario in the backoffice and are not relevant here.
		They do describe how to interprete the rows property, which contain the columns of the dynamic table.
		The rows contain the different columns which should be shown to the user.
			Example:
			{
			  "columns": [
			    {
			      "name":
			"Name",
			      "type":
			"string",
			"possibleValues"
			: []
			    },
			    {
			      "name":
			"Type",
			      "type":
			"string",
			"possibleValues"
			: [

```
        "Text",

"Number",

"DateTime",

"Date",

"Time"
      ]
    },
    {
      "name":
"ExportCode",
      "type":
"string",

"possibleValues"
: []
    }
  ],
  "rows": [
    {
      "values":
[

"Voornaam",
        "Text",

"voornaam"
      ]
    },
    {
      "values":
[
        "Naam",
        "Text",
        "naam"
      ]
    },
    {
      "values":
[

"Geboortedatum",

"DateTime",

"geboortedatum"
      ]
```

			},
			{
			"values":
			[
			"Telefoontje",
			"Number",
			"telefoon"
			]
			}
			]
			}

MaxNumberOfRows	int	The maximum number of rows that should be entered by the user.	0 = unlimited

ScenarioQuestionInputSpotbooking

Name	Type	Description	Remarks
			This model contains no properties

ScenarioCondition

Name	Type	Description	Remarks
Id	Guid	Unique identifier of the scenario condition.	
ScenarioQuestionInputId	Guid	Unique identifier of the scenario question input to which the scenario condition applies.	
ConditionalOperator	ScenarioConditionalOperator	Determines the conditional operator that is used: • Equal = 0 • NotEqual = 1 • GreaterThan = 2 • LessThan = 3 • GreaterThanOrEqualTo = 4 • LessThanOrEqualTo = 5 • Between = 6 • NotBetween = 7 • Contains = 8 • NotContains = 9 • BeginsWith = 10 • EndsWith = 11 • In = 12 • NotIn = 13 • IsEmpty = 14 • NotIsEmpty = 15	
TargetID	Guid	Unique identifier of the target scenario step.	
ValueText	String	The scenario condition text value	
ValueTextBetween	String	The scenario condition text value 2	
ValueBool	Bool?	The scenario condition bool value	
ValueBoolBetween	Bool?	The scenario condition bool value 2	
ValueDateTime	DateTime?	The scenario condition date value	
ValueDateTimeBetween	DateTime?	The scenario condition date value 2	
ValueInt	Int?	The scenario condition int value	

ValueIntBetween	Int?	The scenario condition int value 2	

FindScenarioSessionAttachments

This method is used to find the scenarios attachments as defined in ReCreateX. (public attachments and attachment isn't linked to a question)

Request

The request consists of a service context and a ScenarioSearchCriteria object.
The properties of a ScenarioSearchCriteria look as follows:

FindScenarioSessionAttachmentsSearchCriteria

Name	Type	Description	Remarks
ScenarioQuestionInputSessionId	GUID	ID of the scenario session.	
Includes	FindScenarioSessionAttachmentsIncludes	Additional include object	

FindScenarioSessionAttachmentsIncludes

Name	Type	Description	Remarks
Internals	bool	Indicates if you also want to load internal attachments	

Response

The response will contain a list of **ScenarioSessionAttachment** objects that meet the criteria.

FindScenarioSessions

This method is used to find the scenario sessions as defined in ReCreateX.
It will return all the scenario sessions that apply to the search criteria.

Request

The request consists of a service context and a ScenarioSessionSearchCriteria object.
The properties of a **ScenarioSessionSearchCriteria** look as follows:

ScenarioSessionSearchCriteria

Name	Type	Description	Remarks
ScenarioSessionId	GUID?	Unique identifier of the scenario session.	
ScenarioId	GUID?	Unique identifier of the scenario.	
PersonId	GUID?	Unique identifier of the person that created the scenario session.	
Paging	PagingCriteria	Paging criteria.	
Includes	ScenarioSessionIncludes	This object determines which characteristics of the scenario sessions will be returned.	
FileId	Guid?	Unique identifier of the file linked with a scenario session	

ScenarioSessionIncludes

Name	Type	Description	Remarks
Value	Bool	Include the scenario question input values.	
Checks	Bool	Include the scenario session checks.	
PersonDetails	Bool	Include the person details.	
ScenarioQuestionInputValueFileGeneral	Bool	Include the ScenarioQuestionInputValueFileGeneral	
ScenarioQuestionInputValueFileDetails	Bool	Include the ScenarioQuestionInputValueFileDetails	
Approvers	Bool	Include the list of approvers (employees) on ScenarioSessionCheck.	
Comments	Bool	Include the comments (ScenarioSessionComments)	

Attachments	Bool	Inlcude the attachments (ScenarioSessionAttachments)	
AttachmentsByteStream	Bool	Include the bytestream of the attachments	Added to reduce bandwith traffic

Response

The response will contain a list of ScenarioSession objects that meet the criteria.
The **ScenarioSession** object looks as follows:

ScenarioSession

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the scenario session.	
Number	Long	Unique number of the scenario session.	
CreationDate	DateTime	Creation date of the scenario session.	
Description	String	Description of the scenario session.	
MasterDateStart	DateTime	Master date start.	
MasterDateEnd	DateTime	Master date end.	
ScenarioId	Guid	Unique identifier of the scenario for which the scenario session was created.	
CompletionDate	DateTime?	Completion date of the scenario session.	Only filled when scenario session is completed
PersonId	Guid	Unique identifier of the person who created the scenario session.	
Person	Person	Detailed data of the person who created the scenario session.	See 8.8.1.1: Person
ChecklistStatus	ScenarioChecklistStatus	Status of the scenario session. • Open = 0 • Approving = 1 • Approved = 2 • Disapproved = 3 • Ordered = 4 • Closed = 5	
ChecklistApprovalDate	DateTime?	Approval date of the scenario session.	Only filled when scenario session is approved
QuestionValues	IList<ScenarioSessionQuestionValue>	List of the scenario session question values (which are the answers on Yes/No questions).	
Values	IList<ScenarioSessionValue>	List of the scenario session values (which are the values on all inputs).	
Checks	IList<ScenarioSessionCheck>	List of the checks of the scenario session.	
Codes	List<string>	List of codes which are collected for the scenario session.	
ScenarioType	ScenarioType	The scenario type of the scenario session.	
FileId	Guid?	Contains the ID of the file	Only filled when the scenario session is linked with a file
FileNumber	string	Contains the number of the file	Only filled when the scenario session is linked with a file
FileDescription	string	Contains the description of the file	Only filled when the scenario session is linked with a file
Comments	IList<ScenarioSessionComment>	List of the comments related to the scenario	
Attachments	IList<ScenarioSessionAttachment>	List of the attachments related to the scenario	
SpotbookingStatus	SpotbookingStatus	Spotbooking status of the scenario session. ▪ None = 0 ▪ Pending = 1 ▪ Approved = 2	

		- Disapproved = 3	

ScenarioSessionQuestionValue

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the scenario session question value.	
ScenarioQuestionId	GUID	Unique identifier of the scenario question.	
YesNoValue	Bool?	Value of the Yes/No question.	

ScenarioSessionValue

The ScenarioSessionValue object has multiple overrides:

- ScenarioSessionValueText
- ScenarioSessionValueInt
- ScenarioSessionValueDateTime
- ScenarioSessionValueBool
- ScenarioSessionValueDropDownList
- ScenarioSessionValueChoiceList
- ScenarioSessionValueRadioList
- ScenarioSessionValueReservation
- ScenarioSessionValueRental
- ScenarioSessionValueAttachment
- ScenarioSessionValueCdb
- ScenarioSessionValueDecimal
- ScenarioSessionValueGeo
- ScenarioSessionValueLocation
- ScenarioSessionValueContact
- ScenarioSessionValueTable
- ScenarioSessionValueSpotbooking

Following fields are defined in all overrides:

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the scenario session value.	
ScenarioQuestionInputId	GUID	Unique identifier of the scenario question input.	

ScenarioSessionValueText

Name	Type	Description	Remarks
Value	String	Text value.	

ScenarioSessionValueInt

Name	Type	Description	Remarks
Value	Int?	Numeric value.	

ScenarioSessionValueDateTime

Name	Type	Description	Remarks
Value	DateTime?	Date value.	

ScenarioSessionValueBool

Name	Type	Description	Remarks
Value	Bool?	Boolean value.	

