connect.
engage.
succeed.

STOREWORLD 5.0

Omni channel interface description

centric
connect.engage.succeed.

HISTORY

Version	Date	Name	Comment
1.	16/06/2014	Houtekier, Thomas	First draft.
2.	25/06/2014	De Vidts, Pieter	Added customer order operations
3.	25/10/2014	Houtekier, Thomas	Language code property removed from call to GetCustomerDetails since it is not implemented.
4.	18/05/2015	De Vidts, Pieter	Added IssueExistingGiftvoucher operation.
5.	17/09/2015	De Vidts, Pieter	Added the operations BlockLoyaltyCard, UnblockLoyaltyCard, DeactivateLoyaltyCard and GetTicketDetails.
6.	18/09/2015	De Vidts, Pieter	Added 'ger' to language codes.
7.	12/10/2015	De Vidts, Pieter	Changed language codes.
8.	20/10/2015	De Vidts, Pieter	Changed language codes.
9.	22/01/2016	Clevers, Nico	Changes to customer order operations
10.	26/02/2016	Clevers, Nico	Added extra fault codes to the Place…Order methods
11.	14/03/2016	Clevers, Nico	Added 'AllowPickAlternative'
12.	29/03/2016	Clevers, Nico	Added MaxLength column to the Place…Order methods.
13.	12/05/2016	Clevers, Nico	Added explanation about the change to the OrganizationHierarchy.
14.	17/05/2016	De Vidts, Pieter	Added the operations UpdateCustomerNumber, TransferLoyaltyPoints, CorrectLoyaltyPoints and ActivateLoyaltyCard.
15.	20/06/2016	Clevers, Nico	Added operation TopUpGiftvoucher
16.	11/08/2016	De Vidts, Pieter	Added error code 12026 to the operations transferLoyaltyPoints and CorrectLoyaltyPoints.
17.	5/09/2016	Clevers, Nico	Added extra possible error code in creation customer order.
18.	9/01/2017	De Vidts, Pieter	Added CustomerNumber and HasCreditRight to registerCustomer. Added HasCreditRight and DeactivationDate to updateCustomer.

			Added operations registerCustomerDetails and updateCustomerDetails. Added CheckAddress to CustomerAddress. Added Name and LegalForm to CustomerEnterprise.
9	19/01/2017	Clevers, Nico	Added operation RegisterArtsDitigalReceipt operation
	03/02/2017	Magop, Valentina	Extra functionality like multiple delivery addresses, method to calculate basket value, other
21	24/02/2017	Agache, Ștefan	Removed stamp registration; removed pointTypeId from point registration and added invoice email to register customer and update customer.
22	09/03/2017	Magop, Valentina	Added Password property on the commands to register customer

centric
connect.engage.succeed.

CONTENTS

History **2**
Contents **4**
1 Purpose and goal **8**
2 Implementation **8**
3 Customer operations **9**
3.1 Command operations 9
3.1.1 Register a new customer 9
3.1.2 Register a new customer with details 11
3.1.3 Update an existing customer 13
3.1.4 Update an existing customer with details 14
3.1.5 Set the customers default address 16
3.1.6 Set the customers delivery address 17
3.1.7 Set the customers invoicing address 18
3.1.8 Update customer number 19
3.2 Query operations 20
3.2.1 Get the customer details 20
3.2.2 Get the customer details by email address 21
3.3 Data contracts 21
3.3.1 Customer 21
3.3.2 CustomerName 22
3.3.3 CustomerEnterprise 22
3.3.4 CustomerEmailAddress 22
3.3.5 CustomerAddress 23
3.3.6 CustomerContactAddress 23
4 Loyalty operations **24**
4.1 Command operations 24
4.1.1 Create a default loyalty card 24
4.1.2 Register loyalty points 25
4.1.3 Redeem loyalty points for a value 26
4.1.4 Link an existing loyalty card to an existing customer 28
4.1.5 Block loyalty card 29
4.1.6 Unblock loyalty card 30
4.1.7 Deactivate loyalty card 31

4.1.8 Transfer loyalty points 32
4.1.9 Correct loyalty points 33
4.1.10 Activate loyalty card 34
4.2 Query operations 35
4.2.1 Get the loyalty card details 35
4.2.2 Get the point types 36
4.2.3 Get the point groups 37
4.2.4 Get the loyalty card types 38
4.2.5 Get valid redeem values 39
4.3 Data contracts 40
4.3.1 LoyaltyCard 40
4.3.2 PointTypeBalance 40
4.3.3 PointGroupBalance 40
4.3.4 PointType 41
4.3.5 PointGroup 41
4.3.6 LoyaltyCardType 41
4.3.7 RedeemValue 41
5 E-voucher operations 42
5.1 Command operations 42
5.1.1 Issue a giftvoucher 42
5.1.2 Issue an existing giftvoucher 43
5.1.3 Top-up a giftvoucher 44
5.1.4 Issue a discount coupon 45
5.1.5 Redeem a giftvoucher 46
5.1.6 Redeem a discount coupon 47
5.1.7 Redeem a return coupon 49
5.1.8 Redeem a deposit coupon 50
5.2 Query operations 51
5.2.1 Get the details of an e-voucher 51
5.3 Data contracts 52
5.3.1 EvoucherTransaction 52
6 CustomerOrder Operations 53
6.1 Command operations 53
6.1.1 Place web order 53
6.1.2 Place in shop order 55

6.1.3 Cancel obsolete orders 56
6.2 Query operations 57
6.3 Data contracts 58
6.3.1 WebCustomerOrderLine 58
6.3.2 InShopCustomerOrderLine 58
6.3.3 DeliveryTimeFrame 58
6.3.4 WebOrderLine 59
6.3.5 PriceWebOrderLine 59
6.3.6 PromopackDiscount 59
6.3.7 PromopackInfo 60
6.3.8 PointRegistration 60
6.3.9 StampRegistration 60
6.3.10 PromopackVoucherRegistration 60
7 Ticket operations 61
7.1 Command operations 61
7.1.1 Register an ARTS Digital receipt 61
7.2 Query operations 62
7.2.1 Get ticket details 62
7.3 Data contracts 62
7.3.1 Ticket reference 62
8 Common data contracts 63
8.1 Amount 63
8.2 Organization Hierachies 63
9 Codes 64
9.1 Languages codes 64
9.2 Gender codes 64
9.3 Country codes 64
9.4 Currency codes 64
9.5 E-voucher type codes 65
9.6 E-voucher status codes 65
9.7 E-voucher transaction type codes 65
9.8 Organization Hierarchies 66
9.9 Delivery type codes 66

Copyright 2013, Centric Belgium NV

1 PURPOSE AND GOAL

This document describes how integrating parties can integrate with Storeworld OBP. The document gives an overview of the integration possibilities within the following domains:

- Customers
- Loyalty
- E-vouchers
- Customer orders

The interfaces are described at a functional level. Data contracts (in and out) and fault contracts are defined for each operation.

