Priority APICORE – Integration Guide

API Version 2.0

October 2017

1. The API general description

Priority Retail suite provides a comprehensive package of services for various vectors of retail businesses. The package consists of ERP services, such as logistics, stock, finance, accounting, loyalty, mobile and desktop POS management, e-com, CRM and more.

The REST API facilitates the easy, safe and intact integration of Priority Retail Suite with external services i.e. e-com websites, CRM systems, Marketing automation systems etc., whilst enhancing these channels using the core functionality of the Priority retail suite.

The API includes various queries and insert methods transferring data to the core retail system, exposing business logic which enables the satellite systems to integrate with the core business logic and enhancing them to provide similar customer experience in all business channels creating a true Omni-channel system. The capability to make use of the retail suite core business logic, provides a better alternative to developing the business logic separately in each of the channels. This might result in a much expensive, unidentical and inferior solution.

The entire data required for the API services is stored in a local DB installed on the API server. The data is kept up to date constantly with the enterprise DB by a synchronization service. This architecture provides better accessibility and off-line capability i.e. the API will continue providing services to critical channels during any system disaster, communication failure or maintenance down-time.

2. API services repository and description

Subsidiary to this document, the API server provides on-line documentation available by the following link: https://[Server_Address]/[Instance_Name]/swagger/ui/index

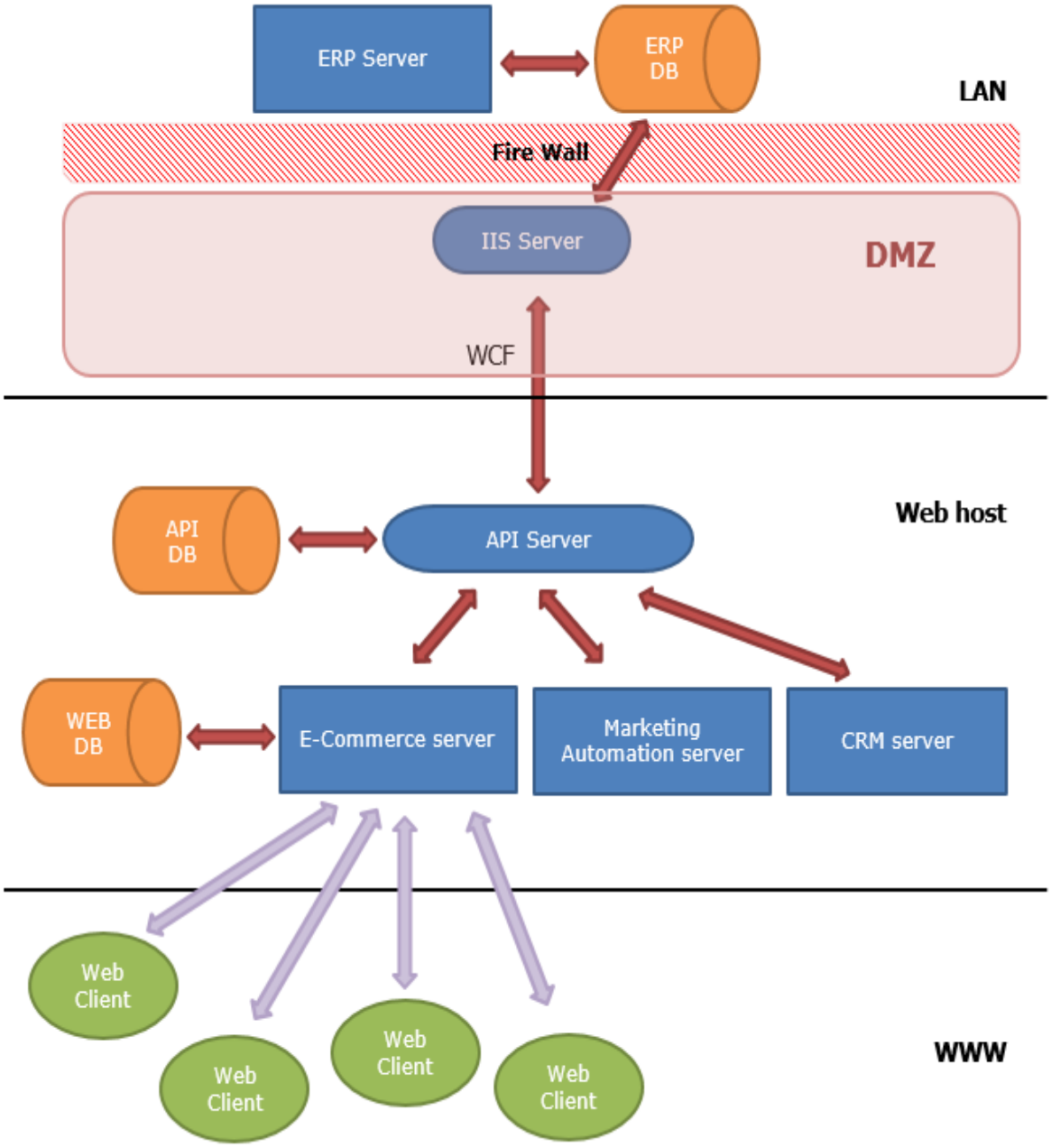
The on-line documentation provides information on all available methods, the required approach and parameters for each method, the returned structure and error list.

