**Required Setup for API Outbound (v1.1)**

Last Updated Date:29th July 2019

1. **Schema:**

Make sure the following tables are available in the database. As on 29thJuly, 2019, the schema (.df files) can be taken from the attachments on zoho ticket# 36026.

* APIOutbound – This table holds configuration required for Outbound APIs
* APIOutboundDetail – This table is an extension to APIOutbound table and holds configuration required for Outbound APIs, when request data has nested JSON
* APIOutboundEvent – This table keeps track of all Outbound events and serves as a log as well

1. **API Configuration data:**

Make sure the following tables has the required configuration data.

* APIOutbound
* APIOutboundDetail

1. **cURL utility:**

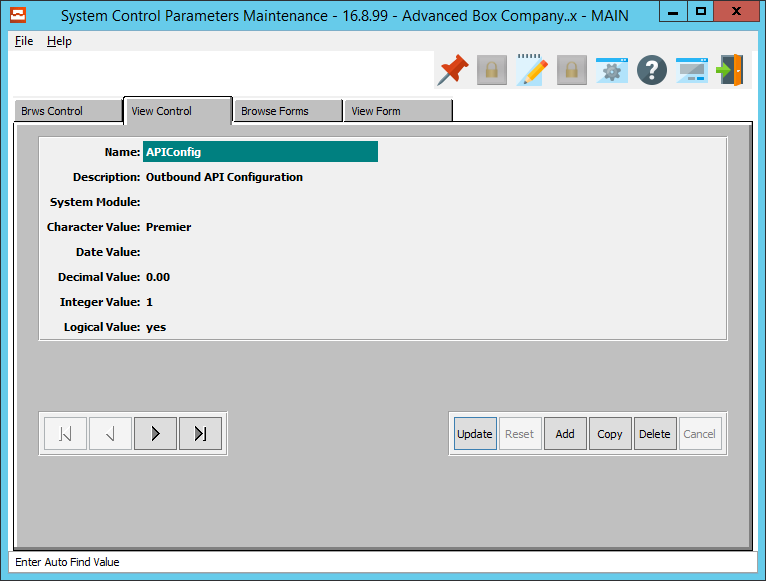
Make sure executable of cURL (32-bit) utility is available in the PROPATH preferably under

“Resources” directory.

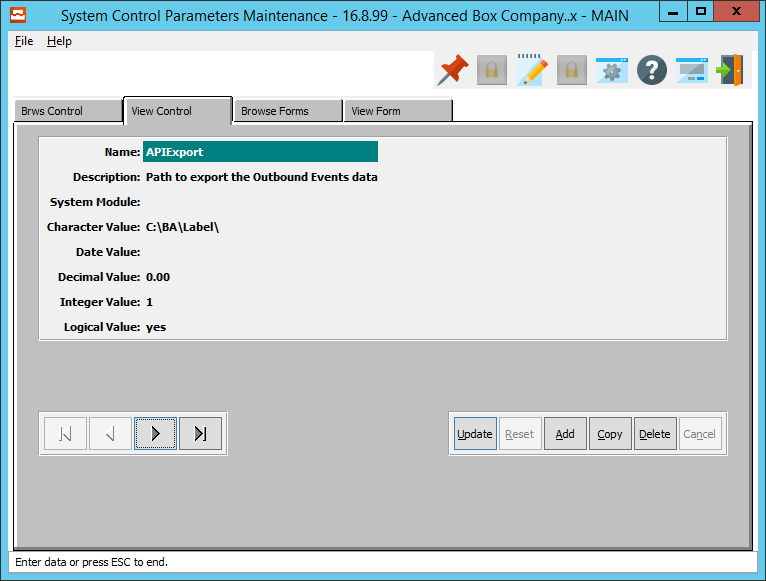
1. **sys-ctrl Configuration:**

Make sure to create the following sys-ctrl records using System Control Parameter Maitenance (N-K-1) as shown below. Note that these sys-ctrl records will be automatically created if one does not exist.

* This is required to validate the request and response handler codes with the program names available in APIOutbound configuration.