This implementation does not take any security requirements into consideration yet. The service is accessible through a basic http binding. No binding policies are active yet.

All interface descriptions are under embargo.

2 IMPLEMENTATION

Integration between a web-shop (or any other online sales platform) and Storeworld can be achieved via the OnlineSalesService within Storeworld OBP.

3 CUSTOMER OPERATIONS

3.1 COMMAND OPERATIONS

3.1.1 REGISTER A NEW CUSTOMER

The operation **RegisterCustomer** on the **OnlineSaleService** can be used.

A **RegisterCustomerCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CustomerNumber	Default this should be empty, because Storeworld defines the customer number. Required when Storeworld is not leading for customer data and the external system defines the customer number.
Name	See **CustomerName** validation rules.
Enterprise	See **CustomerEnterprise** validation rules.
Gender	Required, should be one of the supported gender codes.
LanguageCode	Required, should be one of the supported language codes.
EmailAddresses	A list of customer email addresses. See **CustomerEmailAddress** data contract.
MobileNumbers	A list of customer mobile numbers. Mobile numbers must follow the regex `0[0-9]{9}$`
CardNumber	Optional. The number of the loyalty card to be linked to the customer. The loyalty card must exist.
HasCreditRight	Required, flag indicating if the customer has credit rights.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.
IsOnlineActive	Optional, flag indicating if customer email address was activated or not.
Password	Optional, should be the customer encrypted password (the encryption and decryption is done by webshop party).
InvoicingEmailAddress	The email address used for invoicing. See **CustomerEmailAddress** data contract.

A **RegisterCustomerResult** is returned when the customer registration succeeded. It includes the customer number of the newly registered customer.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1001	The loyalty card was not found.
1002	The loyalty card is already linked to a customer (the loyalty card is not pending).
1003	The customer already exists.

3.1.2 REGISTER A NEW CUSTOMER WITH DETAILS

The operation **RegisterCustomerDetails** on the **OnlineSaleService** can be used.

A **RegisterCustomerDetailsCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CustomerNumber	Default this should be empty, because Storeworld defines the customer number. Required when Storeworld is not leading for customer data and the external system defines the customer number.
Name	See **CustomerName** validation rules.
Enterprise	See **CustomerEnterprise** validation rules.
Gender	Required, should be one of the supported gender codes.
LanguageCode	Required, should be one of the supported language codes.
EmailAddresses	A list of customer email addresses. See **CustomerEmailAddress** data contract.
MobileNumbers	A list of customer mobile numbers. Mobile numbers must follow the regex `0[0-9]{9}$`
CardNumber	Optional. The number of the loyalty card to be linked to the customer. The loyalty card must exist.
HasCreditRight	Required, flag indicating if the customer has credit rights.
DefaultAddress	The customers default address. See **CustomerAddress** data contract.
DeliveryAddress	The customer's default delivery contact address. See **CustomerContactAddress** data contract.
DeliveryAddresses	The customer delivery contact addresses – it enables adding more than one delivery address; only the one mentioned in the DeliveryAddress field will be created as the default delivery address. See **CustomerContactAddress** data contract.
InvoicingAddress	The customer invoicing contact address. See **CustomerContactAddress** data contract.
InvoicingEmailAddress	The email address used for invoicing. See **CustomerEmailAddress** data contract.
DeactivationDate	Optional, deactivation date of the customer.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.
IsOnlineActive	Optional, flag indicating if customer email address was activated or not.
Password	Optional, should be the customer encrypted password (the

encryption and decryption is done by webshop party).

A **RegisterCustomerDetailsResult** is returned when the customer registration succeeded. It includes the customer number of the newly registered customer. An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1001	The loyalty card was not found.
1002	The loyalty card is already linked to a customer (the loyalty card is not pending).
1003	The customer already exists.

Remark: It is possible to configure this operation to allow updates when the customer already exists (upsert). This is only possible when Storeworld is NOT leading for the customer data.

3.1.3 UPDATE AN EXISTING CUSTOMER

The operation **UpdateCustomer** on the **OnlineSaleService** can be used.

An **UpdateCustomerCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CustomerNumber	Required.
Name	See **CustomerName** data contract.
CustomerEnterprise	See **CustomerEnterprise** data contract.
Gender	Required, should be one of the supported gender codes.
LanguageCode	Required, should be one of the supported language codes.
EmailAddresses	A list of customer email addresses. See **CustomerEmailAddress** data contract.
MobileNumbers	A list of customer mobile numbers. Mobile numbers must follow the regex `0[0-9]{9}$`
HasCreditRight	Required, flag indicating if the customer has credit rights.
DeactivationDate	Optional, deactivation date of the customer.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.
IsOnlineActive	Optional, flag indicating if customer email address was activated or not.
Password	Optional, should be the customer encrypted password (the encryption and decryption is done by webshop party).
InvoicingEmailAddress	The email address used for invoicing. See **CustomerEmailAddress** data contract.

Nothing is returned when the update succeeded. An **OmniChannelFault** is returned when the registration failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer to be updated was not found.

3.1.4 UPDATE AN EXISTING CUSTOMER WITH DETAILS

The operation **UpdateCustomerDetails** on the **OnlineSaleService** can be used.

An **UpdateCustomerDetailsCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CustomerNumber	Required.
Name	See **CustomerName** data contract.
CustomerEnterprise	See **CustomerEnterprise** data contract.
Gender	Required, should be one of the supported gender codes.
LanguageCode	Required, should be one of the supported language codes.
EmailAddresses	A list of customer email addresses. See **CustomerEmailAddress** data contract.
MobileNumbers	A list of customer mobile numbers. Mobile numbers must follow the regex `0[0-9]{9}$`
HasCreditRight	Required, flag indicating if the customer has credit rights.
DeactivationDate	Optional, deactivation date of the customer.
DefaultAddress	The customers default address. See **CustomerAddress** data contract.
DeliveryAddress	The customer's default delivery contact address. See **CustomerContactAddress** data contract.
DeliveryAddresses	The customer delivery contact addresses – it enables adding more than one delivery address; only the one mentioned in the DeliveryAddress field will be created as the default delivery address. See **CustomerContactAddress** data contract.
InvoicingAddress	The customer invoicing contact address. See **CustomerContactAddress** data contract.
InvoicingEmailAddress	The email address used for invoicing. See **CustomerEmailAddress** data contract.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.
IsOnlineActive	Optional, flag indicating if customer email address was activated or not.
Password	Optional, should be the customer encrypted password (the encryption and decryption is done by webshop party).