www.priority-software.com

info@priority-software.com | +972 3-925-1000

2 Ha-Amal St., Afek Industrial Park, Rosh Haayin, Israel

3. The API environment architecture



priority retail
www.priority-software.com
info@priority-software.com | +972 3-925-1000
2 Ha-Amal St., Afek Industrial Park, Rosh Haayin, Israel

4. The channel registration and identification process

Before any request to the API server, the approaching channel must identify and register itself to the API server, using the registration method: ***RegisterPos***.

The registration method should be provided with the following parameters:

- The branch code to which the channel is attached
- The lane code
- Unique identifier of the channel manufacturer – Issued once by Priority per manufacturer.
- Channel unique identifier – Identify the specific client.
- Product type – The type of service describing the channel. The API verifies that the channel type is included in the channel license.

After successful registration, every approach to any of the API's methods should include the client unique identification object.

5. On-line methods

- This set of methods has the word "online" in their description.
- These methods get access to the main DB directly, the ERP DB, thus its availability is required for these methods to operate.
- The methods which are not stated "online" are making use of the local API DB and therefore available and operates in off-line state.

6. The methods types

The methods are of four types, deferring by the type of data transferred by methods:

- Query/Search methods for a specific record or result list i.e. customer search.
- Queries for chunks of records, intended to retrieve massive data divided to chunks i.e. all customers updated from a certain date.
- Insert/register static data i.e. the data transferred will be stored in the DB as is and the channel will receive success/error indication.
- Calculated data insert – the data transferred by these methods will be processed on-line by the API retail business logic engine and the result returned to the channel will include processed data with all the business logic implications inflicted on the original data sent to the method. For example; calculating an order with all sale campaigns defined in the API engine and returning the order data with the effected prices.

7. Integrating to the API by channel

priority retail
www.priority-software.com
info@priority-software.com | +972 3-925-1000
2 Ha-Amal St., Afek Industrial Park, Rosh Haayin, Israel

a. BASIC

The API BASIC module is the basic model which form the infrastructure set of methods for any other module. It is composed of master data queries for items, item parameters, customers, stock levels, price lists and more... The DATA module enables registration of customer order documents with a new customer record. This module does not support loyalty data.

The order can contain payment methods and credit card pre-authorization for actual charge upon supply. This module provides a workable solution for e-com sites which do not support loyalty and the retail suite sale campaigns. Furthermore, this module does not support complexed items such as sets, serial devices and trigger items that operates the retail suite various business logics.

b. Pro

The API Pro module facilitates the registration and update of POS and commercial customers make queries on customers' data such as orders and invoices and make queries on composite items like sets and serial devices.

c. CLM

The methods in this module serves the customer loyalty channels, both queries and insert methods. The module facilitates the implementation of Priority retail CLM as a complete solution or as a complimentary for e-com sites which provide CLM basic capabilities. It is also the gateway for marketing automation systems and CRM systems to the customers' assets. The methods included under the CLM retrieves customers' data, customers' assets such as loyalty credits and eligibilities and facilitates the creation of new assets for loyalty members. Customers' segmentation and loyalty sub plans are also supported.

d. CRM

The CRM provides reading and writing of service calls from and to the retail and commercial CRM modules. A new service call can be recorded with a new customer record and serial devices and return service calls history.

e. CART

The CART module or as it may be otherwise called; "Quote", is substantially different from the previous modules, as it is the gateway to the retail module API business logic engine.

However, the Cart module is totally depending on the previous modules i.e. Basic, Pro & CLM, to provide the master data and the parameters required to operate the complexed functionality it comprises.

priority retail

www.priority-software.com

info@priority-software.com | +972 3-925-1000

2 Ha-Amal St., Afek Industrial Park, Rosh Haayin, Israel

The Cart is processing the data provided by the external channel through the retail business logic engine and returns the processed data after sale campaigns calculation, triggers operation and customer's loyalty assets effect on the transaction. This is done through a set of methods which are operated in a chronologic order as described hereafter.