ScenarioSessionValueDropDownList

Name	Type	Description	Remarks

ValueGuid	Guid	Guid value.	Fill when input is LookupEnabled
ValueText	String	Text value (value).	
ValueInt	Int	Numeric value (index).	

ScenarioSessionValueChoiceList

Name	Type	Description	Remarks
ValueText	List<String>	List of selected values.	
ValueInt	Int	Selected value (index).	

ScenarioSessionValueRadioList

Name	Type	Description	Remarks
ValueInt	Int	Selected value (index).	
ValueText	String	Text value	

ScenarioSessionValueReservation

Name	Type	Description	Remarks
InfrastructureId	GUID	Unique identifier of the selected infrastructure.	
InfrastructureCode	String	Code of the selected infrastructure.	
InfrastructureDescription	String	Description of the selected infrastructure.	
PlaceId	GUID	Unique identifier of the selected place.	
PlaceCode	String	Code of the selected place.	
PlaceDescription	String	Description of the selected place.	
ReservationActivityId	GUID	Unique identifier of the selected reservation activity.	
ReservationActivityCode	String	Code of the selected reservation activity.	
ReservationActivityDescription	String	Description of the selected reservation activity.	
StartDate	DateTime	Start date and time of the reservation.	
EndDate	DateTime	End date and time of the reservation.	
Comment	String	Extra comment for the reservation.	

ScenarioSessionValueRental

Name	Type	Description	Remarks
StartDate	DateTime	Start date and time of the rental order.	
EndDate	DateTime	End date and time of the rental order.	
PlannedDeliveryDate	DateTime	Planned delivery date and time of the rental order.	
PlannedReturnDate	DateTime	Planned return date and time of the rental order.	
Lines	IList<ScenarioSessionValueRentalLine>	List of rental lines	

ScenarioSessionValueRentalLine

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the rental scenario rental line.	
ArticleId	GUID	Unique identifier of the rental article.	
ArticleCode	String	Code of the rental article.	
ArticleDescription	String	Description of the rental article.	
Quantity	Int	Quantity to rent	

StartDate	DateTime	Start date and time of the rental line.	
EndDate	DateTime	End date and time of the rental line.	
PlannedDeliveryDate	DateTime	Planned delivery date and time of the rental line.	
PlannedReturnDate	DateTime	Planned return date and time of the rental line.	

ScenarioSessionValueAttachment

Name	Type	Description	Remarks
AttachmentType	ScenarioAttachmentType	The type of attachment. • Picture = 0 • Word = 1 • Excel = 2 • Text = 3 • Html = 4 • Pdf = 5	
Attachment	Attachment	The file attachment.	
Comment	String	Extra comment concerning the attachment.	
CdbExport	Bool	Determines whether to export the attachment to CDB.	
CdbCopyright	String	The CDB copyright.	
CdbMediaFileType	String	The CDB media file type.	

ScenarioSessionValueCdb

Name	Type	Description	Remarks
CdbCategoryId	GUID	Unique identifier of the CDB category.	
CdbCategoryDescription	String	Description of the CDB category.	

ScenarioSessionValueDecimal

Name	Type	Description	Remarks
Value	Decimal?	Decimal value.	

ScenarioSessionValueGeo

Name	Type	Description	Remarks
DateFrom	DateTime	Start date for the geo record.	
DateTo	DateTime	End date for the geo record.	
Description	String	Description for the geo record.	
Address	Address	Address details for the geo record.	See 8.8.1.3 for details concerning the Address object.
Comment	String	Extra comment for the geo record.	

ScenarioSessionValueLocation

Name	**Type**	**Description**	**Remarks**
Location	Person	The selected location	

ScenarioSessionValueContact

Name	**Type**	**Description**	**Remarks**
Contact	Contact	The selected contact	

ScenarioSessionValueTable

Name	Type	Description	Remarks
Table	String	JSON representation of the filled in table. The columns contain the columns of the dynamic table, saying which type they represent. The rows contain the data entered by the user.	Example:

```
{
  "columns": [
    {
      "name": "Voornaam",
      "type": "Text",
      "possibleValues": null
    },
    {
      "name": "Naam",
      "type": "Text",
      "possibleValues": null
    },
    {
      "name": "Telefoon",
      "type": "Number",
      "possibleValues": null
    },
    {
      "name": "Datum",
      "type": "DateTime",
      "possibleValues": null
    }
  ],
  "rows": [
    {
      "values": [
        "test",
        "test",
        "1",
        "2017-10-28T10:00:00.0000000"
      ]
    },
    {
      "values": [
        "test2",
        "test2",
        "2",
        "2017-10-29T00:00:00.0000000"
      ]
    }
    ]
}
```

ScenarioSessionValueSpotbooking

Name	Type	Description	Remarks
SpotbookingId	int	The ID known in the spotbooking application	
Status	SpotbookingStatus	Spotbooking status: • None = 0 • Pending = 1 • Approved = 2 • Disapproved = 3	

ScenarioSessionCheck

Name	Type	Description	Remarks
ChecklistType	String	The checklist type.	
ChecklistCategory	String	The checklist category.	
ChecklistItem	String	The checklist item.	
Processed	DateTime?	The date and time the check was processed.	
Comment	String	Extra comment for the check.	
Status	ScenarioCheckStatus	The status for the check. • NeedsToBeChecked = 0 • CheckSucceeded = 1 • CheckFailed = 2	
Id	GUID	Unique identifier of the scenario session check	
SortOrder	Int	SortOrder (on generated checks)	
CheckListCategorySortOrder	Int	Sort Order on category	
CheckListItemId	GUID	Unique identieifier of the checklist item	
CheckListItemIsBlocking	Bool	Indicates whether the checklist item is blocking	
CheckListItemSortOrder	Int	Sort order (on non-generated checks)	
InternalComment	String	Internal comment of the check	
Approvers	IList<Employee>	List of employees that are authorized to approve the check	
ApprovedBy	Employee	Employee that (dis)approved the check	

ScenarioType

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the scenario type.	
Code	String	Code of the scenario type.	
Description	String	Description of the scenario type.	
Comment	String	Extra comment of the scenario type.	
Weight	Int	Weight of the scenario type.	
Codes	List<string>	List of codes that are necessary for the scenario type.	

ScenarioSessionAttachments

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the ScenarioSessionAttachment	
ScenarioSessionId	GUID	Unique identifier of the ScenarioSession	
AttachmentType	ScenarioAttachmentType	The type of attachment. • Picture = 0 • Word = 1 • Excel = 2 • Text = 3 • Html = 4 • Pdf = 5	
Attachment	Attachment	The file attachment.	
Comment	String	Extra comment concerning the attachment.	
DocumentUrl	string	The document url path	Sample: SCENARIOATTACHMENTS/00000000-0000-0000-0000-000000000640/2cee5708-9497-e711-84bc-4c72b941b3a7
Internal	bool	Indicates if this attachment is for internal use.	

ScenarioSessionComments

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the ScenarioSessionComment	
ScenarioSessionId	GUID	Unique identifier of the ScenarioSession	
EmployeeId	GUID	Unique identifier of the employee that submitted the comment	Overrides 'UserId' when filled in
CreationDate	Datetime	Creationdate of the comment	
UserId	GUID	Unique identifier of the user that submltted the comment	If the EmployeeId has been filled is, it overrides the 'UserId' field.
Description	String	Header of the comment	
Comment	String	The comment	

```
{
"columns": [
{
"name": "Voornaam",
"type": "Text",
"possibleValues": null
},
{
"name": "Naam",
"type": "Text",
"possibleValues": null
},
{
"name": "Telefoon",
"type": "Number",
"possibleValues": null
},
{
"name": "Datum",
"type": "DateTime",
"possibleValues": null
}
],
"rows": [
{
"values": [
"test",
"test",
"1",
"2017-10-28T10:00:00.0000000"
]
},
{
"values": [
"test2",
"test2",
"2",
"2017-10-29T00:00:00.0000000"
]
}
]
}
```

FindScenarioTypes

This method is used to find the scenario types as defined in ReCreateX.

Request

The request consists of a service context and a ScenarioTypeSearchCriteria object.
The properties of a **ScenarioTypeSearchCriteria** look as follows:

ScenarioTypeSearchCriteria

Name	Type	Description	Remarks
ScenarioTypeId	GUID?	ID of the scenario type.	
Keyword	String	A keyword to search for.	
Paging	PagingCriteria	Paging criteria.	

Response

The response will contain a list of ScenarioType objects that meet the criteria.

GetScenarioQuestionInputValueFileDetails

This method is used to find the detailed information of a file as defined in ReCreateX.