* This is required to fetch the directory to which data needs to be exported from the API Outbound Events Viewer screen.

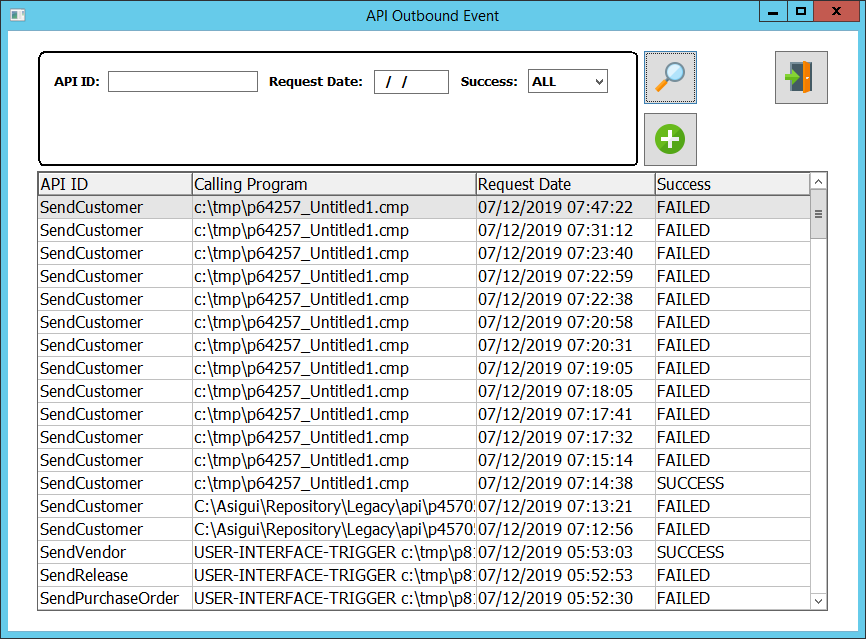


**Using API Outbound Tester for Outbound APIs directly**

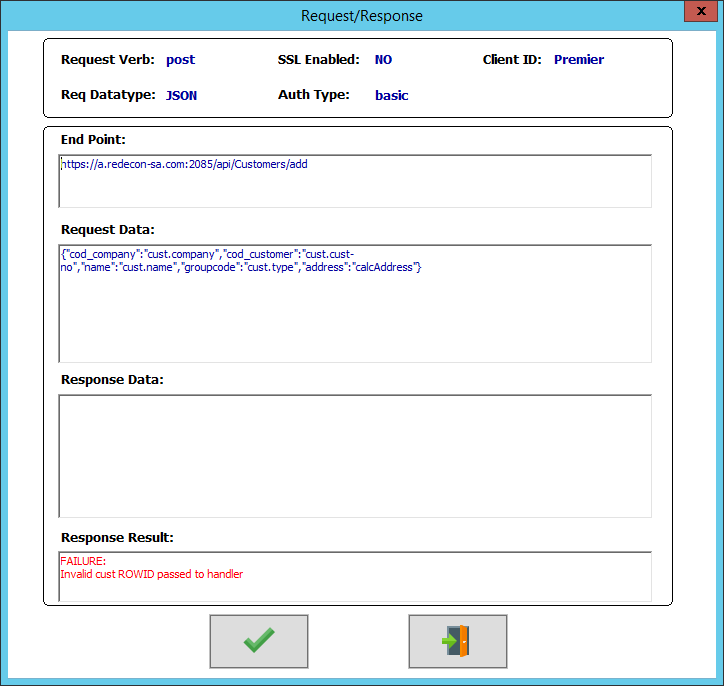
**API Outbound Viewer program:**

The program (api/APIOutboundEventsViewer.w) can be launched directly or can be setup to be run from a menu. The following is the example how the screen looks like.

