ULTIMATE®

Webservice reference of Ultimate enterprise management software

ver. 5.7.0

www.ultimatebusinessware.com

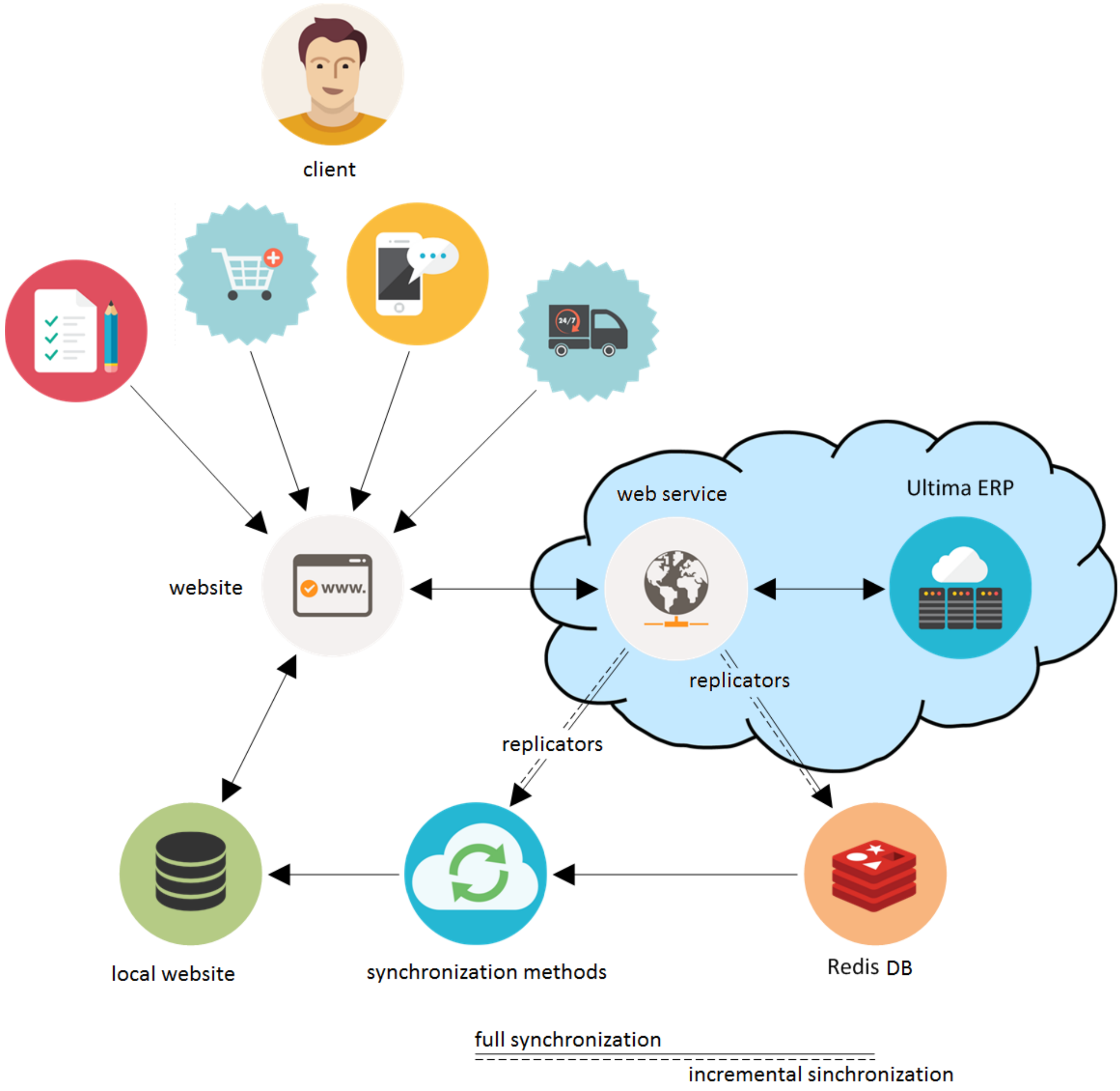
Table of Contents:

Web service

Intro

This document is intended for application-oriented developers and will help to realize own option of integration of a website/online shop of the company and Ultimate e-Trade.

General model of interaction of the website and Ultimate e-Trade is given in the diagram:



From the provided diagram of interaction it is visible that the website has no access to the database of the company, and all interaction is carried out only through web service.

The main part of the document is made by the description of all methods of web service of the configuration Ultimate e-Trade. At the same time, methods are divided into groups according to their functional purpose: methods of Authentication, operation with Clients, creations of Reserves of articles under clients' orders, etc.

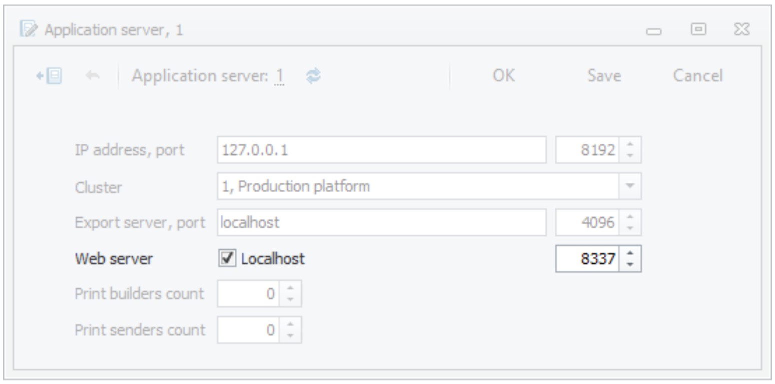
In separate group methods-replicators are selected, which task is to copy necessary data from the database of the system to the local website DB. Data replication using these methods is carried out both directly and in some cases through an intermediate database Redis. Redis is usually used for synchronization of big directories (from 20'000 commodity lines items and more), but depending on a specific situation, it can be needed earlier. Some methods-replicators support incremental synchronization of data, however for this purpose it is necessary to remember time of their last successful call.