Request

The request consists of a service context and a ScenarioQuestionInputValueFileDetailSearchCriteria object.
The properties of a ScenarioQuestionInputValueFileDetailSearchCriteria as follows:

ScenarioQuestionInputValueFileGeneralSearchCriteria

Name	Type	Description	Remarks
FileId	GUID	ID of the file.	

Response

The response consists of a list with ScenarioQuestionInputValueFileDetail objects.
The properties of a ScenarioQuestionInputValueFileDetail object look as follows:

Name	Type	Description	Remarks
Id	Guid	The id of the subfile	
IsInvoice	bool	Indicates if the subfile is of type invoice	
Type	FileDetailType?	The file detail type	
Description	string	Description of the subfile	
Quantity	int?	Quantity	
TotalHours	int?	Total hours	

Date	DateTime?	Date	
CalculationType	FileItemCalculationType?	The calculation type	
Price	decimal?	The price	
CustomerPriceGroup	string	The customer price group	
TotalPrice	decimal?	The total price	
ItemDepartment	string	The item department	
FileTypePart	string	The file type part	
FileActivity	string	The file activity	
Place	string	The place	
Comment	string	The comment	
Status	string	The status	
StatusComment	string	The status comment	
Employee	string	The name and firstname of the employee	

GetScenarioQuestionInputValueFileGeneral

This method is used to find the general information of a file as defined in ReCreateX.

Request

The request consists of a service context and a ScenarioQuestionInputValueFileGeneralSearchCriteria object.
The properties of a ScenarioQuestionInputValueFileGeneralSearchCriterialook as follows:

ScenarioQuestionInputValueFileGeneralSearchCriteria

Name	Type	Description	Remarks
FileId	GUID	ID of the file.	

Response

The response consists of a ScenarioQuestionInputValueFileGeneral object.
The properties of a ScenarioQuestionInputValueFileGeneral object look as follows:

Name	Type	Description	Remarks
Number	String	The number of the file.	
Status	FileStatus	The file status	
AgeCategory	FileAgeGroup	The file age group	
Description	String	The description of the file.	
Customer	Person	The customer	
Contact	Contact	The contact	
PriceGroup	PriceGroup	The pricegroup	
DesiredPaymentMethod	BasketPaymentMethod	The desired payment method	
InvoiceCustomer	Person	The invoice customer	
FileManager	Employee	The file manager	
SalesManager	Employee	The sales manager	
ExecutionDateTime	DateTime?	The execution date & time	
ExpirationDate	DateTime?	The expiration date	
PriceDate	DateTime?	The price date	
CustomerReference	string	The customer reference	

Comment	string	The comment	
Leader1	Employee	The first leader	
Leader2	Employee	The second leader	
Leader3	Employee	The thirth leader	
Leader4	Employee	The fourth leader	
Leader5	Employee	The fifth leader	

GetScenarioSessionEmailMessage

This method is used to load a specific e-mail message that is used within the scenario module.

Request

The request consists of a service context and a ScenarioSessionEmailMessageSearchCriteria object.
The properties of a **ScenarioSessionEmailMessageSearchCriteria** look as follows:

ScenarioSessionEmailMessageSearchCriteria

Name	Type	Description	Remarks
ScenarioSessionId	GUID	Unique identifier of the scenario session.	
Type	ScenarioSessionEmailMessageType	The type of e-mail message you want to get: • Customer • FollowUpEmployee	

Response

The response will contain a ScenarioSessionEmailMessageValidationResult object that meet the criteria.
The **ScenarioSessionEmailMessageValidationResult** object looks as follows:

ScenarioSessionEmailMessageValidationResult

Name	Type	Description	Remarks
EmailMessage	ScenarioSessionEmailMessage	The e-mail message.	
EmailMessageType	ScenarioSessionEmailMessageType	The type of the e-mail message. • Customer • FollowUpEmployee	

ScenarioSessionEmailMessage

Name	Type	Description	Remarks
Subject	String	The e-mail subject.	
Body	String	The e-mail body.	
To	List<string>	The e-mail receiver.	
Person	Person	Person details of the e-mail receiver.	

GetScenarioSessionEmailMessages

Request

The request consists of a service context and a ScenarioSessionEmailMessagesSearchCriteria object.
The properties of a **ScenarioSessionEmailMessagesSearchCriteria** look as follows:

ScenarioSessionEmailMessagesSearchCriteria

Name	Type	Description	Remarks
ScenarioSessionId	GUID	Unique identifier of the scenario session.	

Response

The response will contain a list of ScenarioSessionEmailMessageValidationResult objects that meet the criteria.

ListCdbCategories

This method is used to list all CDB categories as defined in ReCreateX.
This list is used to fill the ScenarioSessionValueCdb object.

Request

The request consists of a service context object.

Response

The response will contain a list of CdbCategory objects that meet the criteria.

Name	Type	Description	Remarks
Id	GUID	Unique identifier of the CDB category.	
CNetId	String	CNetId of the CDB category.	
Description	String	Description of the CDB category.	
Notes	String	Extra notes of the CDB category.	
Synonymes	String	Synonyms of the CDB category.	
CdbVersion	String	Version of the CDB category.	

SaveScenarioSession

This method is used to save a scenario session.

Request

The request consists of a service context and a **ScenarioSession** object.

Name	Type	Description	Remarks
ScenarioSession	ScenarioSession	Details of the scenario session to save.	

Response

The response will contain a **ScenarioValidationResult** object with following values:

Name	Type	Description	Remarks
ScenarioSessionId	GUID	Unique identifier of the scenario session id that was saved.	
ScenarioType	ScenarioType	The type that was given to the scenario (based on the collected codes).	Only filled at scenario completion

SaveScenarioSessionAttachment

This method is used to save a scenario session attachment.

Request

The request consists of a service context and a **ScenarioSessionAttachment** object.

Name	Type	Description	Remarks
ScenarioSessionAttachment	ScenarioSessionAttachment	Details of the scenario session attachment to save.	

Response

The response will contain a **ScenarioSessionAttachmentValidationResult** object with following values:

Name	Type	Description	Remarks
ScenarioSessionAttachment	ScenarioSessionAttachment	The scenario session attachment	

SaveScenarioSessionComment

This method is used to save a scenario session comment.

Request

The request consists of a service context and a **ScenarioSessionComment** object.

Name	Type	Description	Remarks
ScenarioSessionComment	ScenarioSessionComment	Details of the scenario session comment to save.	

Response

The response will contain a **ScenarioSessionCommentValidationResult** object with following values:

Name	Type	Description	Remarks
ScenarioSessionComment	ScenarioSessionComment	The scenario session comment	

SaveSpotbookingStatus

This method is used to approve or disapprove a spotbooking control

Request

The request consists of a service context and a SaveSpotbookingStatusCriteria object.
The properties of a **SaveSpotbookingStatusCriteria** look as follows:

SaveSpotbookingStatusCriteria

Name	Type	Description	Remarks
SpotbookingId	int	The spotbooking ID	
IsApproved	bool	Determines whether the spotbooking control must be approved or disapproved	

Response

The response is a SaveSpotbookingStatusResult enum.

SaveSpotbookingStatusResult contains the following values:

- Success = 0
- NotFound = 1
- IsAlreadyAccepted = 2
- IsAlreadyRejected = 3
- SaveFailed = 4

Scenarios – Brief overview

As of ReCreateX 5.8.0 it is possible to use the Scenario module.
The system functions as follows:

- A scenario flow is defined in ReCreateX and published for web usage.
- A person creates a new scenario session (based on a scenario as defined in ReCreateX), navigates through all steps, fills in the fields and completes his scenario session.
- The completed scenario sessions are managed and checked in ReCreateX.

The Webservice methods that have been created for this purpose are the following:

- FindScenarios (find the scenarios)
- FindScenarioSessions (find existing scenario sessions)
- DeleteScenarioSession (delete one scenario session)
- SaveScenarioSession (save one scenario session)
- FindScenarioTypes (find the scenario types)
- ListCdbCategories (list all CDB categories)
- GetScenarioSessionEmailMessage (get one scenario session e-mail message)

- GetScenarioSessionEmailMessages (get all scenario session e-mail messages)
- SaveScenarioSessionAttachment (save one scenario session attachment)
- DeleteScenarioSessionAttachment (delete one scenario session attachment)
- SaveScenarioSessionComment (save on scenario session comment)
- DeleteScenarioSessionComment (delete one scenario session comment)

ValidateScenarioSessionCheck

This method is used to validate a scenario session check.