This screen lists all the API Outbound events occurred within the application



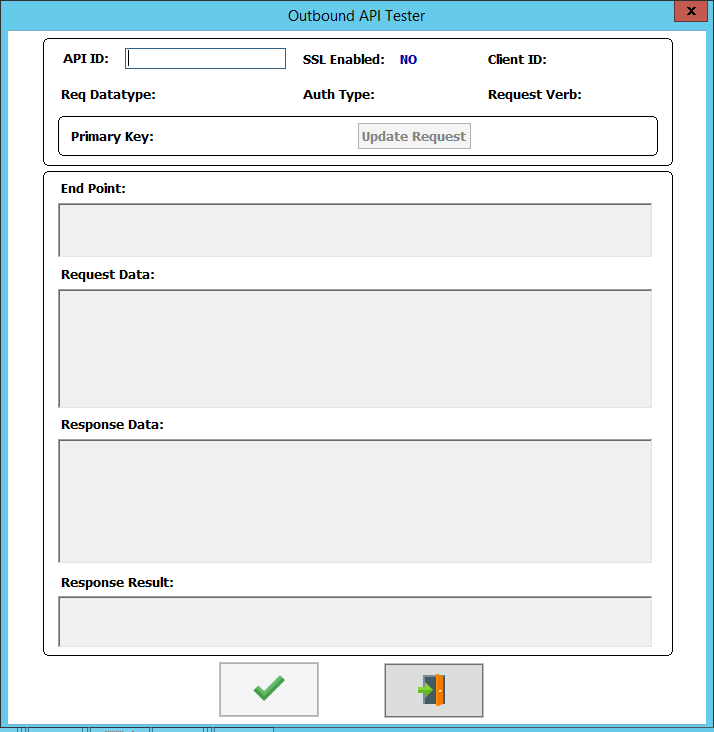
If you double-click on a row more details will be displayed about the event as below.



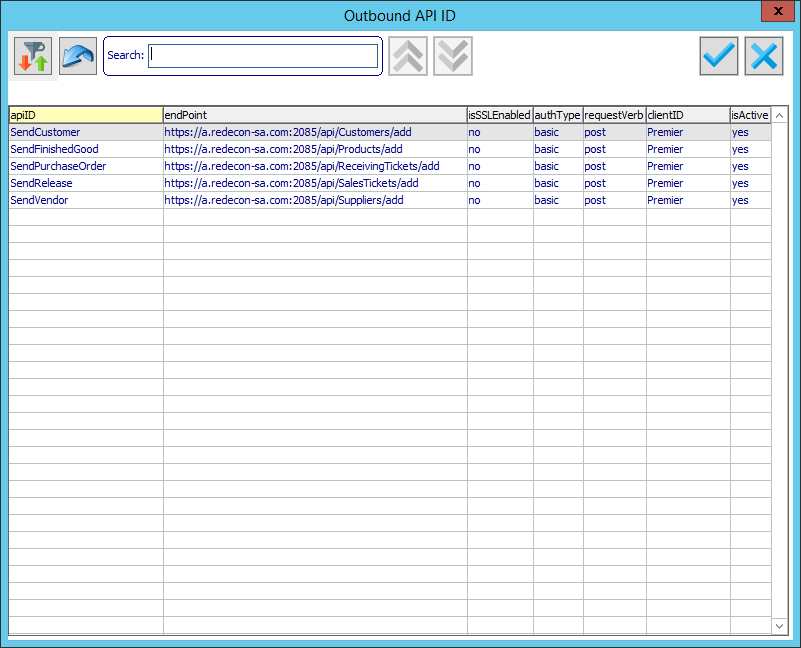
**API Outbound Tester program:**

The program (api/APIOutboundTester.w) can be launched by clicking the “+” button in the API Outbound Viewer screen. This screen can be used to make the Outbound API calls for API IDs. This is for testing purpose only and should be disabled in production.