The following entities are figured in the description of the web service methods:

- *Client* (client) – physical person working with the website;
- *Agent* (agent) – legal entity, who is affiliated with the client-physical person. One client can have several agents-legal entities. The client, having passed authorization on the website, can work as directly as himself, and so as one of his legal entities;
- *Article*, *Product* (article) – realized by means of the website of the company;
- *Category* (category) – category of articles, totality of which makes the article directory of the company;
- *Product property* (product property) – property (one of), describing the product;
- *Reserve* (reserve, order) – order document created upon purchase on the website.

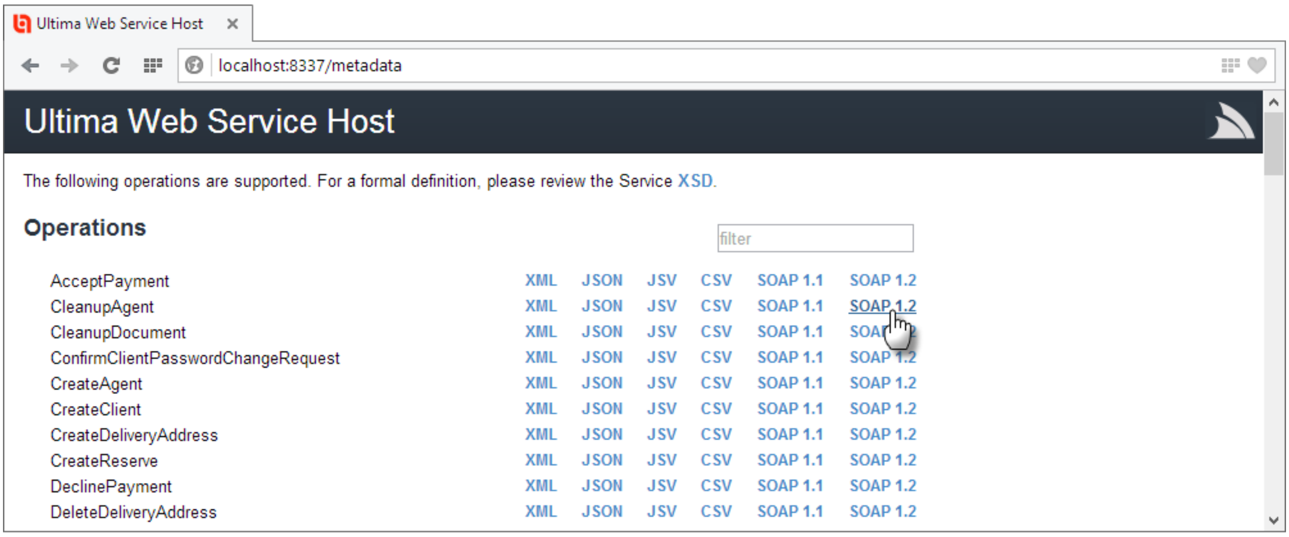
Web service

In addition to public web service, for example, http://192.168.0.28:8437/metadata, it can be also run locally. For this purpose, in application server configuration it is necessary to specify *Localhost* as its address:



IP address, port	127.0.0.1	8192
Cluster	1, Production platform	
Export server, port	localhost	4096
Web server	☑ Localhost	8337
Print builders count	0	
Print senders count	0	

Web developer can receive a list of the available methods of web service to its address http://localhost:8337/metadata:



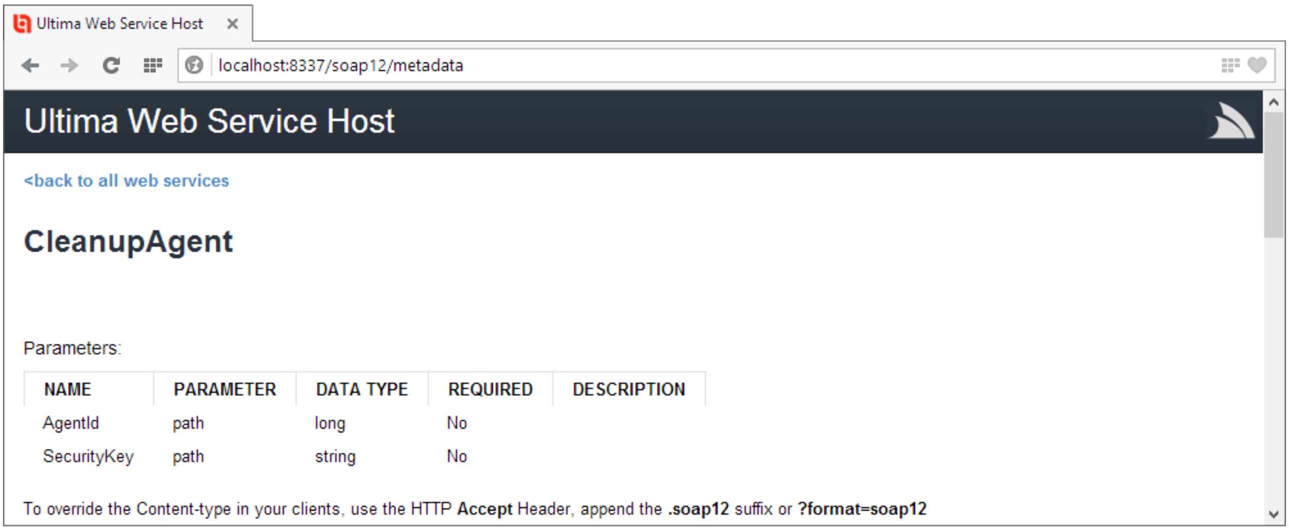
Ultima Web Service Host

The following operations are supported. For a formal definition, please review the Service XSD.