Nothing is returned when the update succeededAn **OmniChannelFault** is returned when the registration failed. Following error codes can be returned:

Error code	Description

1	Validation error.
1000	The customer to be updated was not found.

3.1.5 SET THE CUSTOMERS DEFAULT ADDRESS

The operation **SetCustomerDefaultAddress** on the **OnlineSaleService** can be used.

A **SetCustomerDefaultAddressCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CustomerNumber	Required.
Address	See **CustomerAddress** data contract.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the registration failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer to be updated was not found.

3.1.6 SET THE CUSTOMERS DELIVERY ADDRESS

The operation **SetCustomerDeliveryAddress** on the **OnlineSaleService** can be used.

A **SetCustomerDeliveryAddressCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CustomerNumber	Required.
ContactAddress	See **CustomerContactAddress** data contract.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the registration failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer to be updated was not found.

3.1.7 SET THE CUSTOMERS INVOICING ADDRESS

The operation **SetCustomerInvoicingAddress** on the **OnlineSaleService** can be used.

A **SetCustomerInvoicingAddressCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CustomerNumber	Required.
ContactAddress	See **CustomerContactAddress** data contract.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the registration failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer to be updated was not found.

3.1.8 UPDATE CUSTOMER NUMBER

The operation **UpdateCustomerNumber** on the **OnlineSaleService** can be used.

A **UpdateCustomerNumberCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required if old customer number is not specified.
OldCustomerNumber	Required if card number is not specified.
NewCustomerNumber	Required.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer to be updated was not found.
1001	The loyalty card was not found.

3.2 QUERY OPERATIONS

3.2.1 GET THE CUSTOMER DETAILS

The operation **GetCustomerDetails** on the **OnlineSaleService** can be used.

A **GetCustomerDetailsCall** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CustomerNumber	Required if CardNumber is not specified.
CardNumber	Required if CustomerNumber is not specified.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.
EmailAddress	Required. Indicates customer email address.

A **GetCustomerDetailsResult** is returned when a customer was found for the given card number or customer number. An empty **GetCustomerDetailsResult** is returned when no customer was found for the given criteria. The **GetCustomerDetailsResult** includes following properties:

Property	Description
Customer	The details of the customer. See **Customer** data contract.

No faults are defined for this operation.

3.3 DATA CONTRACTS

3.3.1 CUSTOMER

These are the properties for customer:

Property	Description
CustomerNumber	The customer number of the returned customer. No customer was found when value is not specified.
Name	The name of the customer. Including title, given name, infix and family name. Family name can never be null.
Gender	Gender of the customer. One of the supported gender codes.
LanguageCode	Language of the customer. One of the supported language codes.
DefaultAddress	The customers default address.
DeliveryAddress	The customer's default delivery contact address. See **CustomerContactAddress** data contract.
DeliveryAddresses	The customer list of delivery contact addresses. See CustomerContactAddress contract.
InvoicingEmailAddress	The email address used for invoicing. See **CustomerEmailAddress** data contract.
InvoicingAddress	The customer invoicing contact address.
MobileNumbers	A list of the customer mobile numbers.
EmailAddresses	A list of the customer email addresses. Each email address includes a flag to indicate that the email address is marked for a verification request.
Enterprise	The customers enterprise information.
HasCreditRight	Flag indicates if customer has credit rights.
DeactivationDate	Deactivation date of the customer.
IsOnlineActive	Optional, flag indicating if a customer is online active or not.
Password	Optional, should be the customer encrypted password (the encryption and decryption is done by webshop party).
InvoicingEmailAddress	Optional. Entered by the customer. In case it is not entered, the login e-mail address is used.

3.3.2 CUSTOMERNAME

These are the properties for a customer name:

Property	Description
Title	Optional, maximum 10 positions
GivenName	Optional, maximum 30 positions
Infix	Optional, maximum 10 positions
FamilyName	Required, maximum 30 positions

3.3.3 CUSTOMERENTERPRISE

These are the properties for a customer enterprise:

Property	Description
CountryCode	Required when number is defined, one of the supported country codes.
Number	Required when country code is defined Enterprise number that is valid for the given country code. Validation is applied for these countries: • Netherlands • Belgium • France
Name	Optional, the name of the enterprise.
LegalForm	Optional, the legal form of the enterprise.

3.3.4 CUSTOMEREMAILADDRESS

These are the properties for a customer email address:

Property	Description
EmailAddress	Required, a valid email address.
ReceiveEFolder	Required, flag indicating if an e-folder is desired for the customer email address.

3.3.5 CUSTOMERADDRESS

These are the properties for a customer address:

Property	Description
Street	Required, maximum 60 positions
HouseNumber	Required, maximum 10 positions
BoxNumber	Optional, maximum 10 positions
PostalCode	Required, maximum 10 positions
Municipality	Required, maximum 60 positions
CountryCode	Required, one of the supported country codes.
FaxNumber	The validity of the fax number is verified against the country code prefix used in the input. E.g. "+3250833333" will validate against BE.
PhoneNumbers	A list of customer phone numbers. The validity of the phone numbers is verified against the country code prefix used in the input. E.g. "+3250833333" will validate against BE.
CheckAddress	Flag indicates if address should be checked.
AddressNumber	All addresses will have a number that will be unique per combination customer and address type. This field should be set only in OBP.

3.3.6 CUSTOMERCONTACTADDRESS

These are the properties for a customer address:

Property	Description
Name	See **CustomerName** validation rules.
Address	See **CustomerAddress** validation rules.

4 LOYALTY OPERATIONS

4.1 COMMAND OPERATIONS

4.1.1 CREATE A DEFAULT LOYALTY CARD

The operation **CreateDefaultLoyaltyCard** on the **OnlineSalesService** can be used.

The **CreateDefaultLoyaltyCardCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required. The number of the card to be created.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **CreateDefaultLoyaltyCard Result** is returned when the default card creation succeeded. It includes these properties:

Property	Description
CardNumber	The number of the card that has been created.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
12021	The loyalty card already exists.
12022	The loyalty card number is invalid.

4.1.2 REGISTER LOYALTY POINTS

The operation **RegisterLoyaltyPoints** on the **OnlineSalesService** can be used.

The **RegisterLoyaltyPointsCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required.
PointTypeNumber	Required.
NumberOfPoints	Required, positive integer.
ReceiptNumber	The same receipt number as in the receipt that will be delivered though the ticket injection (ARTS) interface.
RegistrationTimestamp	Optional, current timestamp is used when not specified.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **RegisterLoyaltyPointsResult** is returned when the points registration succeeded. It includes the loyalty transaction id (GUID) informatively.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1001	The loyalty card was not found.
12000	The point type is not known.

4.1.3 REDEEM LOYALTY POINTS FOR A VALUE

The operation **RedeemLoyaltyPointsForValue** on the **OnlineSalesService** can be used.

The **RedeemLoyaltyPointsForValueCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required.
PointTypeNumber	Required if point group number is not specified.
PointGroupNumber	Required if point type number is not specified.
NumberOfPoints	Required, positive integer.
ReceiptNumber	The same receipt number as in the receipt that will be delivered though the ticket injection (ARTS) interface.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **RedeemLoyaltyPointsForValueResult** is returned when the points registration succeeded. It includes these properties:

Property	Description
LoyaltyTransactionId	Identification of the loyalty transaction (GUID).
RedeemedNumberPoints	The number of points that have been redeemed.
RedeemedValue	The total value of the redeemed points. See **Amount** data contract.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1001	The loyalty card was not found.
12000	The point type is not known.
12003	The card is inactive.
12007	The card is expired.
12008	The card type is invalid.
12012	The card is blocked.
12013	The card cannot be used for redeeming.
12014	The point type expired.

12015	The number of points to redeem is invalid.
12016	The point balance is insufficient.
12017	The point group is unknown.

4.1.4 LINK AN EXISTING LOYALTY CARD TO AN EXISTING CUSTOMER

The operation **LinkLoyaltyCardToCustomer** on the **OnlineSaleService** can be used.

A **LinkLoyaltyCardToCustomer Command** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required. The number of the card to be linked to the customer.
CustomerNumber	Required. The number of the customer to which the card should be linked.
AllowRelink	Optional. Flag to allow relink when loyalty card is not pending.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the linking succeeded.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer was not found.
1001	The loyalty card was not found.
1002	If the loyalty card is already linked to a customer (the loyalty card is not pending) and the optional flag AllowRelink is not given or is false. (If the optional flag AllowRelink is true a non-pending card will be linked to the given customer number.

4.1.5 BLOCK LOYALTY CARD

The operation **BlockLoyaltyCard** on the **OnlineSaleService** can be used.

A **BlockLoyaltyCardCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required. The number of the card to be blocked.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1001	The loyalty card was not found.

4.1.6 UNBLOCK LOYALTY CARD

The operation **UnblockLoyaltyCard** on the **OnlineSaleService** can be used.

An **UnblockLoyaltyCardCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required. The number of the card to be unblocked.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1001	The loyalty card was not found.

4.1.7 DEACTIVATE LOYALTY CARD

The operation **DeactivateLoyaltyCard** on the **OnlineSaleService** can be used.

A **DeactivateLoyaltyCardCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required. The number of the card to be blocked.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1001	The loyalty card was not found.

4.1.8 TRANSFER LOYALTY POINTS

The operation **TransferLoyaltyPoints** on the **OnlineSaleService** can be used.

A **TransferLoyaltyPointsCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
SourceCardNumber	Required. The number of the source card.
DestinationCardNumber	Required. The number of the destination card.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1001	The loyalty card was not found.
12003	The card is inactive.
12007	The card is expired.
12008	The card type is invalid.
12016	The point balance is insufficient.
12026	The point correction is invalid for blocked card.

4.1.9 CORRECT LOYALTY POINTS

The operation **CorrectLoyaltyPoints** on the **OnlineSaleService** can be used.

A **CorrectLoyaltyPointsCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required. The number of the card.
PointTypeNumber	Required. The number of the point type.
NumberOfPoints	Required. The number of points for the correction (positive number=increment; negative number=decrement)
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1001	The loyalty card was not found.
12000	The point type is not known.
12003	The card is inactive.
12007	The card is expired.
12008	The card type is invalid.
12016	The point balance is insufficient.
12026	The point correction is invalid for blocked card.

4.1.10 ACTIVATE LOYALTY CARD

The operation **ActivateLoyaltyCard** on the **OnlineSaleService** can be used.

An **ActivateLoyaltyCardCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required. The number of the card to be activated.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1001	The loyalty card was not found.

4.2 QUERY OPERATIONS

4.2.1 GET THE LOYALTY CARD DETAILS

The operation **GetLoyaltyCardDetails** on the **OnlineSalesService** can be used.

The **GetLoyaltyCardDetailsCall** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **GetLoyaltyCardDetailsResult** is returned when a card was found for the given card number. An empty **GetLoyaltyCardDetailsResult** is returned when no card was found.

The **GetLoyaltyCardDetailsResult** includes following properties:

Property	Description
CustomerNumber	The customer number of the card owner.
Card	The details of the loyalty card. See **LoyaltyCard** data contract.
PointTypeBalances	A list of point type balances. See **PointTypeBalance** data contract.
PointGroupBalances	A list of point group balances. See **PointGroupBalance** data contract.

No faults are defined for this operation.

4.2.2 GET THE POINT TYPES

The operation **GetPointTypes** on the **OnlineSalesService** can be used.

The **GetPointTypesCall** is taken as input argument. It contains these properties and validation rules:

Property	Validations
LanguageCode	Required, one of the supported language codes.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **GetPointTypesResult** is returned including all point types.

The **GetPointTypesResult** includes following properties:

Property	Description
PointTypes	A list of point types. See **PointType** data contract.

An **OmniChannelFault** is returned when the retrieval of point types failed. Following error codes can be returned:

Error code	Description
1	Validation error.

4.2.3 GET THE POINT GROUPS

The operation **GetPointGroups** on the **OnlineSalesService** can be used.

The **GetPointGroupsCall** is taken as input argument. It contains these properties and validation rules:

Property	Validations
LanguageCode	Required, one of the supported language codes.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **GetPointGroupsResult** is returned including all point groups.