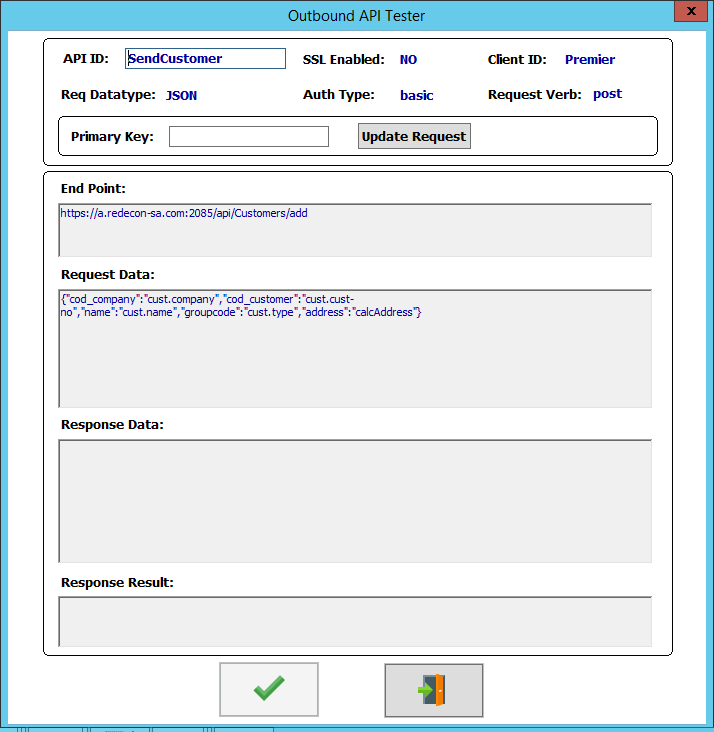
The following is the example how the screen looks like.



“F1” lookup on “API ID” field will launch the lookup for “Outbound API ID”.



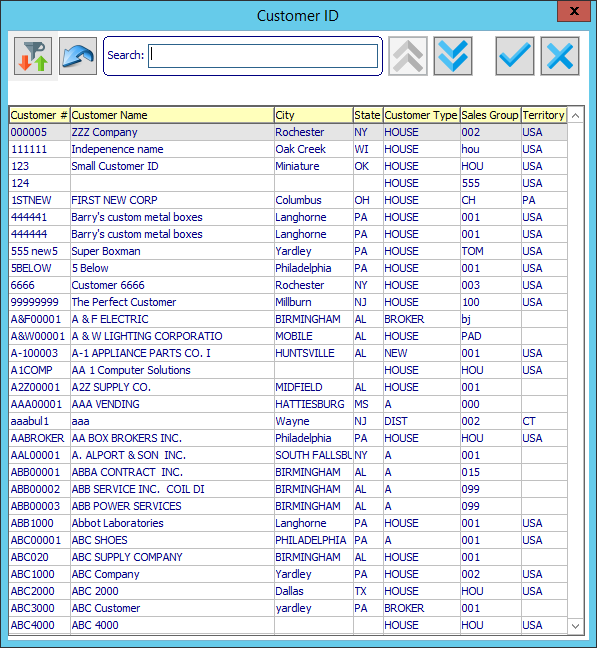
Once a valid API ID is provided in the “API ID“ field, “Primary Key” field and “Update Request” button will be enabled. “F1” lookup on “Primary Field” will launch different lookup screens according to the API ID entered.



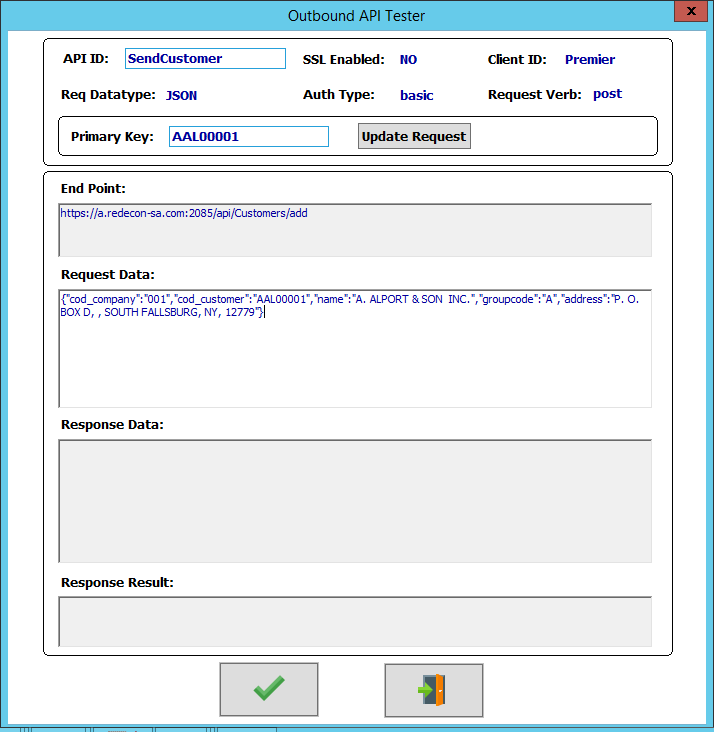
**The following are the steps to test individual API IDs:**