Operations

AcceptPayment	XML	JSON	JSV	CSV	SOAP 1.1	SOAP 1.2
CleanupAgent	XML	JSON	JSV	CSV	SOAP 1.1	SOAP 1.2
CleanupDocument	XML	JSON	JSV	CSV	SOAP 1.1	SOAP 1.2
ConfirmClientPasswordChangeRequest	XML	JSON	JSV	CSV	SOAP 1.1	SOAP 1.2
CreateAgent	XML	JSON	JSV	CSV	SOAP 1.1	SOAP 1.2
CreateClient	XML	JSON	JSV	CSV	SOAP 1.1	SOAP 1.2
CreateDeliveryAddress	XML	JSON	JSV	CSV	SOAP 1.1	SOAP 1.2
CreateReserve	XML	JSON	JSV	CSV	SOAP 1.1	SOAP 1.2
DeclinePayment	XML	JSON	JSV	CSV	SOAP 1.1	SOAP 1.2
DeleteDeliveryAddress	XML	JSON	JSV	CSV	SOAP 1.1	SOAP 1.2

When choosing a protocol of the appropriate method an example of a call is shown:



Ultima Web Service Host

<back to all web services

CleanupAgent

Parameters:

NAME	PARAMETER	DATA TYPE	REQUIRED	DESCRIPTION
AgentId	path	long	No	
SecurityKey	path	string	No	

To override the Content-type in your clients, use the HTTP **Accept** Header, append the **.soap12** suffix or **?format=soap12**

Recommended method of operation – JSON.

Text of exception is specified together with HTTP code of response state in the heading.

Errors

Extended information on errors is transferred to the client of web service in HTTP headers *UltimaErrorCode* and *UltimaErrorText*. The first is an error code, the second is its text.

Codes 100-199 are reserved for errors of the basic decision. For example, code 100 is the error of client authentication.

To return extended information on the error, it is necessary to use the class *ExtendedHttpError*:

```
thLine new ExtendedHttpError(1234, "Error description")
```

Exception *ExtendedHttpError* is intended for use only and directly in the code of web services.

Use it in other places, as a rule, means logic doubling – a situation when data processing procedures differ for web and not for web. It should be avoided carefully. While meeting the requirements the exception will not be required to be thLinen out except the code of web services anywhere.

For processing of the extended information on web service errors clients of ServiceStack should obtain data of HTTP headers from the exception:

```
try
{
        var request = new GetNow();
        var response = client.Send(request);
}
catch (WebServiceException e)
{
        var errorCode = e.ResponseHeaders.GetValues("UltimaErrorCode");
        var errorText = e.ResponseHeaders.GetValues("UltimaErrorText");
}
```

Authorization

Access to web services is limited and is provided only to authorized users.

Such limitation is realized for the purpose of safety that different applications through the appropriate users have access only to their web services. In this case, the compromise of one application will not be able to influence on other applications which are also using web services.

Authorization is required for all web service methods. For authorization HTTP Basic authentication is used.

In case of unsuccessful or incorrect authentication of web service user the HTTP code ***#401*** is returned (Unauthorized).

In case of rights absence of the user on the method HTTP code ***#403*** is returned (Forbidden).

To not pass the authentication procedure constantly, it is necessary to save cookie "ss-id", transferred by web service, and transfer it in case of requests.

To use authorization in SoapUI (⇨ http://www.soapui.org/), it is necessary to set *Authorization: Basic*, add *Username* and *Password* and select *Authenticate pre-emptively*.

However, authorization does not work with Soap 1.1/Soap 1.2/Xml the clients of ServiceStack.

Redis

Address of the Redis database: **192.168.0.20**

The standard response in Redis methods: `string[] keys` – the Redis key list.

Each record in Redis contains the record list of the required service.

Lifetime of record in Redis is one hour. Base practice is independent deleting the unloaded data.

Methods

In the description of methods their parameters which are specified in a format [*Type*] [*Name*] – [*Description*], are subdivided into the following appropriately selected categories:

- **mandatory**;
- normal;
- *unrealized*.

Registration

There two variants for registration in the system:

- regular on-site registration;
- registration via social networks. If a user name with the Email specified already exists, he shall be sent a message confirming consent to link his social network account.

Create agent

Create a new agent.

Request parameters:

- **string Name** – name;

- **string Inn** – INN;
- **string Kpp** – KPP;
- **string Okpo** – OKPO;
- **string SettlementAccount** – Bank account;
- **string Bic** – BIC;
- **string BankName** – bank name;
- **string BankCorrAccount** – bank correspondent account;
- string Email – e-mail.

Response parameters:

- long Id – code of created agent.

Authentication

After successful passing of authentication by the client, information on it is saved in the current session of web service. The client of web service needs to save cookie to continue request within this session (see Authorization of web service methods).

If in case of request to the method requiring the authenticated user, the session is not marked as authenticated, the error # 100 is returned.

The previous version with SecurityKey is not used any more.

SignInClientWithEmail

Sign In Client With Email.

Query parameters:

- **string Email** – Email;
- **string Password** – password.

Response parameters:

- bool Success – result.

SignInClientWithSocial

Sign In Client With Social.

Query parameters:

- **string SocialId** – OAuth token;
- **string SocialCode** – OAuth type (‘facebook’, ...).