The **GetPointGroupsResult** includes following properties:

Property	Description
PointGroups	A list of point groups. See **PointGroup** data contract.

An **OmniChannelFault** is returned when the retrieval of point groups failed. Following error codes can be returned:

Error code	Description
1	Validation error.

4.2.4 GET THE LOYALTY CARD TYPES

The operation **GetLoyaltyCardTypes** on the **OnlineSalesService** can be used.

The **GetLoyaltyCardTypesCall** is taken as input argument. It contains these properties and validation rules:

Property	Validations
LanguageCode	Required, one of the supported language codes.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **GetLoyaltyCardTypesResult** is returned including all loyalty card types.

The **GetLoyaltyCardTypesResult** includes following properties:

Property	Description
LoyaltyCardTypes	A list of point groups. See **LoyaltyCardType** data contract.

An **OmniChannelFault** is returned when the retrieval of loyalty card types failed. Following error codes can be returned:

Error code	Description
1	Validation error.

4.2.5 GET VALID REDEEM VALUES

The operation **GetValidRedeemValues** on the **OnlineSalesService** can be used.

The **GetValidRedeemValuesCall** is taken as input argument. It contains these properties and validation rules:

Property	Validations
CardNumber	Required.
PointTypeNumber	Required if point group number is not specified.
PointGroupNumber	Required if point group number is not specified.
ReceiptTotal	Required, see **Amount** data contract.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **GetValidRedeemValuesResult** is returned including all allowed redeem values and its corresponding amount.

The **GetValidRedeemValuesResult** includes following properties:

Property	Description
AllowDeviation	Flag that indicates if it is allowed to deviate from the proposed redeem values.
PointTypeNumber	The number of the point type for which the proposed redeem values are valid.
PointGroupNumber	The number of the point group for which the proposed redeem values are valid.
RedeemValues	The list of proposed redeem values. See **RedeemValue** data contract.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1001	The loyalty card was not found.
12000	The point type is not known.
12017	The point group is not known.

4.3 DATA CONTRACTS

4.3.1 LOYALTYCARD

These are the properties of a loyalty card:

Property	Description
CardNumber	Number of the loyalty card.
CardTypeNumber	Number identifying the loyalty card type.
IsBlocked	Flag indicating if the loyalty card is blocked.
CanRedeem	Flag indicating if the loyalty card can be used for redeeming.
IsActive	Flag indicating if the loyalty card is active.
ActivationDate	Activation date of the loyalty card.
ExpiryDate	Expiry date of the loyalty card.

4.3.2 POINTTYPEBALANCE

These are the properties of a point type balance:

Property	Description
PointTypeNumber	Number of the point type.
NumberOfPoints	Point balance for the point type.

4.3.3 POINTGROUPBALANCE

These are the properties of a point group balance:

Property	Description
PointGroupNumber	Number of the point group.
NumberOfPoints	Point balance for the point group.

4.3.4 POINTTYPE

These are the properties of a point type:

Property	Description
Number	Number of the point type.
Description	Description of the point type.

4.3.5 POINTGROUP

These are the properties of a point group:

Property	Description
Number	Number of the point group.
Description	Description of the point group.

4.3.6 LOYALTYCARDTYPE

These are the properties of a loyalty card type:

Property	Description
Number	The number of the loyalty card type.
Description	Description of the loyalty card type.

4.3.7 REDEEMVALUE

These are the properties of a redeem value:

Property	Description
NumberOfPoints	The number of points to be redeemed for the monetary value.
Value	The monetary value to be received for the number of points. See **Amount** data contract.

5 E-VOUCHER OPERATIONS

5.1 COMMAND OPERATIONS

5.1.1 ISSUE A GIFTVOUCHER

The operation **IssueGiftvoucher** on the **OnlineSalesService** can be used.

The **IssueGiftvoucherCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
Value	The value of the gift voucher. See **Amount** data contract.
ValidityPeriod	Optional. The number of days that the gift voucher is valid.
ReceiptNumber	The same receipt number as in the receipt that will be delivered though the ticket injection (ARTS) interface.
CustomerNumber	The number of the customer for which the gift voucher is created.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **IssueGiftvoucherResult** is returned when gift voucher is successfully issued. It includes these properties:

Property	Validations
Barcode	The barcode of the giftvoucher.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer is not found.

5.1.2 ISSUE AN EXISTING GIFTVOUCHER

The operation **IssueExistingGiftvoucher** on the **OnlineSalesService** can be used.

The **IssueExistingGiftvoucherCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
Barcode	The barcode of the gift voucher.
Value	The value of the gift voucher. See **Amount** data contract.
ReceiptNumber	The same receipt number as in the receipt that will be delivered though the ticket injection (ARTS) interface.
CustomerNumber	The number of the customer for which the gift voucher is created.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer is not found.
7507	Gift voucher not found.
7509	Gift voucher is blocked.
7517	Gift voucher has already been issued.
7515	Gift voucher has already been redeemed.
7530	The currency of the gift voucher deviates from the currency of the chain.

5.1.3 TOP-UP A GIFTVOUCHER

The operation **TopUpGiftvoucher** on the **OnlineSalesService** can be used.

The **TopUpGiftvoucherCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
Barcode	The barcode of the gift voucher.
Value	The value for the top-up of the gift voucher. See **Amount** data contract.
ReceiptNumber	The same receipt number as in the receipt that will be delivered though the ticket injection (ARTS) interface.
CustomerNumber	The number of the customer that performs the top-up of the gift voucher.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer is not found.
7507	Gift voucher not found.
7508	Gift voucher not linked to the chain.
7509	Gift voucher is blocked.
7513	Gift voucher is not issued.
7530	The currency of the gift voucher deviates from the currency of the chain.
7560	Top-up is not allowed on the giftvoucher.
7561	The amount of the top-up is too high.

5.1.4 ISSUE A DISCOUNT COUPON

The operation **IssueDiscountCoupon** on the **OnlineSalesService** can be used.

The **IssueDiscountCouponCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
Value	The value of the discount coupon. See **Amount** data contract.
ReceiptNumber	The same receipt number as in the receipt that will be delivered though the ticket injection (ARTS) interface.
CustomerNumber	The number of the customer that for which the discount coupon is created.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

An **IssueDiscountCouponResult** is returned when discount coupon is successfully issued. It includes these properties:

Property	Validations
Barcode	The barcode of the discount coupon.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer is not found.