Request

The request consists of a service context and a **ValidatedScenarioSessionCheck** object.

Name	Type	Description	Remarks
Validation	ValidatedScenarioSessionCheck	Details of the validated scenario check	

ValidatedScenarioSessionCheck

Name	Type	Description	Remarks
ScenarioSessionCheckID	Guid	Id of the session that should be validated	
ValidatorID	Guid	Id of the employee that validates the check	
IsApproved	Bool	Indicates whether the validater approves or rejects the check	

Response

The response will contain a **ValidatedScenarioSessionCheckResult** object with following values:

Name	Type	Description	Remarks
ValidationResult	ScenarioSessionCheckValidationResult	Contains the ValidatedScenarioSessionCheck	

28 Manager module

ListCONSOLIDATIONCODES

It is important that the user of the manager app can filter the result with using consolidation codes. Consolidation codes can be devided into types.
The necessary types in the manager app are 'item', 'activity' and 'sale'.

Request

The requests consist of a service context and a ConsolidationCodeSearchCriteria object.
The properties of a ConsolidationCodeSearchCriteria object looks as follows:

ConsolidationCodeSearchCriteria

Name	Type	Description	Remarks
Type	Int	The requested type	0 = item 1 = activity 4 = sale Default = 0 (item)

Response

The response will contain a list of ConsolidationCode objects that meet the criteria.
The **ConsolidationCode** object looks as follows:

Name	Type	Description	Remarks

Id	Guid	Unique identifier of the ConsolidationCode	
ShortName	String	The short name of the Consolidation Code	
Description	String	The description of the Consolidation Code	

Listdivisions

Request

The request consists of a service context and a DivisionSearchCriteria object.
The properties of a DivisionSearchCriteria looks as follows:

DivisionSearchCriteria

The DivisionSearchCriteria contains no properties.

Name	Type	Description
AllowedForPointOfSale	Bool	Will only return Divisions that are allowed for current PointOfSale

Response

The response will contain a list of Division objects that meet the criteria.
The **Division** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	Unique identifier of the division.	
Name	String	The name of the division	
Address	Address	The address of the division	
Phone	String	The phone of the division	
Fax	String	The fax of the division	
Email	String	The email of the division	
Website	String	The website of the division	
BankAccount	String	The bankaccount of the division	
Comments	String	The comments of the division	
Credential	Crediantial	The credential of the division	
IBAN	String	The iban of the division	
BIC	STRING	The bic of the division	

ListSalesInformation

Request

The request consists of a service context and a SaleInformationSearchCriteria object.
The properties of a SaleInformationSearchCriteria looks as follows:

SaleInformationSearchCriteria

Name	Type	Description	Remarks
DivisionIds	List<Guid>	Filter divisions on the ids inside the collection. If the collection is empty, take all divisions.	
From	DateTime?	Sale information occuring after this date	Optional
Until	DateTime?	Sale information occurring before this date	Optional
ArticleGroupIds	List<Guid>	Filter article groups on the ids inside the collection. If the collection is empty, take all article groups.	Possible to receive all ArticleGroups with the method 'ListArticleGroups'
ConsolidationCodeItemCodeIds	List<Guid>	Filter consolidationcodes on the ids inside the collection. If the collection is empty, take all consolidation codes.	

ConsolidationCodeActivityCodeIds	List<Guid>	Filter consolidationcodes on the ids inside the collection. If the collection is empty, take all consolidation codes.	
ConsolidationCodeSaleCodeIds	List<Guid>	Filter consolidationcodes on the ids inside the collection. If the collection is empty, take all consolidation codes.	
GroupByDivision	Bool	Group sale information by division	Default: false
GroupByDate	Bool	Group sale information by date	Default: false
GroupByArticleGroup	Bool	Group sale information by article group	Default: false
GroupByConsolidationCodeItem	Bool	Group sale information by ConsolidationCode (item)	Default: false
GroupByConsolidationCodeActivity	Bool	Group sale information by ConsolidationCode (article)	Default: false
GroupByConsolidationCodeSales	Bool	Group sale information by ConsolidationCode (sale)	Default: false

Response

The response will contain a list of SaleInformation objects that meets the criteria.
The **SaleInformation** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	The ID of the sale information	
Amount	Decimal	The amount of the sale	(Amount = LineAmount + VatAmount)
LineAmount	Decimal	The LineAmount of the sale	
VatAmount	Decimal	The VatAmount of the sale	
Date	DateTime?	The date of the sale. Will only contain data when the property 'GroupByDate' in the search criteria is true	
DivisionID	Guid?	The division of the sale. Will only contain data when the property 'GroupByDivision' in the search criteria is true	
ArticleGroupID	Guid?	The article group of the sale. Will only contain data when the property 'GroupByArticleGroup in the search criteria is true	
ConsolidationCodeItemID	Guid?	The ConsolidationCodeItem of the sale. Will only contain data when the property 'GroupByConsolidationCodeItem' in the search criteria is true	
ConsolidationCodeActivityID	Guid?	The ConsolidationCodeActivity of the sale. Will only contain data when the property 'GroupByConsolidationCodeActivity' in the search criteria is true	
ConsolidationCodeSalesID	Guid?	The ConsolidationCodeSale of the sale. Will only contain data when the property 'GroupByConsolidationCodeSales' in the search criteria is true	

ListVisitingCustomersInformation

Request

The request consists of a service context and a VisitingCustomerInformationSearchCriteria object.
The properties of a VisitingCustomerInformationSearchCriteria looks as follows:

VisitingCustomerInformationSearchCriteria

Name	Type	Description	Remarks
DivisionIds	List<Guid>	Filter divisions on the ids inside the collection. If the collection is empty, take all divisions.	
From	DateTime?	Customer information occurred after this date	Optional
Until	DateTime?	Customer information occurred before this date	Optional
ArticleGroupIds	List<Guid>	Filter article groups on the ids inside the collection. If the collection is empty, take all article groups.	Possible to receive all ArticleGroups with the method 'ListArticleGroups'
ConsolidationCodeItemIds	List<Guid>	Filter consolidationcodes on the ids inside the collection. If the collection is empty, take all consolidation codes.	
ConsolidationCodeActivityIds	List<Guid>	Filter consolidationcodes on the ids inside the collection. If the collection is empty, take all consolidation codes.	
ConsolidationCodeSalesIds	List<Guid>	Filter consolidationcodes on the ids inside the collection. If the collection is empty, take all consolidation codes.	
GroupByDIvision	Bool	Customer information by division	Optional
GroupByDate	Bool	Customer information by date	Optional

GroupByArticleGroup	Bool	Customer information by article group	Optional
GroupByConsolidationCodeItem	Bool	Customer information by consolidation code (item)	Optional
GroupByConsolidationCodeActivity	Bool	Customer information by consolidation code (activity)	Optional
GroupByConsolidationCodeSales	Bool	Customer information by consolidation code (sale)	Optional

Response

The response will contain a list of VisitingCustomer objecstthat meet the cirteria.
The **visitingCustomer** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	The ID of the customer information	
TotalVisitors	Int	The number of visitors	TotalVisitors = SubscriptionVisitors + SaleVisitors
SubscriptionVisitors	Int	The number of visitors with a subscription	
SaleVisitors	Int	The number of visitors with a sale	
Date	DateTime?	The date of the visit. Will only contain data when the property 'GroupByDate' in the search criteria is true	
DivisionID	Guid?	The division of the visit. Will only contain data when the property 'GroupByDivision' in the search criteria is true	
ArticleGroupID	Guid?	The article group of the visit. Will only contain data when the property 'GroupByArticleGroup in the search criteria is true	
ConsolidationCodeActivityID	Guid?	The ConsolidationCodeActivity of the visit. Will only contain data when the property 'GroupByConsolidationCodeActivity' in the search criteria is true	
ConsoldiationCodeItemID	Guid?	The ConsolidationCodeItem of the visit. Will only contain data when the property 'GroupByConsolidationCodeItem' in the search criteria is true	
ConsolidationCodeSalesID	Guid?	The ConsolidationCodeSale of the visit. Will only contain data when the property 'GroupByConsolidationCodeSales' in the search criteria is true	

Manager – Brief overview

As of ReCreateX 5.8.0 it is possible to use the Manager module (specifically built for the manager app).The manager app is used to receive sale information and customer information. It is possible to filter or to group the information by it's data with using the (optional) parameters.
The Webservice methods that have been created for this purpose are the following:

- ListDivisions (get a list with divisions)
- ListConsolidationCodes (get a list with consolidation codes)
- ListSalesInformation (get a list of information about sales)
- ListVisitingCustomersInformation (get a list of information about visiting customers)

29 Attachments

Attachments – Brief overview

As of service 3.4.0 is it possible to use attachments. It is possible to get information about attachment, the attachment itself, get attachment types and subtypes and save attachments.