**“SendCustomer”:**

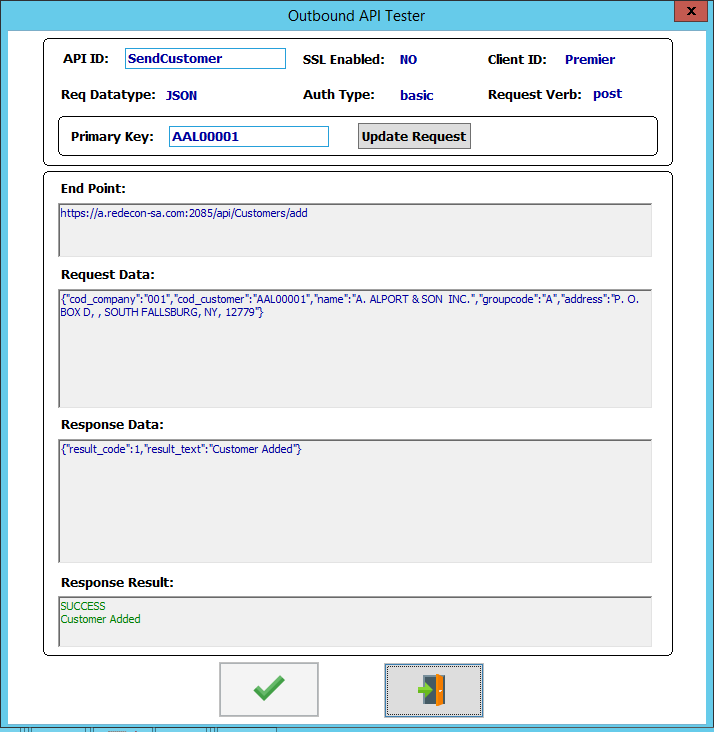
1. Enter “SendCustomer” in the “API ID” field or select from “F1” lookup on “API ID” field. “F1” lookup on the “Primary Key” will now launch the “Customer” lookup screen.



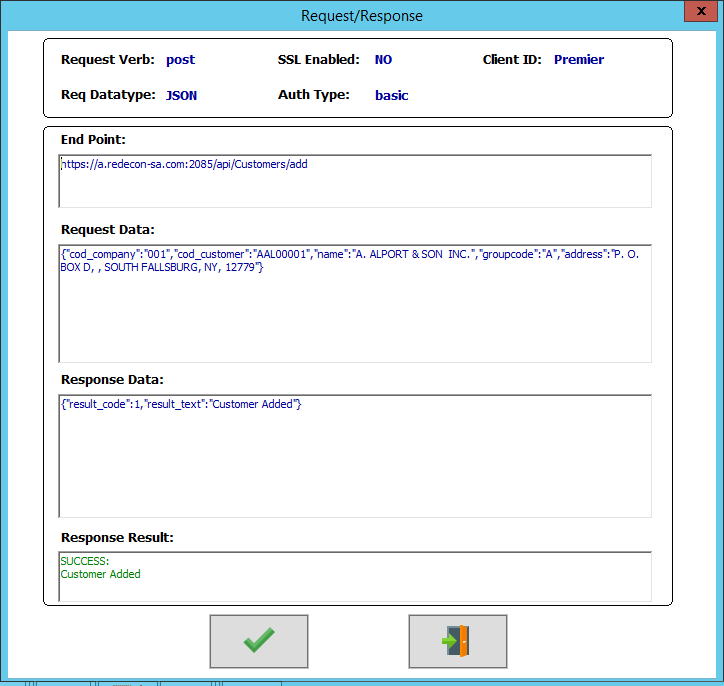
1. Selecting a row from the “Customer” lookup screen will populate the request data for the selected record. User can also manually enter a valid customer number in the “Primary Key” and click the “Update Request” button to populate the request data for the customer record. User can further modify the request data manually before calling the endpoint.