5.1.5 REDEEM A GIFTVOUCHER

The operation **RedeemGiftvoucher** on the **OnlineSalesService** can be used.

The **RedeemGiftvoucherCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
Barcode	The barcode of the gift voucher.
Value	The value to be redeemed. See **Amount** data contract.
ReceiptNumber	The same receipt number as in the receipt that will be delivered though the ticket injection (ARTS) interface.
CustomerNumber	The number of the customer that redeemed the gift voucher.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **RedeemGiftVoucherResult** is returned when the gift voucher is successfully redeemed. It includes these properties:

Property	Validations
EvoucherTransactionId	Identification of the e-voucher transaction (Guid)
NewBalance	The new balance of the gift voucher. See **Amount** data contract.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer is not found.
7507	Gift voucher not found.
7508	Gift voucher not linked to the chain.
7509	Gift voucher is blocked.
7513	Gift voucher is not issued.
7514	Gift voucher is expired.
7515	Gift voucher has already been redeemed.
7516	Insufficient balance to redeem the gift voucher.
7530	The currency of the gift voucher deviates from the currency of the chain.

5.1.6 REDEEM A DISCOUNT COUPON

The operation **RedeemDiscountCoupon** on the **OnlineSalesService** can be used.

The **RedeemDiscountCouponCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
Barcode	The barcode of the discount coupon.
Value	The value to be redeemed. See **Amount** data contract.
ReceiptNumber	The same receipt number as in the receipt that will be delivered though the ticket injection (ARTS) interface.
CustomerNumber	The number of the customer that for which the discount coupon will be redeemed.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **RedeemDiscountCouponResult** is returned when the discount coupon is successfully redeemed. It includes these properties:

Property	Validations
EvoucherTransactionId	Identification of the e-voucher transaction (Guid)
NewBalance	The new balance of the discount coupon. See **Amount** data contract.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer is not found.
7535	The currency of the discount coupon deviates from the currency of the chain.
7536	Discount coupon not found.
7537	Discount coupon not linked to the chain.
7538	Discount coupon is blocked.
7539	Discount coupon is expired.
7540	Discount coupon has already been redeemed.
7541	Insufficient balance to redeem the discount coupon.
7546	Discount coupon is not issued.

7550	The discount coupon cannot be redeemed since its start date is not reached yet.
7551	Partial redeeming of the discount coupon is not allowed.

5.1.7 REDEEM A RETURN COUPON

The operation **RedeemReturnCoupon** on the **OnlineSalesService** can be used.

The **RedeemReturnCouponCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
Barcode	The barcode of the return coupon.
Value	The value to be redeemed. See **Amount** data contract.
ReceiptNumber	The same receipt number as in the receipt that will be delivered though the ticket injection (ARTS) interface.
CustomerNumber	The number of the customer that redeemed the return coupon.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **RedeemReturnCouponResult** is returned when the return coupon is successfully redeemed. It includes these properties:

Property	Validations
EvoucherTransactionId	Identification of the e-voucher transaction (Guid)
NewBalance	The new balance of the return coupon. See **Amount** data contract.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	The customer is not found.
7519	Return coupon not found.
7520	The currency of the return coupon deviates from the currency of the chain.
7521	Return coupon is blocked.
7522	Return coupon not linked to the chain.
7523	Return coupon is expired.
7525	Return coupon has already been redeemed.
7526	Insufficient balance to redeem the discount coupon.
7545	The return coupon is not issued.

5.1.8 REDEEM A DEPOSIT COUPON

The operation **RedeemDepositCoupon** on the **OnlineSalesService** can be used.

The **RedeemDepositCouponCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
Barcode	The barcode of the return coupon.
CurrencyCode	See one of the supported currency code constants.
ReceiptNumber	The same receipt number as in the receipt that will be delivered though the ticket injection (ARTS) interface.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **RedeemDepositCouponResult** is returned when the deposit coupon is successfully redeemed. It includes these properties:

Property	Validations
RedeemedValue	The value that have been redeemed for the deposit coupon. See **Amount** data contract.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
7500	The currency of the return coupon deviates from the currency of the chain.
7501	Deposit coupon not found.
7503	Deposit coupon not linked to the chain.
7504	Deposit coupon has already been redeemed.
7505	Deposit coupon is expired.
7506	Deposit coupon is blocked.

5.2 QUERY OPERATIONS

5.2.1 GET THE DETAILS OF AN E-VOUCHER

The operation **GetEvoucherDetails** on the **OnlineSalesService** can be used.

The **GetEvoucherDetailsCall** is taken as input argument. It contains these properties and validation rules:

Property	Validations
Barcode	Required.
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

A **GetEvoucherDetailsResult** is returned when an e-voucher is found for the specified barcode. An empty **GetEvoucherDetailsResult** is returned when no e-vouchers was found. The **GetEvoucherDetailsResult** includes following properties:

Property	Description
Barcode	Barcode of the e-voucher.
EvoucherTypeCode	Number identifying the type of the e-voucher. See supported e-voucher types codes.
IssuedTimestamp	Timestamp when the e-vouchers was issued.
StatusCode	Status of the e-voucher. See supported e-voucher status codes.
ActualBalance	The amount that is still available for redeeming. See **Amount** data contract.
InitialBalance	The initial balance at the moment the e-voucher was issued. See **Amount** data contract.
IsBlocked	Flag indicating if the e-voucher is blocked or not.
Transactions	List of e-voucher transactions. See **EvoucherTransaction** data contract.

No faults are defined for this operation.

5.3 DATA CONTRACTS

5.3.1 EVOUCHERTRANSACTION

These are the properties of a e-voucher transaction:

Property	Description
TransactionTypeCode	The transaction type. One of the transaction type codes.
Value	The monetary value of the transaction. See **Amount** data contract.
Timestamp	Timestamp of the transaction.

6 CUSTOMERORDER OPERATIONS

6.1 COMMAND OPERATIONS

6.1.1 PLACE WEB ORDER

The operation **PlaceWebOrder** on the **OnlineSalesService** can be used.