Response parameters:

- bool Success – result.

Password recovery

Recovery of the password is carried out in two stages:

- at first by method *SendClientPasswordChangeRequest* a link to the website is sent to client's email for password recovery;
- then by method *ConfirmClientPasswordChangeRequest* password change is confirmed.

SendClientPasswordChangeRequest

Send Client Password Change Request. The reference is in the format ...?h=..., e.g.: http://www.ultimaerp.com/confirm_change/?h=i2g34u2b34uh234eu. The part following the "?h=" is at the same time a *Hash* parameter of *ConfirmClientPasswordChangeRequest* method.

Query parameters:
- **string Email** – e-mail.

Response parameters:
- bool Success – result;

ConfirmClientPasswordChangeRequest

Confirmation of client password change request.

Request parameters:
- **string Hash** – a line (part of a link), sent to the Email of the client for confirmation of client password change request. Sent link has a format ...?h=.... *Hash* – is a part that follows after "?h=". For example, for the link http://www.ultimaerp.com/confirm_change/?h=**i2g34u2b34uh234eu** it will be a line "*i2g34u2b34uh234eu*";
- **string Password** – password;

Response parameters:
- bool Success – result.

Clients

IsClientExists

Checking of client existence.

Request parameters:
- **string Email** – Email.

Response parameters:
- bool Exists – client exists.

GetClientInfo

Get client information.

This method does not have query parameters.

Response parameters:
- long Id – a client code;
- string Email – client e-mail;
- string Phone – phone;
- string FirstName – first name;
- string MiddleName – middle name;
- string LastName – last name.

GetClientBalance

Get client balance value.

This method does not have query parameters.

Response parameters:
- decimal Value – balance.

GetClientBonusBalance

Get client bonus balance value.

This method does not have query parameters.

Response parameters:
- decimal Value – bonus balance.

UpdateClientInfo

Update Client Info. If new query parameter value is *null* or “”, its value is not updated.

Request parameters:
- string FirstName – first name;
- string MiddleName – middle name;
- string LastName – last name;
- string Phone – phone;

Response parameters:
- bool Success – result.

GetClientOrdersStatistics

Get client order statistics. Information is available by the following order group:
- reserves – reserves;
- shipping – delivering/shipping;
- shipped – delivered/ shipped.

This method does not have query parameters.

Response parameters:
- []
  - string GroupKey – order group (from the list above-mentioned);
  - long Count – order quantity in group.

Social networks

GetClientSocialNetworkAccounts

Get a list of client social network account.

This method does not have query parameters.

Response parameters:

- []
  - string SocialCode – OAuth type (‘facebook’, ...).

LinkClientSocialNetworkAccount

Binding of social network account to the current client.

Request parameters:

- string SocialId – OAuth token;
- string SocialCode – OAuth type (‘facebook’, ...).

Response parameters:

- bool Success – result.

UnlinkClientSocialNetworkAccount

Unlink of social network account from the current client.

Request parameters:

- string SocialCode – OAuth type (‘facebook’, ...).

Response parameters:

- bool Success – result.

Agents

CreateAgent

Create a new agent.

Request parameters:

- **string Name** – name;
- **string Inn** – INN;
- **string Kpp** – KPP;
- **string Okpo** – OKPO;
- **string SettlementAccount** – Bank account;
- **string Bic** – BIC;
- **string BankName** – bank name;
- **string BankCorrAccount** – bank correspondent account;
- string Email – e-mail.

Response parameters:

- long Id – code of created agent.

GetAgentsBalance

Get agent balance values.

Query parameters:

- long[] Ids – the agent code list. If codes are not specified, all agent balance is requested.

Response parameters:

- []
    - long Id – an agent code;
    - decimal Value – balance.

GetClientAgents

Get the client agent list.

Query parameters:

- **bool IncludeBalance** – to include information on agent balance in the response.

Response parameters:

- []
    - long Id – an agent code;
    - string Name – name;
    - string Inn – TIN;
    - string Kpp – RRC;
    - string Okpo – RNCBO.
    - string SettlementAccount – Bank account;
    - string Bic – RCBIC;
    - string BankName – bank name;
    - string BankCorrAccount – correspondent account;
    - string Email – e-mail;
    - decimal BalanceAmount – agent balance;
    - long PriceCategoryId – a price category code;
    - decimal? MaxReserveAmount - maximum reserve amount.

LinkAgentToClient

Binding of specified agent to the current client.

Request parameters:

- **string Id** – agent code.

Response parameters:

- bool Success – result.

UnlinkAgentFromClient

Unlink of specified agent from the current client.

Request parameters:

- **string Id** – agent code.

Response parameters:

- bool Success – result.

GetAgentsWithDetails

Get agents with details.

Query parameters:

- **string Inn** – TIN;
- **string Kpp** – RRC;
- **string Okpo** – RNCBO.

Response parameters:

- []
  - long Id – an agent code;
  - string Name – name;
  - string Inn – TIN;
  - string Kpp – RRC;
  - string Okpo – RNCBO.
  - string SettlementAccount – Bank account;
  - string Bic – RCBIC;
  - string BankName – bank name;
  - string BankCorrAccount – correspondent account;
  - string Email – e-mail.

GetAgentsReservesAmount

Get agent reserves amount values.

This method does not have query parameters.

Response parameters:

- long Id – an agent code;
- decimal Value – reserve amount.

UpdateAgentDetails

Update Agent Details. If new query parameter value is *null* or “”, its value is not updated.