8. Calculated transaction chronology (CART / POS)

As above said, the **CART** module is the most intimate integration between the API and the external channel, as this set of methods operates the retail business logic engine and must maintain all the rules and validations managed by the business logic engine. The **CART** is the real Omni-channel facilitator for the on-line and off-line channels.

Following is the order in which the **CART** methods should be operated:

9.1 Retrieving the master data

The first stage should be to retrieve the items, customers, prices and loyalty data through the Basic, Pro & CLM modules, store it at the client side and use it to create a valid transaction.

9.2 Registered customer

Every call to the **CART** methods should include a POS customer entity. This can be achieved in one of the following ways:

9.2.1 If the transaction is for an existing customer, the POS customer number should be provided to the method.

9.2.2 If the customer is not yet registered in the retail system, the create customer method should be successfully operated and a POS customer number should be achieved, for the further operation of the Cart methods.

9.2.3 In case the transaction is made for an unidentified or unregistered customer, the general customer of the API branch should be sent in the customer parameter. Another option is to use the ***RegisterByGeneralPosCustomer*** flag in the methods parameters and the API will populate the customer number field.

9.3 CART calculation methods

The Cart calculation and registration methods, should be operated in a certain chronological and logical order. Each stage of the transaction is facilitated by a designated method.

9.3.1 Open transaction

This method creates a parallel Cart in the API DB to the cart handled by the external channel.

priority retail
www.priority-software.com
info@priority-software.com | +972 3-925-1000
2 Ha-Amal St., Afek Industrial Park, Rosh Haayin, Israel

The external channel indicates to the API if the transaction is intended to be an order or is immediate supply transaction. This is flagged to the API by the "IsOrder" parameter.

The external channel is to provide its unique identifier for the cart or quote for which it has requested to open the transaction in the API. The unique cart identifier should be provided in the "ExternalOrderNumber" parameter.

Once the *OpenTransaction* method is successfully approached, it will return a temporary transaction number that uniquely refers to the Cart created in the API DB and is handled solely for the specific Cart for which it was designated in the external channel.

From now on, every approach to any of the following cart methods, should be accompanied with the temporary transaction number returned by the Open Transaction method. Not providing the correct temporary transaction number might result in a fatal error.

9.3.2 Update Transaction

Through this method items are added or removed from the transaction. The transaction is recalculated for sale campaigns, coupons, credits and all the business rules. This method returns a complete cart with all the sub entities of the cart i.e. transaction items repository, customer information, customer assets update etc.
The external channel should reflect the updated transaction to the final user on the client side.
The status of the transaction with all its contents can be retrieved at any time through the *GetTemporaryTransaction* method.

9.3.3 Add coupons and use loyalty credits

Adding a coupon to the transaction and using the loyalty credits is possible at any time during the life cycle of the transaction unless the close transaction method has not been executed.
This can be executed by the *AddCoupn* method and by populating the parameter *TotalRequestedUsePoints.*
The use of coupons and loyalty points, however, must be executed before the Approve transaction phase after which the customer can pay with his gift cards and credit vouchers.

9.3.4 Gift cards and credit vouchers execution

priority retail
www.priority-software.com
info@priority-software.com | +972 3-925-1000
2 Ha-Amal St., Afek Industrial Park, Rosh Haayin, Israel

The gift cards and vouchers payment should be executed before the checkout stage. This is the last stage before the credit card transaction which will conclude the checkout process.

9.3.5 Cart validation before payment

Since the customer might leave the cart for an unknown period, during which a change might occur in his assets value and validity, the cart must be validated for campaigns, assets and loyalty data. Therefore, the cart validation method should be executed.

9.3.6 2PC to customer's assets

The 2 phase commit is executed to record the change in the customer's assets which were until this stage recorded as temporary transaction. The *ApproveTransaction* method locks the customer's assets so they are not accessible to any other channel but the channel which executed the lock. Thus, it is essential to execute this method before moving forward to the payment by credit card / PayPal.

Note! After the Approve execution, no changes can be made in items, coupons, loyalty credits or any other part of the transaction.

9.3.7 Credit card payment and transaction conclusion

At this stage, the API has no role, however, it is important to emphasize that at this stage the external channel charges the customer for the balance to pay in the transaction and after this stage the transaction should be finalized.

9.3.8 Cart closing

After executing all the customer's assets the channel continue to receiving the funds for the balance by credit card or PayPal. Upon payment successful conclusion, the ***CloseTransaction*** method should be executed. The methods should receive all the data for the cart and the payment methods which were used to finalize the deal. This method closes the transaction which is the trigger to creating a customer's logistic order.