The **PlaceWebOrderCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations	MaxLength
OrderNumber	Required. The number of the order.	255
OrderBarcode	Optional. The barcode of the order.	255
CustomerNumber	Optional. The customer number that is used to place the order.	
CustomerCardNumber	Optional. The customer card number that is used to place the order.	128
CustomerReference	Optional. The customer reference of the customer who placed the order.	255
PickupTradematrix	Required. The tradematrix of the store where you want to pick up the order.	
PlannedPickupDate	Required. The date you plan to pick up the order.	
DeliveryTypeCode	Required. The delivery code, see codes for possible values	
InstructionsForPicker	Optional. The instructions for the picker.	255
InstructionsForCashier	Optional. The instructions for the cashier.	255
NumberOfCrates	Required. The number of crates the order should be/is packaged in.	
AllowPickAlternative	Required. Is it allowed to 'pick' an alternative article(s) (eg. replace an out of stock article with a similar article)?	
OrderLines	Required. The placed order lines. See **WebCustomerOrderLine** datacontract.	
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.	

DeliveryAddressNumber	Optional.Indicates the delivery address number selected for this order. If not filled the order will be picked up from the store.
DeliveryTime	Indicates when the customer wants to pick up his items, or have them delivered. See **DeliveryTimeFrame.**
PromopackInfo	Information about the promotions that have stamps, points or coupons given on the receipt total. See **PromopackInfo** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	Customer not found
8001	Shop could not be found
8002	Item could not be found
8003	Item barcode could not be found
8004	Customer order with the given number already exists

6.1.2 PLACE IN SHOP ORDER

The operation **PlaceInShopOrder** on the **OnlineSalesService** can be used.

The **PlaceInShopOrderCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations	Maxlength
OrderNumber	Required. The number of the order.	255
OrderBarcode	Optional. The barcode of the order.	255
CustomerNumber	Optional. The customer number that is used to place the order.	
CustomerCardNumber	Optional. The customer card number that is used to place the order.	128
CustomerReference	Optional. The customer reference of the customer who placed the order.	255
PickupTradematrix	Required. The tradematrix of the store where you want to pick up the order.	
PlannedPickupDate	Required. The date you plan to pick up the order.	
DeliveryTypeCode	Required. The delivery code, see codes for possible values	
InstructionsForPicker	Optional. The instructions for the picker.	255
InstructionsForCashier	Optional. The instructions for the cashier.	255
NumberOfCrates	Required. The number of crates the order should be/is packaged in.	
AllowPickAlternative	Required. Is it allowed to 'pick' an alternative article(s) (eg. replace an out of stock article with a similar article)?	
OrderLines	Required. The placed order lines. See **InShopCustomerOrderLine** datacontract.	
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.	
DeliveryAddressNumber	Indicates the address number of the delivery address	
DeliveryTime	Indicates when the customer wants to pick up his items, or have them delivered. See **DeliveryTimeFrame.**	
PromopackInfo	Information about the promotions that have stamps, points or coupons given on the receipt total. See	

PromopackInfo datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
1000	Customer not found
8001	Shop could not be found
8002	Item could not be found
8003	Item barcode could not be found
8004	Customer order with the given number already exists

6.1.3 CANCEL OBSOLETE ORDERS

The operation **CancelObsoleteOrders** on the **OnlineSalesService** can be used.

The **CancelObsoleteOrdersCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.

Nothing is returned when the update succeeded.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.
8000	The customer order is not found.

6.2 QUERY OPERATIONS

The operation **GetPricesForWebOrder** on the **OnlineSaleService** can be used.

A **GetPricesForWebOrderCall** is taken as input argument. It contains these properties and validation rules:

Property	Validations	Maxlength
OrderNumber	Required. The number of the order.	255
OrderBarcode	Optional. The barcode of the order.	255
CustomerNumber	Optional. The customer number that is used to place the order.	
CustomerCardNumber	Optional. The customer card number that is used to place the order.	128
CustomerReference	Optional. The customer reference of the customer who placed the order.	255
PickupStore	Required. The tradematrix of the store where you want to pick up the order.	
OrderLines	Required. The placed order lines. See **WebOrderLine** datacontract.	
OrganizationHierarchies	Required. Identification of the store. See **OrganizationHierachies** datacontract.	

A **GetPricesForWebOrderResult** is returned when the calculation of prices is done for given items taking into consideration promotions applied. The **GetCustomerDetailsResult** includes following properties:

Property	Validations	Maxlength
OrderNumber	Required. The number of the order.	255
OrderBarcode	Optional. The barcode of the order.	255
CustomerNumber	Optional. The customer number that is used to place the order.	
CustomerCardNumber	Optional. The customer card number that is used to place the order.	128
CustomerReference	Optional. The customer reference of the customer who placed the order.	255
PickupStore	Required. The tradematrix of the store where you want to pick up the order.	
PricesForWebOrderLine	Required. The placed order lines. See **PriceWebOrderLine** datacontract.	
PromopackInfo	Required. Information about the promotions that have stamps, points or	

	coupons given on the receipt total. See **PromopackInfo** datacontract.

6.3 DATA CONTRACTS

6.3.1 WEBCUSTOMERORDERLINE

These are the properties for web customer order line:

Property	Description
ArticleNumber	Required. The article number of the ordered item.
BarcodeNumber	Required. The barcode number of the ordered item.
LineNumber	Required. A sequence number for the ordered items. Will be used as picking order.
QuantityOrdered	Required. The quantity to be ordered.
QuantityUnitOrdered	Required. The unit (measure) quantity to be ordered. To be filled in for measured articles otherwise 0.
PriceInclVatOrdered	Required. The price inclusive vat of the ordered item.

6.3.2 INSHOPCUSTOMERORDERLINE

These are the properties for an in shop customer order line:

Property	Description
ArticleNumber	Required. The article number of the ordered item.
BarcodeNumber	Required. The barcode number of the ordered item.
LineNumber	Required. A sequence number for the ordered items. Will be used as picking order.
QuantityOrdered	Required. The quantity to be ordered.
QuantityUnitOrdered	Required. The unit (measure) quantity to be ordered. To be filled in for measured articles otherwise 0.
PriceInclVatOrdered	Required. The price inclusive vat of the ordered item.

6.3.3 DELIVERYTIMEFRAME

Property	Description
BeginTime	Indicates the start of delivery time frame.
EndTime	Indicates the end of delivery time frame.

6.3.4 WEBORDERLINE

Property	Description
ArticleNumber	Required. The article number of the ordered item.
BarcodeNumber	Required. The barcode number of the ordered item.
LineNumber	Required. A sequence number for the ordered items. Will be used as picking order.
Quantity	Required. The quantity to be ordered.
QuantityUnit	Required. The unit (measure) quantity to be ordered. To be filled in for measured articles otherwise 0.