Request parameters:

- **string Id** – agent code;
- string Name – name;
- string Inn – INN;
- string Kpp – KPP;
- string Okpo – OKPO;
- string SettlementAccount – Bank account;
- string Bic – BIC;
- string BankName – bank name;
- string BankCorrAccount – bank correspondent account;
- string Email – e-mail.

Response parameters:

- bool Success – result.

Deliveries

CreateDeliveryAddress

Creation of delivery address. If the agent is not specified, address is created for the client.

Request parameters:

- **string Address** – address;
- **decimal Latitude** – latitude;
- **decimal Longitude** – longitude;
- long? AgentId – agent code.

Response parameters:

- long Id – code of created delivery address.

UpdateDeliveryAddress

Updating of delivery address. If new query parameter value is *null* or “”, its value is not updated.

Request parameters:

- **iong Id** – delivery address code;
- long? AgentId – agent code;
- string Address – address;
- decimal Latitude – latitude;
- decimal Longitude – longitude.

Response parameters:

- bool Success – result.

DeleteDeliveryAddress

Deleting of delivery address.

Request parameters:

- **iong Id** – delivery address code;
- long? AgentId – agent code.

Response parameters:

- bool Success – result.

GetDeliveryAddresses

Get delivery addresses. If the agent is not specified, the response will contain addresses of client deliveries.

Query parameters:

- long? AgentId – agent code.

Response parameters:

- []
  - long Id – delivery addresses;
  - string Address – address.
  - decimal Latitude – latitude;
  - decimal Longitude – longitude.

GetDeliveryCost

Get delivery cost.

Query parameters:

- long? AgentId – agent code;
- **long ReserveId** – a reserve code;
- **object Delivery** – delivery:
  - **long AddressId** – a delivery address code;
  - **long TimeId** – a delivery time code;
  - **DateTime Date** – delivery date;
  - **string Option** – delivery parameters. Can have values:
    - own – own;
    - outsource – third-party logistic company;
    - hybrid – own to the logistic company and further by the logistic company;
  - long? LogisticCompanyId – a code of the logistic company;
  - decimal? LogisticCompanyCost – delivery cost by the logistic company.

Response parameters:

- decimal Value – delivery cost.

Reserves

CreateReserve

Creation of the document reserving article by the created order.

Request parameters:

- long? AgentId – agent code;
- **string ObtainMethod** – obtain method. It may have values:
    - delivery – delivery;
    - ownStorePickup – own store pickup;
- **long ReserveOfficeId** – office code;
- long? ManagerId – manager code;
- long? TerminalId – self-service terminal code, from which the order was created;
- string TargetReserveType – reserve typr. It may have values:
    - reserve – reserve. Value by default, if in a query parameter nothing is specified;
    - draft – draft. Reservation of articles with this type is not made;
- **object[] Articles** – articles:
  - long Id – code;
  - decimal Quantity – ordered quantity;
- *decimal BonusAmount* – bonus amount, which will be used for payment of a reserve;
- *long PaymentTypeId* – code of payment method;
- object Delivery – delivery information:
  - **long AddressId** – delivery address code;
  - **long AddressId** – delivery time code;
  - **DateTime Date** – delivery date;
  - **string Option** – delivery options. It may have values:
    - own – own;
    - outsource – third-party transport company;
    - hybrid – own to logistic company, and then from logistic company;
  - long? LogisticCompanyId – logistic company code;

  - decimal? LogisticCompanyCost – delivery cost by logistic company;
  - *string Comments* – comments to an order;
  - *string ContactName* – contact name;
  - *string ContactPhone* – contact phone.

Response parameters:

- long Id – code of created reserve by the request .
- decimal Amount – reserve amount;
- DateTime DeadDate – dead date of the reserve;
- bool OutOfStock –article quantity, that could not be reserved;
- long Version – version of created reserve document.

GetReserveInfo

Getting information on placed order.

Request parameters:

- long? AgentId Agent code
- **long Id**– document reserve code;

Response parameters:

- decimal Amount – reserve amount;
- string ObtainMethod– order obtain method. It may have values:
    - delivery – delivery;
    - ownStorePickup – own store pickup;
- long ReserveStoreId – store code where the order shipment is made;
- long ReserveOfficeId– code of office;
- long PaymentTypeId – code of payment method;
- DateTime? DeadDate – dead date of reserve;
- DateTime CreationDate – reserve creation date;
- DateTime TransactionDate – transaction date of reserve;
- long SubTypeId – document subtype code;
- decimal ChargedBonusAmount – amount of charged bonus;
- decimal UsedBonusAmount – amount of used bonus;
- long PriceCategoryId – price category code;
- long Version – document version.
- object Delivery – delivery information:
  - long AddressId – delivery address code;
  - long TimeId – delivery time code;
  - DateTime Date – delivery date;
  - string Option – delivery options. It may have values:
    - own – own;
    - outsource – third-party of logistic company;
    - hybrid – own to logistic company, and then from logistic company;
  - long? LogisticCompanyId – logistic company code;
  - decimal? LogisticCompanyCost – delivery cost by logistic company;
  - string Comments – comments to an order;
  - string ContactName – contact name;
  - string ContactPhone – contact phone.

GetReserveArticles

Getting a list of the order of products, which were reserved at the store.

Request parameters:

- long? AgentId – code of agent;
- **long Id** – document reserve code.

Response parameters:

- []:
  - long Id – product code;
  - long Quantity – total quantity of products in order;
  - long StoreQuantity – quantity which was reserved;
  - decimal Price – price;
  - decimal Amount – amount.

GetReserveDraftedArticles

Getting a list of the order of products, which were not reserved at the store.

Request parameters:

- long? AgentId – code of agent;
- **long Id** – document reserve code.