GetAddressAttachment

Request

The request consists of a service context and a AddressAttachmentCriteria object.
The properties of an AddressAttachmentCriteria looks as follows:

AddressAttachmentCriteria

Name	Type	Description	Remarks
Id	Guid	Id of the attachment	

Response

The response will contain a AddressAttachment object and looks as follows:

Name	Type	Description	Remarks
Id	Guid	Id fo the attachment	
Attachment	Byte[]	The attachment in byte array	
FileName	String	The name of the attachment	

GetAddressAttachmentsInformation

Request

The request consists of a service context and a AddressAttachmentsInformationCriteria.
The properties of a AddressAttachmentsInformationCriteria looks as follows:

AddressAttachmentsInformationSearchCriteria

Name	Type	Description	Remarks
Id	Guid?	Unqiue identifier of the attachment	Optional
AddressID	Guid?	The address of the attachment	Optional
AddressAttachmentSubTypeID	Guid?	The id of the subtype	Optional
Paging	PagingCriteria		

Response

The response will contain a list of AddressAttachmentInformation objects that meet the criteria.
The AddressAttachmentInformation looks as follows:

Name	Type	Description	Remarks
Id	Guid	Unique identifier	
AddressId	Guid	Address	
FileName	String	Name of the attachment	
Comment	String	Comment	
AddressAttachmentSubTypeId	Guid	SubType	
AddressAttachmentSubTypeShortName	String	Shortname of the subtype	
AddressAttachmentSubTypeDescription	String	Description of the subtype	
AddressAttachmentTypeId	Guid	Type	
AddressAttachmentTypeShortName	String	Short name of the type	
AddressAttachmentTypeDescription	String	Description of the type	
DateUploaded	DateTime?	Date when attachment is uploaded	Is optional, only the attachments made in RCX5.9.0 or later will contain this field

GetAddressAttachmentSubTypes

Request

The request consists of a service context and a AddressAttachmentSubTypeCriteria object.
The properties of an AddressAttachmentSubTypeCriteria looks as follows:

AddressAttachmentSubTypeCriteria

Name	Type	Description	Remarks
Id	Guid?	Id of the subtype	
AddressAttachmentTypeID	Guid?	Id of the type	

Response

The response will contain a list of AddressAttachmentSubType objects that meet the criteria.
The **AddressAttachmentSubType** object looks as follows:

Name	Type	Description	Remark
Id	Guid	Id of the subtype	
Shortname	String	Short name of the subtype	
Description	String	Description of the subtype	
Email	String	Email of the subtype	
EmailMessageId	Guid	Id of the emailmessage	
AddressAttachmentTypeID	Guid	Id of the type	

GetAddressAttachmentTypes

Request

The request consists of a service context and a AddressAttachmentTypeCriteria object.
The properties of an AddrsesAttachmentTypeCriteria looks as follows:

AddressAttachmentTypeCriteria

Name	Type	Description	Remarks
Id	Guid?	Id of the type	Optional

Response

The response will contain a list of AddressAttachmentType objects that meet the criteria.
The **AddressAttachmentType** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	Id of the type	
ShortName	String	Short name of the type	
Description	String	Description of the type	

SaveAddressAttachment

Request

The request consists of a service context and a SaveAddressAttachmentOptions object.
The properties of a SaveAddressAttachmentOption object looks as follows:

Name	Type	Description	Remarks
PersonId	Guid	Id of the person	
FileName	String	Name of the file	
Attachment	Byte[]	The attachment in byte array	
Comment	String	Comment	
AddressAttachmentSubTypeId	Guid	The subtype of the attachment	

Response

The response will contain a SaveAddressAttachmentResult object and looks as follows:

Name	Type	Description	Remarks
ValidationResults	IList<ValidationResult>	If the save was no success, then this collection will contain details about the failure.	

30 App integration

The ApplicationConfiguration objects and corresponding methods are used to save and load the configurations of mobile apps.

LoadApplicationConfigurations

This method loads all ApplicationConfigurations matching the criteria.

Request

The request consists of a **ApplicationConfigurationSearchCriteria** object containing following properties:

Name	Type	Description
ApplicationIds	List<Guid>	A filter on ApplicationIds
ApplicationNames	List<string>	A filter on ApplicationNames

Response

The response consists a list of **ApplicationConfiguration** objects.

ApplicationConfiguration

Name	Type	Description
Id	Guid	The unique id of the ApplicationConfiguration
ApplicationId	Guid	The unique id of the Application
ApplicationName	string	The name of the application
Configuration	string	The configuration for the application
IsAdministratorConfig	bool	Is the administrator uploading this and is this the default configuration for all applications?

LoadApplications

This method loads all applications.

Request

The request contains no additional parameters next to the ServiceContext object.

Response

The response consists a list of **Application** objects

Application

Name	Type	Description
Id	Guid	The unique id of the application
Name	string	The name of the application

Log

This method is used to log data of a client application to a specific server location.

Request

The request contains a **ServiceContext** and a **LogCriteria** object.

LogCriteria

Name	Type	Description
ServerPath	String	The full server path to which the log should be written
Application	String	The application name to which the log applies

Data	String	The data to log

⚠ The full log path will be created as following: "{ServerPath}/{Application}/{ShopID}/Logging/{Date}.log"

Response

The response contains a **ValidationResult** object which indiciates if the Log was saved successfully.

Name	Type	Description	Remarks
BrokenRuleName	String	The code of the error message.	
Message	String	The standard error.	
IsValid	Bool	Indicates if this is a valid ValidationResult.	

SaveApplication

This method saves a new application if it does not yet exist.

The name of the application and the id must be unique.

Request

The request consists of a **Application** object containing following properties:

Name	Type	Description
Id	Guid	The unique id of the application
Name	string	The name of the application

Response

The response consists of a **ApplicationValidationResult** object (which inherits from the ValidationResult object) containing following properties:

Name	Type	Description
BrokenRuleName	string	If not valid, this property contains the reason why the SaveApplication was not valid.
IsValid	bool	If false, the Application is not saved. If true, the Application is saved.
Message	string	Additional information field
ApplicationId	Guid	The id of the saved Application.

SaveApplicationConfiguration

This method saves a new ApplicationConfiguration if it does not exist and updates an existing ApplicationConfiguration if it already exists.

Request

The request consists of a **ApplicationConfiguration** object containing following properties:

Name	Type	Description
Id	Guid	The unique id of the ApplicationConfiguration
ApplicationId	Guid	The unique id of the Application
ApplicationName	string	The name of the application
Configuration	string	The configuration for the application
IsAdministratorConfig	bool	Is the administrator uploading this and is this the default configuration for all applications?

Response

The response consists of a **ApplicationConfigurationValidationResult** object containing following properties:

Name	Type	Description
BrokenRuleName	string	If not valid, this property contains the reason why the SaveApplication was not valid.
IsValid	bool	If false, the Application is not saved. If true, the Application is saved.
Message	string	Additional information field
ApplicationConfigurationId	Guid	The id of the saved ApplicationConfiguration.

31 InfoPlay

FindAccessZones

Request

The request consists of a service context and a AccessZoneCriteria object.
The properties used for a AccessZoneCriteria object look as follows:

FindPersonsCriteria

Name	Type	Description	Remarks
Id	GUID?	Unique identifier of the access zone.	Nullable, not obligatory.
Includes	AccessZoneIncludes	Determines which additional details of the access zone will be included in the response.	
OccupancyFrom	DateTime?	Get the current visitors after a certain date	
OccupancyUntil	DateTime?	Get the current visitors before a certain date	

AccessZoneIncludes

This object allows to include certain parts of a access zone.

Name	Type	Description	Remarks
EntranceReaders	Bool	Include the linked entrance readers.	
ExitReaders	Bool	Include the linked exit readers.	
Occupancy	Bool	Include occupancy summary.	
InactiveZoneControl	Bool	Include the access zone with an inactive zone control	

Response

The response visualises a list of AccessZone objects that meet the AccessZoneCriteria .