6.3.5 PRICEWEBORDERLINE

Property	Description
ArticleNumber	Required. The article number of the ordered item.
BarcodeNumber	Required. The barcode number of the ordered item.
LineNumber	Required. A sequence number for the ordered items. Will be used as picking order.
Quantity	Required. The quantity to be ordered.
QuantityUnit	Required. The unit (measure) quantity to be ordered. Should be filled only for measured articles otherwise 0.
PriceInclVat	Required. Actual price per item
PromopacksDiscounts	A list containing the promopack discounts given by the promotions. See **PromopackDiscount** datacontract.

6.3.6 PROMOPACKDISCOUNT

Property	Description
PromopackNumber	Required. The promopack number of the promotion.
Discount	Required. The discount given by the promotion.
Description	Required. The description of the promotion that generates the discount.

6.3.7 PROMOPACKINFO

Property	Description
Points	Optional. List of **PointRegistration**. Will be filled in with the list of points given by the promotions. These points will be returned on the order level. See **PointRegistration** contract**.**
Vouchers	Optional. List of **PromopackVoucherRegistration**. Will be filled in with the list of vouchers given by the promotions. These vouchers/coupons will be returned on the order level. See **PromopackVoucherRegistration** contract**.**

6.3.8 POINTREGISTRATION

Property	Description
PromopackNumber	Required. Indicates promotion number.
ChainNumber	Required. Used to identify a chain.
PromopackDescription	Required. Promotion description.
PointTypeNumber	Required. The number of the pointType.
NumberOfPoints	Required. The number of points that the promotions generates.
Reason	Required. The description for the promotion that generates the points.

6.3.9 PROMOPACKVOUCHERREGISTRATION

Property	Description
PromopackNumber	Required. Indicates promotion number.
ChainNumber	Required. Used to identify a chain.
PromopackDescription	Required. Promotion description.
Number	Required. The voucher number that identifies the voucher generated by the promotions.
Value	Required. The value of the voucher generated by the promotions.

7 TICKET OPERATIONS

7.1 COMMAND OPERATIONS

7.1.1 REGISTER AN ARTS DIGITAL RECEIPT

The operation **RegisterArtsDigitalReceipt** on the **OnlineSalesService** can be used to register an receipt in the ARTS DIGITAL RECEIPT format in OBP.

The **RegisterArtsDigitalReceiptCommand** is taken as input argument. It contains these properties and validation rules:

Property	Validations
ArtsDigitalReceiptXml	Required. The 'Arts DigitalReceipt'. This should be a string representing an UTF8 encoded XML.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.

7.2 QUERY OPERATIONS

7.2.1 GET TICKET DETAILS

The operation **GetTicketDetails** on the **OnlineSalesService** can be used.

The **GetTicketDetailsCall** is taken as input argument it contains these properties and validation rules.

Property	Validations
TicketReference	Required.

A **GetTicketDetailsResult** is returned when a ticket was found for the given input. An empty result is returned when no ticket could be found. The **GetTicketDetailsResult** includes the following properties:

Property	Description
VisualRepresentation	A list of textlines representing a visual, human readable representation of a ticket, comparable to the lines printed on a physical ticket.

An **OmniChannelFault** is returned when the update failed. Following error codes can be returned:

Error code	Description
1	Validation error.

7.3 DATA CONTRACTS

7.3.1 TICKET REFERENCE

These are the properties for a ticket reference

Property	Description
TicketId	The identification of a ticket
RegistrationTimestamp	The registration timestamp of a ticket

8 COMMON DATA CONTRACTS

8.1 AMOUNT

These are the amount validation rules:

Property	Validations
Value	Required, decimal.
CurrencyCode	Required, should be one of the supported currency codes.

8.2 ORGANIZATION HIERACHIES

This is a list of nodes used to identify the node within the organization hierarchy. Only stores are supported.

These are the properties of an organization hierarchy:

Property	Validations
Level	Required, should be one of the supported hierarchy levels.
Id	Optional. Identification of the hierarchy.
ExternalReference	Optional. External reference of the store

There are two possible ways to provide identification of a store. The default way is to provide it similar to the 'tradematrix' format (chain.storenumber).

Example: tradematrix 01.0320 results in this list of organization hierarchies:

- Id = 1; Level = Concept
- Id = 320; Level = RetailStore

The second possibility is to use the 'ExternalReference' of a store. This method of providing the store identification can be activated by changing a parameter in OBP. When this parameter is active, the OrganizationHierarcy list should only contain one element of type 'RetailStore' and with the 'ExternalReference' filled in. This will be used to find the store in OBP.

Example: store with externalreference 010320 results in this list of organization hierarchies:

- Level = RetailStore; ExternalReference = 010320

9 CODES

9.1 LANGUAGES CODES

The ISO-639-1 and ISO-639-2 languages codes are used. Some languages have alternative codes for bibliographic or terminology purposes. In this case you should use the terminology code.

9.2 GENDER CODES

The ARTS gender codes are used in the interface. Following genders are supported:

Code	Description
Male	Male
Female	Female
Unknown	Unknown

9.3 COUNTRY CODES

The ARTS country codes are used in the interface. Following countries are supported:

Code	Description
BE	Belgium
FR	France
NL	Netherlands

9.4 CURRENCY CODES

The ARTS currency codes are used in the interface. Following countries are supported:

Code	Description
EUR	Euro

9.5 E-VOUCHER TYPE CODES

Following e-voucher types are supported:

Code	Description
gift	Gift voucher
deposit	Deposit coupon
discount	Discount coupon
return	Return coupon

9.6 E-VOUCHER STATUS CODES

Following e-voucher statuses are supported:

Code	Description
available	The e-voucher is available to be issued (only used for gift vouchers).
issued	The e-voucher has been issued. It can be used for redeeming. Possibly it is already partially redeemed.
redeemed	The e-voucher is fully redeemed.

9.7 E-VOUCHER TRANSACTION TYPE CODES

Following e-voucher statuses are supported:

Code	Description
create	The e-voucher has been created.
open	The initial balance of the voucher has been set.
redeem	An amount has been redeemed.
cancel	A transaction has been cancelled.

9.8 ORGANIZATION HIERARCHIES

The ARTS organization hierarchies are used. Following organization hierarchies are supported:

Code	Description
Concept	Used to identify a chain.
RetailStore	Used to identify a store.

9.9 DELIVERY TYPE CODES

Following delivery types are supported:

Code	Description
store	Order will be delivered to/retrieved from the store
home	Order will be delivered to the home of the customer