Response parameters:

- []
  - long Id – product code;
  - long TotalQuantity – total quantity of products in order;
  - long DraftedQuantity – drafted quantity which was not reserved.

GetDocumentPrintFormData

Get Document Print Form Data.

Query parameters:

- long? AgentId – agent code;
- **long Id** – reserve document code;
- **string DataType** – data format. Can accept values:
  - pdf – Acrobat Reader format;
  - xls – Microsoft Excel format;
  - xlsx – Microsoft Excel 2007 version and older;
  - rtf – Rich Text Format (maintained by Microsoft Word);
  - png – raster image;
  - html – html page.

Response parameters:

- byte[] Data – data.

UpdateReserve

Updating of order. Only articles specified in the list *Articles* are updated. At the same time, if the quantity (*Quantity*) of article is equal to zero in the list, article is deleted from the order.

Request parameters:

- long? AgentId – agent code;
- **long Id** – document version;
- **long Version** – document version;
- decimal? BonusAmount – bonus amount, which will be used for payment of a reserve. If the amount is not changed, the value *null* is used;
- object[] Articles – articles:
  - long Id – product code;
  - long Quantity – article quantity If quantity of the article is equal to zero, article is deleted from the order.

Response parameters:

- bool OutOfStock –article quantity, that could not be reserved;
- bool Success – result;
- long Version – document version after up-dating.

UpdateReserveDelivery

Update of reserve delivery.
The method is used for changing of delivery options for the given order. The order must be in state Reserve or Draft.

Request Options:

- **long ID** – document code;
- **string ObtainMethod** - method of getting:
  - ownStorePickup - customer pickup; delivery is deleted from the document;
  - delivery - delivery, existing delivery is deleted and a new one from the given options below is created;
- DeliveryInfo Delivery- Delivery options:
  - **long AddressID** - code of the delivery address;
  - decimal LogisticCompanyCost - Delivery price by logistic company;
  - string Comments - comments to delivery address;
  - string ContactPhone - Contact phone;
  - string ContactName - contact person name;
  - **DateTime Date** - delivery date;
  - **long TimeId** - code of delivery time;
  - long? LogisticCompanyId - logistic companies code;
  - **string Option** - delivery options. May have values:
    - own – delivery by own means;
    - outsource – external transport company;
    - hybrid – by own means to the transport company and then by transport company;

Answer options:

- decimal Amount – delivery cost.

ExecuteDocumentsCommand

Execute Documents Command.

Query parameters:

- long? AgentId – agent code;
- **long[] DocumentsId** – the document code list which it is necessary should execute a command over
- **long CommandId** – a command code;
- object[] CommandParameters – command parameters:
  - string Key – a parameter name;
  - object Value – parameter value.

Response parameters:

- bool Success – result;
- long[] Versions – the document version list after command execution in the order specified in the query parameter *DocumentsId*.

DeleteReserve

Deleting of reserve document.

Request parameters:

- long? AgentId – agent code;
- **long Id** – document code.

Response parameters:

- bool Success – result.

GetReserves

Getting a list of reserve documents.

Inquiry parameters:

- long? AgentId – agent id;
- decimal? AmountFrom – the sum of documents should not be less than specified one;
- decimal? AmountTo – the sum of documents should not exceed the specified one;
- DateTime? CreationDateFrom – the creation date of documents should not be less than specified one;
- DateTime? CreationDateTo – the creation date of documents should not exceed the specified one;
- DateTime? TransactionDateFrom – the transaction date of documents should not be less than specified one;
- DateTime? TransactionDateTo – the creation date of documents should not exceed the specified one;
- DateTime? DeadDateFrom – date of removal of a reserve on the documents should not be less than specified one;
- DateTime? DeadDateTo – date of removal of a reserve on the documents should not exceed the specified one;
- string [] Groups – groups of subtypes of reserve documents. They can have values:
  - draft – a draft. Reservation of articles with such subtype is not made;
  - shipping – delivering/shipping. In these subtypes the documents are in processing (delivery);
  - shipped – delivered/shipped. In these subtypes the documents are received by the client;
- long? PageNumber – page number from 1 (for a paginal output). For example, in total 30 documents satisfy the inquiry parameters. At the same time the number of documents for a page is set equal 10 to by the *RecordsPerPage* parameter. If to specify the value 2 in the *PageNumber* parameter, the documents will get in response from 10 to 19 (page 1 - documents 1-9);

- long? RecordsPerPage – the number of documents on the page (for a paginal output).

Reply parameters:

- object [] Documents – the documents satisfying to inquiry parameters:
  - long Id – document id;
  - decimal Amount – document sum;
  - long Version – version of the document;
  - long ReserveStoreId – store id;
  - long ReserveOfficeId – office id;
  - long PriceCategoryId – price category id;
  - long SubTypeId – subtype id;
  - DateTime CreationDate – date of the document creation;
  - DateTime DeadDate – date of a reserve removal;
  - DateTime TransactionDate – date of the document transaction;
  - decimal ChargedBonusAmount – the sum of the charged bonuses;
  - decimal UsedBonusAmount – the sum of the used bonuses;
  - string ObtainMethod – a method of an order receiving;
- long TotalRecords – the number of documents in a reply;
- decimal? AmountMin – minimum sum of the document;
- decimal? AmountMax – maximum sum of the document;
- DateTime? CreationDateMin – minimum date of the document creation;
- DateTime? CreationDateMax – maximum date of the document creation;
- DateTime? TransactionDateMin – minimum date of the document transaction;
- DateTime? TransactionDateMax – maximum date of the document transaction;
- DateTime? DeadDateMin – minimum date of a reserve removal;
- DateTime? DeadDateMax – maximum date of a reserve removal;

Accounting

GetInvoices

Get Invoices.