AccessZone

Name	Type	Description	Remarks
Id	Guid	Unique identifier of the access zone.	
Code	String	Code of the access zone.	
Name	String	Name of the access zone.	
Number	int	Number of the access zone.	
Readers	IList<AccessZoneReader>	List of all linked readers (entrance / exit).	Can only contains values when EntranceReaders or ExitReaders in the AccessZoneIncludes is set to true.

Occupancy	AccessZoneOccupancy	Details of the occupancy	Object only exists when Occupancy int he AccessZoneIncludes is set to true.

AccessZoneReader

Name	Type	Description	Remarks
Reader	Reader	The reader details	See more info at GetReaders
Type	AccessZoneReaderType	The type of reader (entry or exit) => 0 = Entry / 1 = Exit.	

AccessZoneReaderType

AccessZoneReaderType is an enum type with two enum values.

- Entry = 0
- Exit = 1

AccessZoneOccupancy

Name	Type	Description	Remarks
MaxVisitors	int	Maximum number of visitors	
MaxVisitorsPerDay	int	Maximum number of visitors per day	
VisitorsToday	int	Number of visitors today	
VisitorsCurrent	int	Number of current visitors	

FindInfoPlayLogs

This method calls up all loggings of a subscription on the basis of certain search criteria.

It is equal to the FindSubscriptionLogs method, the only difference is that this is a lightweight version.

Request

The request consists of the service context and a **InfoPlayLogsSearchCriteria** object.

InfoPlayLogsSearchCriteria

Name	Type	Description	Remarks
SubscriptionLogId	GUID?	ID of a specific logging.	Optional
SubscriptionId	GUID?	ID of the subscription.	Optional
AccessZoneId	Guid?	Only the loggings of a specific access zone.	Optional
CreationDate	DateTime?	Only the loggings for this date.	Time is ignored
Paging	PagingCriteria	Paging criteria.	
Includes	InfoPlayLogsIncludes	Which data has to be included.	
ReaderIds	IList<Guid>	A list with unique identifier of the readers	

InfoPlayLogsIncludes

Name	Type	Description	Remarks
VisitorName	Bool	Include the visitor name	
VisitorId	Bool	Include the visitor id	
VisitorGender	Bool	Include the visitor gender.	
VisitorDateOfBirth	Bool	Include the visitor date of birth.	

SubscriptionName	Bool	Include the subscription name.	
ArticleId	Bool	Include the article id linked with the subscription	

Remark: When VisitorName, VisitorId, VisitorGender and VisitorDateOfBirth are false or not provided in the request. Then the InfoPlayVisitor object in the response will be null.

Response

The response will contain a list of **InfoPlayLog** objects that meet the criteria.

InfoPlayLog

The **InfoPlayLog** object looks as follows:

Name	Type	Description	Remarks
Id	GUID	ID of the logging.	
Creation Date	DateTime	Creation date of the logging.	
Visitor	InfoPlayVisitor	The customer linked with the SubscriptionLog.	Is null when VisitorName, VisitorId, VisitorGender and VisitorDateOfBirth are false or not provided in the request
Subscription	InfoPlaySubscription	The subscription linked with the SubscriptionLog.	
Reader	InfoPlayReader	The reader linked with the SubscriptionLog.	Is null when no reader is linked with the SubscriptionLog
IsValidEntry	bool?	Indiciates if this log is a valid entry	▪ Customer had no access -> false ▪ Customer had access -> true ▪ Subscription unlinked -> null ▪ Card swallowed -> null ▪ Subscription linked -> null

InfoPlayVisitor

The **InfoPlayVisitor** object looks as follows:

Name	Type	Description	Remarks
Id	Guid?	Id of the visitor	Contains a value when VisitorId is true in the includes of the request
Name	PersonName	The name of the visitor	Contains a value when VisitorName is true in the includes of the request
Gender	Gender?	The gender of the visitor	Contains a value when VisitorGender is true in the includes of the request
DateOfBirth	DateTime?	The datetime of the visitor	Contains a value when VisitorDateOfBirth is true in the includes of the request and when the visitor has a valid date of birth

InfoPlaySubscription

The **InfoPlaySubscription** object looks as follows:

Name	Type	Description	Remarks
Name	string	The name of the article linked with the subscription	Contains a value when SubscriptionName is true in the includes of the request
IsValid	Bool	Indicates if the subscription is valid or not	
ArticleId	Guid?	The id of the article linked with the subscription	Contains a value when ArticleId is true in the includes of the request

InfoPlayReader

The **InfoPlayReader** object looks as follows:

Name	Type	Description	Remarks
Id	Guid	The id of the reader	
Name	String	The description of the reader	

GetInfoPlayCustomData

This method is used to get the data of a view.

Request

The request consists of the service context and a InfoPlayCustomDataSearchCriteria object.

InfoPlayCustomDataSearchCriteria

Name	Type	Description	Remarks
SubjectId	Guid	ID of the subject of the view	

Response

The response will contain a list of **InfoPlayCustomDataRecordobjects** that meet the criteria.

InfoPlayCustomDataRecord

The **InfoPlayCustomDataRecord** object looks as follows:

Name	Type	Description	Remarks
Values	IDictionary<string, string>	KeyValuePair<name of the field, value of the field>	

GetInfoPlayCustomFields

This method is used to get the fields of a view.

Request

The request consists of the service context and a **InfoPlayCustomFieldSearchCriteria** object.

Right now this object contains no properties but it's provided for future use.

Response

The response will contain a list of **InfoPlayCustomField** objects that meet the criteria.

InfoPlayCustomField

The **InfoPlayCustomField** object looks as follows:

Name	Type	Description	Remarks
Name	string	The name of the field	

32 UiTPAS

BuyUitpasEvent

This method processes the payment of Articles/Activities combined with the UiTPAS.

This method is implemented in the CheckOutBasket.

Request

The request consists of the service context and a **BuyUitpasEventCriteria** object.

BuyUitpasEventCriteria

Name	Type	Description	Remarks

UitpasNumber	string	The number of the UitPAS card.	
UitpasEventId	string	The ID of the UiTPAS event	
PriceGroup	string	The pricegroup that must be used	This is the property 'PriceGroupToUse' in the **SearchUitpasTariffResult** object.

The response will contain a **BuyUitpasEventResult** object that meet the criteria.

BuyUitpasEventResult

Name	Type	Description	Remarks
IsSuccess	Bool	Indicates if the payment has succeeded	
Price	decimal	The price that has been used.	

SearchUitpasTariff

This method finds the tariff for a certain combination of UiTPASNUMBER and UiTPASEventID

Request

The request consists of the service context and a SearchUitpasTariffCriteriaobject.

SearchUitpasTariffCriteria

Name	Type	Description	Remarks
PersonId	Guid	For which customer is the article?	
UitpasNumber	string	The number of the UitPAS card.	
UitpasEventId	string	The ID of the UiTPAS event	

The response will contain a **SearchUitpasTariffResult** object that meet the criteria.

SearchUitpasTariffResult

Name	Type	Description	Remarks
IsValid	Bool	Indicates if a tariff may be used for the data in the criteria object	
Price	decimal	If the 'IsValid' property is true, then this property will contain the price/tariff that must be used	
PriceGroupToUse	string	If the 'IsValid' property is true, then this property will contain a price group that must be used to finish the payment.	

33 OTA service

The OTA (Over The Air) Service is a separate service which is used to get update information and download updates for mobile applications.
The OTA Service is available at following URI: [BaseUrl]/otaservice.svc

Download

This method downloads the update for a certain package.
= GET

Request

The request is performed by using following URI: [BaseUrl]/otaservice.svc/{pPackageName}/{pApkName}

Name	Type	Description
pPackageName	String	The unique name of the package
pApkName	String	The unique name of the apk

Response

The response is a Stream object containing the package update.

GetAppUpdate

This method verifies if an update is available.
= POST

Request

The request consists of a **GetAppUpdateRequest** object containing following properties:

Name	Type	Description
PackageName	String	The unique name of the package
Version	Int	The version of the package

Response

The response consists of a **GetAppUpdateResponse** object containing following properties:

Name	Type	Description
IsUpdateAvailable	Bool	Indicates whether an update is available
ApkUrl	String	The Url of the apk which can be used to download the update from

34 Samples

Basket state

In the Webservices a state isn't kept anywhere. This means that the client application that uses the Webservices is responsible for maintaining the basket and its items.

This is mainly required between paying and posting the basket.
Since all objects are passed through the Webservices, all objects can be perfectly serialized in XML format.
As a consequence, this is the perfect way to keep objects in for example a customer database.