Note! - At this stage, the API matches the data stored in the temporary transaction in the API DB with the data sent to the CloseTransaction methods. Any discrepancies between the two data sets, will create an error, therefore, no changes should be made in the cart between the Approve and the Close stages.

priority retail
www.priority-software.com
info@priority-software.com | +972 3-925-1000
2 Ha-Amal St., Afek Industrial Park, Rosh Haayin, Israel

In case a change is required, the used assets must be released by the ***RollBack*** method and the cart calculation and assets execution stages must be repeated, since the cart value might has been changed.

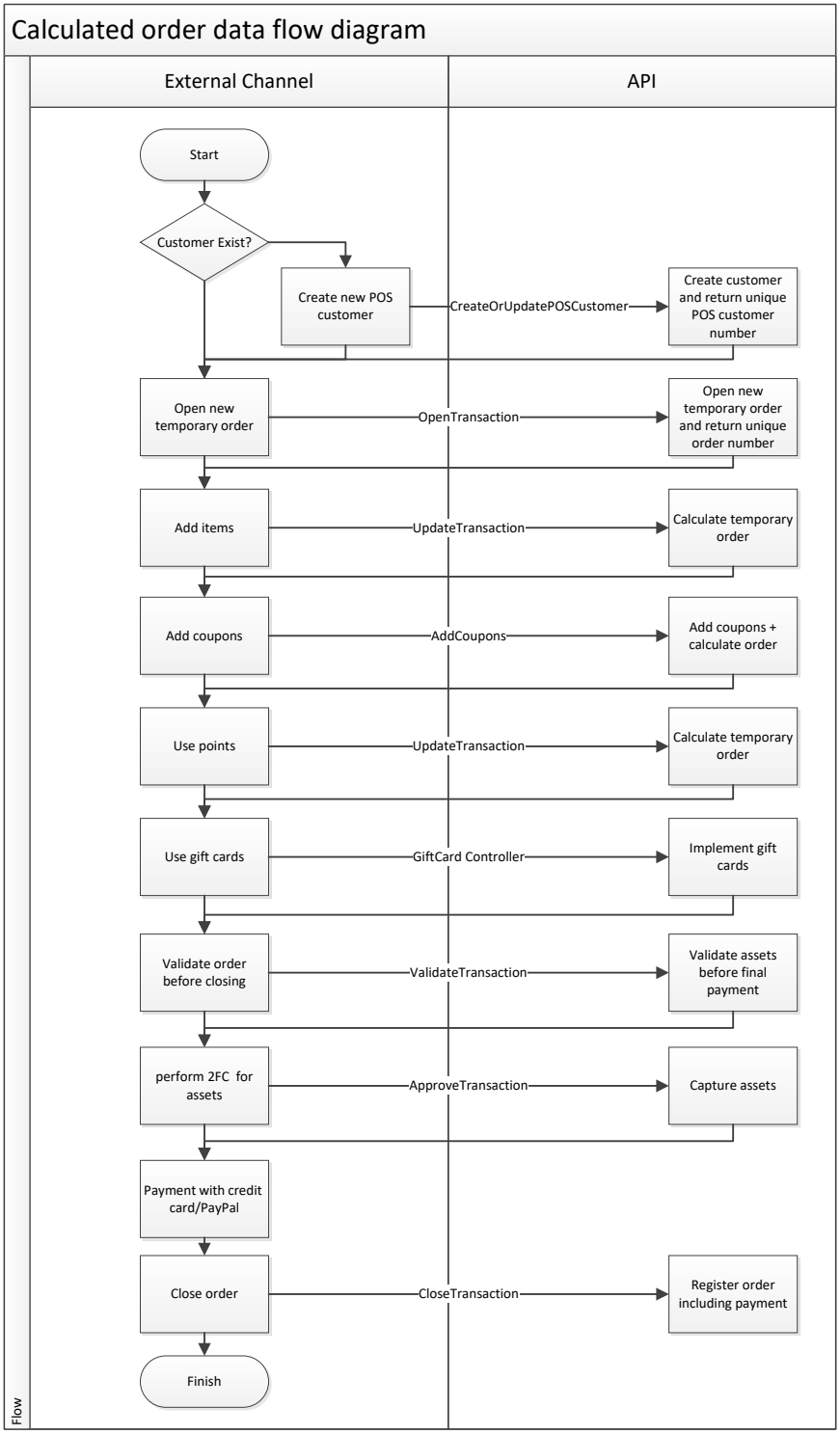
priority retail

www.priority-software.com

info@priority-software.com | +972 3-925-1000

2 Ha-Amal St., Afek Industrial Park, Rosh Haayin, Israel

9. Calculated order data flow diagram (CART / POS)



priority retail
www.priority-software.com
info@priority-software.com | +972 3-925-1000
2 Ha-Amal St., Afek Industrial Park, Rosh Haayin, Israel