Query parameters:

- **long AgentId** – agent code;
- decimal? AmountFrom – documents amount shall not be less than specified;
- decimal? AmountTo – documents amount shall not be greater than specified;
- DateTime? CreationDateFrom – document creation date shall not be less than specified;
- DateTime? CreationDateTo – document creation date shall not be greater than specified;
- long? PageNumber – page number (for page-at-a-time printing). For example, in sum, 30 documents satisfy query parameters. At the same time, the *RecordsPerPage* parameter specifies the number of documents per page equal 10. If enter "1" for the *PageNumber* parameter, the response will be documents from 10 to 19 (page 0 – documents 0...9, page 1 – documents 10...19, page 2 – documents 20...29);
- long? RecordsPerPage – number of documents per page (for page-at-a-time printing).

Response parameters:

- object[] Documents – documents that satisfy query parameters::
  - long Id – document code;
  - string AccountingNo – document's accounting number;
  - decimal Amount – document amount;
  - DateTime CreationDate – document creation date;

- long TotalRecords – number of documents per response;
- decimal? AmountMin – document's minimal amount;
- decimal? AmountMax – document's maximal amount;
- DateTime? CreationDateMin – document's minimal date of creation;
- DateTime? CreationDateMax – document's maximal date of creation.

GetInvoice

Get Invoice.

Query parameters:
- **long AgentId** – agent code;
- **long Id** – document code.

Response parameters:
- long Id – document code;
- string AccountingNo – document's accounting number;
- decimal Amount – document amount;
- DateTime CreationDate – document creation date.

Utility

PrintDocumentWithTerminal

Printed output of the specified document printing form on the self-service terminal printer.

Query parameters
- **long Id** – a document code;
- long? AgentId – agent code;
- **long PrintFormId** – a printing form code;
- **long TerminalId** – the self-service terminal code.

Response parameters:
- bool Success – result.

SendSMS

Send SMS.

Query parameters:
- **string Phone** – phone number;
- **string Text** – message text.

Response parameters:
- bool Success – result.

Integration with different systems

GetDocuments

Enter topic text here.

GetChangedObjects

Enter topic text here.

Replication

Replication is methods designed to back up necessary data from a system database to a website local database. Replication of data through the mentioned methods can be performed either by direct backup or via the intermediate database Redis, if mass data shall be backed up. Some methods can support incremental backup, however to do this you need to know the time of the last successful call of such methods.

Common

Enter topic text here.

GetBanks

Get all bank list.

This method does not have query parameters.

Response parameters:

- []
  - long Id – a bank code;
  - string Name – bank name;
  - string Bic – RCBIC;
  - string CorrAccount – correspondent account;
  - string Address – bank address.

GetBands

Get all brand list.

This method does not have query parameters.

Response parameters:

- []
  - long Id – a brand code;
  - string Name – brand name.

GetCurrencies

Get all currency list.

This method does not have query parameters.

Response parameters:

- []
  - long Id – a currency code;
  - string Name – currency name;
  - bool MainCurrency – main currency;

  - decimal Rate – the main currency course.

GetLocations

Get Locations (geographic locations of company's operations).

There are no query parameters for the method.

Response parameters:

- []
  - long Id – location code;
  - string Name – location name;
  - long? ParentId – parent location code (used to build a tree structure);
  - long? ZoneId – code of a price zone applied to the given location.

GetNow

Get Now (Current Time In Various Formats).

There are no query parameters for the method.

Response parameters:

- []
  - string Time – in the format *DD.MM.YYYY HH:MM:SS*;
  - string IsoTime – in the format ISO 8601;
  - long Timestamp – Unix timestamp;
  - DateTime Date.

GetOffices

Get Offices.

There are no query parameters for the method.

Response parameters:

- []
  - long Id – office code;
  - string Name – office name;
  - long LocationId – code of an office location (a geographic object, where the office is located);
  - string Address – office address.

GetStores

Getting a list of all the stores.

Method has no request parameters.

Response parameters:

- []
  - long Id – store code;
  - long OfficeId – an office code where the store is placed.

GetWarrantyPeriodUnits

Getting a list of all Unit of measurements of warranty periods.

Method has no request parameters.

Response parameters:

- []
  - long Id – Unit of measurement code;
  - string Name – name.

RedisGetProducts

Get a product list. The method supports incremental updating. Data are load through Redis.

Query parameters:

- long[] ProductIds – article codes;
- long[] CategoryIds – article category codes;
- DateTime? UpdatedAfter – to unload only data updated after the specified date.

Response parameters:

- String[] Keys – the Redis key list.

The data unloaded in Redis:

- []
  - long ProductId – an article code;
  - long CategoryId – a category code;
  - string Name – name;
  - long BrandId – a brand code;
  - DateTime CreationDate – creation date;
  - long? OriginalProductId – an original article code (it is used, for example, for markdown articles);
  - decimal Weight – weight;
  - decimal Volume – volume;
  - long WarrantyPeriod – a warranty period;
  - long WarrantyPeriodUnitId – a warranty period unit code.

RedisGetProductPrices

Get product prices. The method supports incremental updating. Data are load through Redis.

Query parameters:

- long[] ZoneIds – a code list of price zones;
- long[] CategoryIds – a code list of price categories;
- DateTime? UpdatedAfter – to unload only data updated after the specified date.

Response parameters:

- String[] Keys – the Redis key list.

The data unloaded in Redis:

- []
  - long ProductId – an article code;
  - decimal Value – price.
  - decimal? PrevValue – previous price;
  - long CategoryId – a price category code;
  - long ZoneId – a price zone code.