Helper classes

The helper class below can be used to quickly serialize and deserialize objects from and to XML.

```
public static class XML
{
        /// <summary>
        /// Serialize an object into an XML string
        /// </summary>
        /// <typeparam name='T'></typeparam>
        /// <param name='obj'></param>
        /// <returns></returns>
        public static string SerializeObject<T>(T obj)
        {
                string serialXML = String.Empty;
                try
                {
                        // Serialize the object
                        XmlSerializer xmlSerializer = new XmlSerializer
(obj.GetType());
                        using (CustomStringWriter stringWriter = new
CustomStringWriter())
```

```
                        {
                                xmlSerializer.Serialize(stringWriter,
obj);
                                stringWriter.Flush();
                                serialXML = stringWriter.ToString();
                        }
                }
                catch { }

                return serialXML;
        }


        /// <summary>
        /// Reconstruct an object from an XML string
        /// </summary>
        /// <param name='xml'></param>
        /// <returns></returns>
        public static T DeserializeObject<T>(string xml)
        {
                using (StringReader stringReader = new StringReader
(xml))
                {
                        XmlSerializer serializer = new XmlSerializer
(typeof(T));
                        return (T)serializer.Deserialize(stringReader);
                }
                return default(T);
        }
}

public class CustomStringWriter : StringWriter
{
        private Encoding encoding = System.Text.Encoding.UTF8;

        public CustomStringWriter() : base()
        {
        }

        public CustomStringWriter(Encoding encoding) : base()
        {
                this.encoding = encoding;
        }

        public override Encoding Encoding
        {
                get
                {
                        return encoding;
                }
        }
}
```

Statebag

The helper classes above allow to make a type of StateBag in which all required objects are kept.

```
public class StateBag
{
        public Webshop.Basket Basket { get; set; }
        public Webshop.BasketValidationResult ValidationResult { get;
set; }
        public Webshop.CheckoutBasketResult CheckOutResult { get; set;
}
        public void Save(string fileName)
        {
                string xml = XML.SerializeObject(this);
                File.WriteAllText(fileName, xml);
        }
        public static StateBag Load(string fileName)
        {
                string xml = File.ReadAllText(fileName);
                return XML.DeserializeObject<StateBag>(xml);
        }
}
```

This statebag is only a demo example.
In this code the basket is kept as wel as the result of a ValidateBasket and the CheckoutBasket.
On the basis of the Save and Load a Statebag can be saved under a file on the hard disk.
Dependent on the client application's needs, this can be modified to save the Statebag in, for example, a database.

```
// Recalculate price
Webshop.Basket basket = new Webshop.Basket();
basket.CustomerId = persons[0].Id;
basket.Items = lockBasketResult.BasketItems;
webshopService.ReCalculateBasket(ref basket, context);

// Keep the status of the shopping basket
StateBag statebag = new StateBag { Basket = basket };
statebag.Save( fileName );
```

This can be required for examining problems in the client code or in the ReCreateX back-office application when differences occur between the posted items in the Webservices and the final result in the back office.
It is advised to keep the Statebag of all sales that are done on the Website for a period of, for example, a month. This allows us to examine what was booked through the Webservices and what is present in ReCreateX.

Creation context item

Each Webshop service call uses a service context object in which a language, division, WebShopID and Origin can be specified.

```
Webshop.ServiceContext context = new Webshop.ServiceContext();
context.Language = 'FR';
context.ShopID = '00000000-0000-0000-0000-000000000001';
```

The language code that can be filled in is especially used to determine in which language code the translations have to be called up.
This is the ISO code of the language.

In most cases, client applications are first developed in a test environment before they are installed in a production environment. The configuration of the Webservices allow to simply switch between a test environment and a production environment. This is done on the basis of the ShopID.
In the Webshop configuration database, each ShopID refers to a configuration in which all settings are kept. Database, user, point of sale, division, ...
The only thing the service consumer has to do is set the ShopID in the context object to the ID of the correct environment. These ID's are attributed by Syx Automations at installation and will not be modified afterwards.
Example:

```
//context.ShopID = '00000000-0000-0000-0000-000000000001'; // Test
environment
//context.ShopID = '00000000-0000-0000-0000-000000000002'; // Production
```

The DivisionId is automatically filled in by the Webservice call if the ShopID has been filled in correctly.
The Origin defines what device is connected to the web services. There are 4 types to choose: Default, Internet, VendingMachine & Desktop.

This context object has to be created once in the client environment and can afterwards be given at each service call.

Expositions

The code below is a code developed for finding, for example, a number of objects concerning expositions and for posting an arbitrary exposition reservation.

Find the customers

By way of illustration, all customers all called up. Afterwards, the first customer will be used to post the reservation.

```
// find customers
Webshop.Person[] persons = webshopService.FindPerson(context,
        new Webshop.FindPersonsCriteria
        {
                Includes = new Webshop.PersonIncludes {
                        Categories = true,
                        CategoriesSpecified = true
                },
        });
```

Find expositions, periods and prices

In order to be able to make a reservation for an exposition, we need an exposition and an exposition period.
The function below calls up all expositions and finds the first period after today for which there is also a valid price.

```
private bool FindFirstExpositionPeriod(out Webshop.Exposition
exposition, out Webshop.ExpositionPeriod expositionPeriod, out Webshop.
Price price)
```

```
{
        exposition = null;
        expositionPeriod = null;
        price = null;
        Webshop.Exposition[] expositions = webshopService.
FindExpositions(context,
                new Webshop.ExpositionSearchCriteria {
                        Includes = new Webshop.ExpositionIncludes {
Image = false, ImageUri = false },
                        From = DateTime.Now,
                        Paging = new Webshop.PagingCriteria { PageIndex
= 0, PageSize = 100 }
                });

        foreach (Webshop.Exposition _exposition in expositions)
        {
                if (_exposition.Periods == null || _exposition.Periods.
Length == 0)
                {
                        _exposition.Periods = webshopService.
ListExpositionPeriods(context,
                                new Webshop.
ExpositionPeriodSearchCriteria {
                                        ExpositionId = _exposition.Id,
                                        From = DateTime.Now
                                });
                }

                if (_exposition.Prices != null && _exposition.Prices.
Length > 0)
                {
                        price = _exposition.Prices.First(p => p.
AmountSpecified && p.Amount > 0);
                        if (price == null)
                                continue;
                }


                if (_exposition.Periods != null && _exposition.Periods.
Length > 0)
                {
                        expositionPeriod = _exposition.Periods.First();
                        exposition = _exposition;
                        break;
                }
        }


        return (expositionPeriod != null);
}
```

This function can then be used in our demo code to find an exposition, price and period so that we can simply create an arbitrary reservation.

```
// Find an arbitrary exposition that has a valid period and price
Webshop.Exposition exposition;
Webshop.ExpositionPeriod expositionPeriod;
Webshop.Price price;

FindFirstExpositionPeriod(out exposition, out expositionPeriod, out
price);

if (exposition == null)
        throw new ApplicationException('exposition == null');
if (expositionPeriod == null)
        throw new ApplicationException('expositionPeriod == null');
```

Create reservation object

On the basis of the exposition data found, the code below will create 1 new reservation object and a lock ticket that is valid for 1 hour.
The reservation contains a reservation for 5 persons of the given price group.
The lock ticket assures that a dummy reservation will automatically be created in the LockBasketItems, so that it is already taken into acount when counting the remaining places.

```
// Create reservation for 5 participants
Webshop.ExpositionPeriodReservation reservation = new Webshop.
ExpositionPeriodReservation
{
        ExpositionPeriodId = expositionPeriod.Id,
        Entries = new Webshop.ExpositionPeriodReservationEntry[]
        {
                new Webshop.ExpositionPeriodReservationEntry
                {
                        ParticipantCount = 5, ParticipantCountSpecified
= true,
                        PriceGroupId = price.Group.Id
                }
        },
        LockTicket = new Webshop.ExpositionReservationLockTicket
        {
                ExpirationTime = DateTime.Now.AddHours(1)
        },
        Quantity = 1
};
```

Add reservation to shopping basket

We have to be able to keep the reservation object created above in a list, so that we can afterwards find back our object.
All objects that can be bought on the Website, inherit of a basketItem.
We can consequently keep up a list of BasketItems using the following code:

```
// Add reservation to list
List<Webshop.BasketItem> basketItems = new List<Webshop.BasketItem>();
basketItems.Add(reservation);
```

Create temporary reservation in the database

After the creation of our reservation object in code, we still have to create it in the database, so that it is taken into account when counting the available places and so that no one else can take away our reservation.
The logic is the same for all basket items that can be locked.
Here, we only have add our list of BasketItems to the LockBasketItems function.

```
// Lock list / reserve
Webshop.LockBasketResult lockBasketResult = webshopService.
LockBasketItems(basketItems.ToArray(), context);
if (lockBasketResult.IsLocked == false)
        throw new ApplicationException('lock could not be set');
```

The LockBasketItems function returns a LockBasketResult.