RedisGetProductRemains

Get product remains at the stores. The method supports incremental updating. Data are load through Redis.

Query parameters:

- long[] Ids – the store code list.
- DateTime? UpdatedAfter – to unload only data updated after the specified date.

Response parameters:

- String[] Keys – the Redis key list.

The data unloaded in Redis:

- []
  - long ProductId – an article code;
  - long StoreId – a store code;
  - long RemainsQuantity – general remain at the store;
  - long ReserveQuantity – reserved quantity (*RemainsQuantity* includes it);
  - decimal SellSpeed – sell speed.

Delivery

Enter topic text here.

GetDeliveryTimeRanges

Get Delivery Time Ranges.

There are no query parameters for the method.

Response parameters:

- []
  - long Id – range code;
  - string Name – range name;
  - string TimeFrom – time range beginning in *HH:MM*;
  - string TimeTo – time range ending in HH:MM;
  - DateTime DateTimeFrom – time range beginning;
  - DateTime DateTimeTo – time range ending.

GetLogisticCompanies

Get Logistic Companies.

There are no query parameters for the method.

Response parameters:

- []
  - long Id – company code;
  - string Name – company name.

Article catalog

Enter topic text here.

GetNativeCategories

Get Native Categories (Articles Dictionary).

There are no query parameters for the method.

Response parameters:

- []
  - long Id – category code;
  - string Name – category name;
  - long? ParentId – parent category code (used to build a tree structure).

GetSiteCategories

Receiving of tree of all article categories of the website (online article Dictionary).

Method has no request parameters.

Response parameters:

- []
  - long Id – code of onlinecategory;
  - string Name – name of onlinecategory;
  - string UrlName – name of online category for URL (ASCII);
  - string BuyName – personified text of the button "Buy" for articles of this online category;
  - long SortIndex – sort index;
  - long? OriginalId – a code of origina online category if the current one is an alias;
  - long? ParentId – a code of parental online category (it is used to build the tree).

GetNativeToSiteCategories

Get Native To Site Categories.

There are no query parameters for the method.

Response parameters:

- []
  - long CategoryId – category code;
  - long SiteCategoryId – site category code.

Article characteristics

Enter topic text here.

GetProductProperties

Get Product Properties. Properties values united into a template make an article description.

There are no query parameters for the method.

Response parameters:

- []
  - long Id – property code;
  - string Name – property name;
  - long? ParentId – parent property code;
  - long TypeId – property type code. Can accept values:
    - 1 – number;
    - 2 – Boolean;
    - 3 – string (up to 2'048 signs);
    - 4 – text (more than 2'048 signs);
    - 5 – selection of one value among several;
    - 6 – selection of several values among several;
    - 7 – complex property;
  - long? UnitId – property Unit of measurement code;
  - long KindId – property kind code. Can accept values:
    - 1 – regular;
    - 2 – filter;
    - 3 – navigation;
  - bool IsHidden – hide online property (yes/no);
  - bool SortValues – sort property values (yes/no).

GetProductPropertyGroups

Get Product Property Groups. Properties are united in groups, e.g., "Display" group defines properties such as "Display size", "resolution", "Matrix type", etc.

There are no query parameters for the method.

Response parameters:

- []
  - long Id – properties group code;
  - long Name – group name;
  - long SortIndex – sort index.

GetProductPropertyUnits

Get Product Property Units.

There are no query parameters for the method.

Response parameters:

- []
  - long Id – Unit of measurement code;
  - long Name – Unit of measurement name.

GetProductPropertiesToTemplates

Get Product Properties To Templates.

There are no query parameters for the method.

Response parameters:

- []
  - long TemplateId – property template code;
  - long PropertyId – property code;
  - long PropertyGroupId – property group code;
  - long PropertySortIndex – sort index.

GetProductTemplatesToNativeCategories

Getting of links of characteristics templates and product types.

Request parameters:

- long[] CategoryIds – a list of product categories codes.

Response parameters:

- []
  - long CategoryId – products categories code;
  - long ProductTemplateId – characteristic template code.

GetProductPropertyValues

Getting of product characteristic values list.

Method has no request parameters.

Response parameters:

- []
  - long Id – value code;
  - long? ParentId – parental value code;
  - string Value – value;
  - long SortIndex – sort index;
  - long PropertyId – product characteristic code.

GetProductsToProductProperties

Getting of product links and product characteristic.

Method has no request parameters.

Response parameters:

- []
  - long ProductId – product code;
  - long PropertyId – product characteristic code;
  - long ValueId – product characteristic value code (0 if value is set through one of properties **Value*);
  - decimal? NumberValue – number value;
  - bool? BooleanValue – boolean value;
  - string StringValue – string value;
  - string TextValue – text value.

GetProductPhotoViews

Get Product Photo Views. E.g., "in package", "unpacked", or "general view".

There are no query parameters for the method.

Response parameters:

- []
  - long Id – view code;
  - string Name – view name;
  - long SortIndex – sort index;
  - bool Required – if the view is required.

GetProductPropertyGroupsToPhotoViews

Get Product Property Groups To Photo Views.

There are no query parameters for the method.

Response parameters:

- []
  - long PhotoViewId – photo view code;
  - long ProductTemplateId – properties group code.

RedisGetProductPhotos

Get product photos The method supports incremental updating. Data are load through Redis.

Query parameters:

- long[] ProductIds – article codes;
- long[] CategoryIds – article category codes;
- DateTime? UpdatedAfter – to unload only data updated after the specified date.

Response parameters:

- String[] Keys – the Redis key list.

The data unloaded in Redis:

- []
  - long ProductId – an article code;
  - long ViewId – a photo type code;
  - byte[] Content – the photo.