In this LockBasketResult the full list of articles that have been locked can be found within the BasketItems.
For the further processing of the Webservices, it is important that this new list of articles is used, and not the original list of articles we have added to the LockBasketItems.
This is necessary because the LockBasketItems already creates the reservations in the database in the form of temporary reservations. The reference to the temporary reservations is returned to you within each BasketItem in the LockBasketResult. This is not filled in in the original list of BasketItems.

Mark that our temporary reservation is only valid for one hour. We have already defined this with the ExpirationTime in the lock ticket we have created earlier.
If this time has elapsed, the reservation will automatically be deleted.
This can happen if the user, for example, has closed the browser without having settled the shopping basket.
If the expiration time has to be prolonged, you can do so using the ExtendLockPeriod function.
This is why it is important that all objects in the shopping basket on the client side are also kept in some sort of StateBag, since no State is kept on the service side.

Moreover, it is possible to test if all reservations have effectively been locked.
We can do so by checking if the LockTicket contains a valid ID.

```
if (lockBasketResult.BasketItems.Any(i => (i is Webshop.
ExpositionPeriodReservation) && String.Equals((i as Webshop.
ExpositionPeriodReservation).LockTicket.Id, Guid.Empty.ToString())))
        throw new ApplicationException('lock could not be set');
```

Recalculate the price of the shopping basket

If the reservation has been created in the database, the total price of the shopping basket can be recalculated.
This in necessary because it is only now that the customer is linked to the shopping basket.
Because the customer is linked only now, the price may change.
Mark as well that we use the BasketItems of the LockBasketResult.
After the recalculation of the price, the entire basket is saved under a file on the hard disk. This is purely for demo purposes. In real-world applications, it is better that these are saved in a database.

```
// Recalculate price
Webshop.Basket basket = new Webshop.Basket();
basket.CustomerId = persons[0].Id;
basket.Items = lockBasketResult.BasketItems;
webshopService.ReCalculateBasket(ref basket, context);

// Keep the status of the shopping basket
StateBag statebag = new StateBag { Basket = basket };
statebag.Save(Path.Combine(Path.GetTempPath(), String.Format('{0}.xml',
'State 1 - ' + fileguid)));
```

Create payments

If the total price of the shopping basket is known, we can create the payment. Payment is necessary because the total price of the shopping basket has to be equal to the total price of the purchases at the final posting of our reservation.
In the code example below, the entire list of payment methods is first called up.
Afterwards, the payment method 'CASH' is searched for purely for demo purposes.

```
 // add payment
Webshop.BasketPaymentMethod[] paymentMethods = webshopService.
ListPaymentMethods(context);
Webshop.BasketPaymentMethod contant = paymentMethods.First(p => p.Code
== 'CASH');

if (contant == null)
        throw new ApplicationException('Payment method cash not found');
```

A shopping basket can contain different payment methods in 1 payment.
If the total value is for example 100 EUR, 80 EUR can be paid in cash and 20 EUR with VISA. This is, however, outside the scope of the example.
In the example below the total amount of the Basket (basket.Price) is paid using the payment method 'CASH' that we called up above.

```
basket.Payments = new Webshop.BasketPayment[]
{
        new Webshop.BasketPayment
        {
                Amount = basket.Price,
                Currency = '€',
                PaymentMethodId = contant.Id
        }
};
```

Validation

The last step that has to be executed before we can post our shopping basket, is validation. Here, it is checked for a last time if everything is in order before we effectively pay and post.

In order to check possible disputes or problems afterwards, is it advisable to also save the result of the ValidationBasket in the StateBag. In this way, it is possible to check afterwards which was the content of the shopping basket and what was the result of the validation. Hence the ValidationResult is also included in the StateBag in this example.

```
// Validation of the shopping basket
statebag.ValidationResult = webshopService.ValidateBasket(basket,
context);
statebag.Save(Path.Combine(Path.GetTempPath(), String.Format('{0}.xml',
'State 2 - ' + fileguid)));
bool isValid = true;

foreach (Webshop.BasketItemValidationResult result in statebag.
ValidationResult.BasketItemValidationResults)
{
        if (!result.IsValid)
        {
                Console.WriteLine('validation failed');
                Console.WriteLine(result.brokenRuleName);
                Console.WriteLine(result.Message);
                isValid = false;
        }
}

if (!isValid || statebag.ValidationResult.IsValid == false)
        throw new ApplicationException('Validation failed');
```

Post

If validation succeeded, the shopping basket can be posted by means of which all temporary reservations are turned into definitive reservations.
If payment is done with an online payment, it is naturally supposed that payment has already been made at this moment. Payment has to be done after validation and before posting.
If the online payment is not successful, the temporary reservations are automatically deleted from the database once the lock tickets expire.
Here as well, the result of the check-out is saved in the StateBag.

```
// CHECKOUT
statebag.CheckOutResult = webshopService.CheckoutBasket(statebag.
Basket, context);
statebag.Save(Path.Combine(Path.GetTempPath(), String.Format('{0}.xml',
'State 3 - ' + fileguid)));
```

Find ETickets

In case of expositions, it is possible to sell access tickets together with the reservation.
These access tickets can also be called up with Webservices in the form of a PDF document.
These Webservices are not classical SOAP services but REST services in case of which documents can be downloaded through a stream.
If implemented on a Website, only a dynamic URL has to be set up. If the user is redirected to this dynamic URL, he automatically gets a PDF document.
In a classic Windows application, this can be downloaded using the following code, in which case the URL and the file name are given via parameters.
In addition, a timeout can be added.

```
private static bool SavePdfFileFromURL(string url, string
destinationFileName, int timeoutInSeconds)
{
        // Create a web request to the URL
        HttpWebRequest MyRequest = (HttpWebRequest)WebRequest.Create
(url);
        MyRequest.Timeout = timeoutInSeconds * 1000;


        try
        {
                // Get the web response
                HttpWebResponse MyResponse = (HttpWebResponse)MyRequest.
GetResponse();
                // Make sure the response is valid
                if (HttpStatusCode.OK == MyResponse.StatusCode)
                {
                        // Open the response stream
                        using (Stream MyResponseStream = MyResponse.
GetResponseStream())
                        {
                                // Open the destination file
                                using (FileStream MyFileStream = new
FileStream(destinationFileName, FileMode.OpenOrCreate, FileAccess.
Write))
                                {
                                        byte[] MyBuffer = new byte
[4096];
                                        int BytesRead;
                                        // Read the chunk of the web
response into the buffer
                                        while (0 < (BytesRead =
MyResponseStream.Read(MyBuffer, 0, MyBuffer.Length)))
                                        {
                                                // Write the chunk from
the buffer to the file
                                                MyFileStream.Write
(MyBuffer, 0, BytesRead);
                                        }
                                        MyFileStream.Flush();
                                        MyFileStream.Close();
                                }
                        }
                }
        }
        catch (Exception err)
        {
                throw new Exception('Error saving file from URL:' + err.
Message, err);
        }
        return true;
}
```

The URL under which the PDF document can be called up consists of some fixed and variable parts:

- Host name: is usually the same location as the one on which the other Webservices are deployed. This is communicated by Syx Automations at deployment of the Webservices.
- Path: Fixed part: WebShopDocumentService.svc/ETickets Variable part: ShopID / Taal /. SalesLineID

The SalesLineID is arbitrary SalesLineID from the sale, since the system checks all sales lines of the same sale to see if ETickets are linked. This ID can be called up from the SalesItems of the CheckOutResult.
Example:

```
string url = String.Format('http://{0}/WebShopDocumentService.svc
/ETickets/{1}/{2}/{3}', webshopService.Endpoint.Address.Uri.Authority,
context.ShopID, context.Language, salesLineId);
string salesLineId = statebag.CheckOutResult.SalesItems.First().Id;
string fileName = Path.Combine(Path.GetTempPath(), String.Format('{0}.
pdf', Guid.NewGuid()));
SavePdfFileFromURL(url, fileName, 60);
```

The final result is a PDF file, having a new unique identifier as a name, so that the name remains unique.