1. Clicking the “Submit” (button with Tick mark at the bottom) button will make the Outbound call to the endpoint and updates the “Response Data” and “Response Result” fields with response received from the Outbound call. “Response Result” will either update with a “Failure” or “Success” response. **The same applies to all other Outbound APIs.**

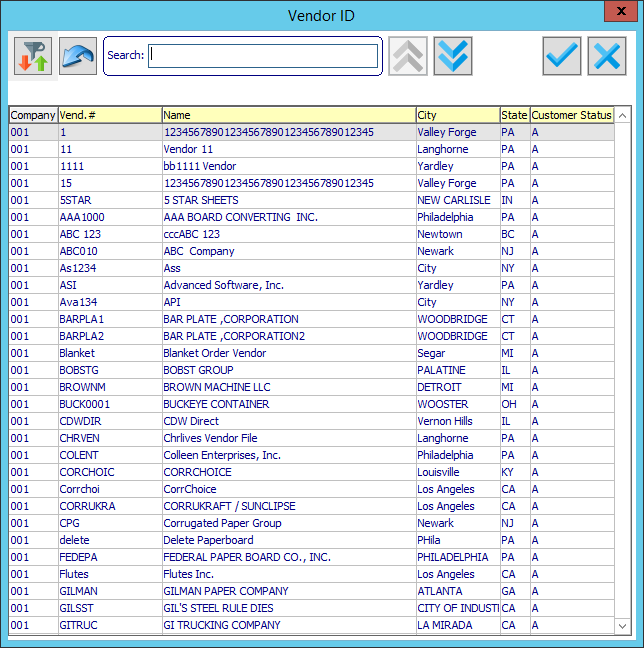


1. The event will also be available in the **API Outbound Event Viewer** screen for any future references. **The same applies to all other Outbound APIs.**

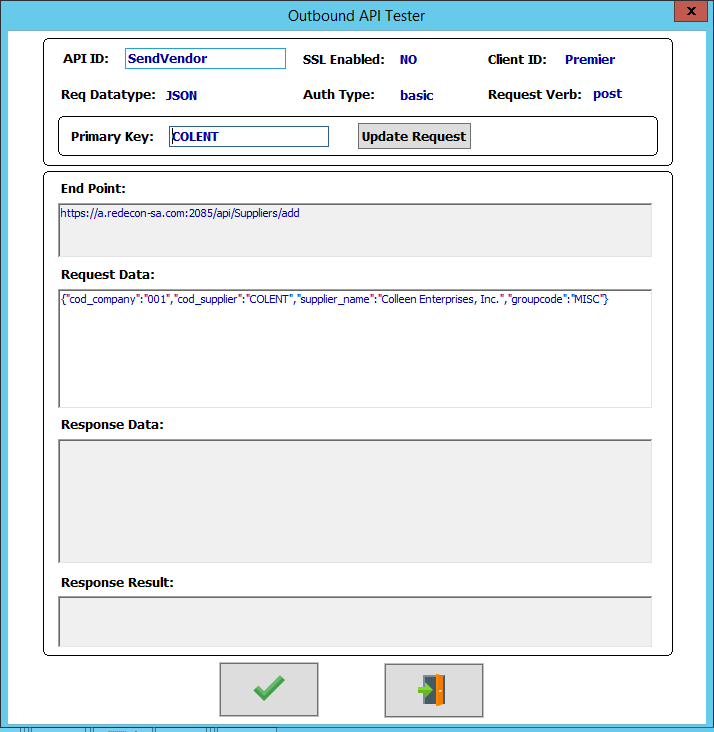


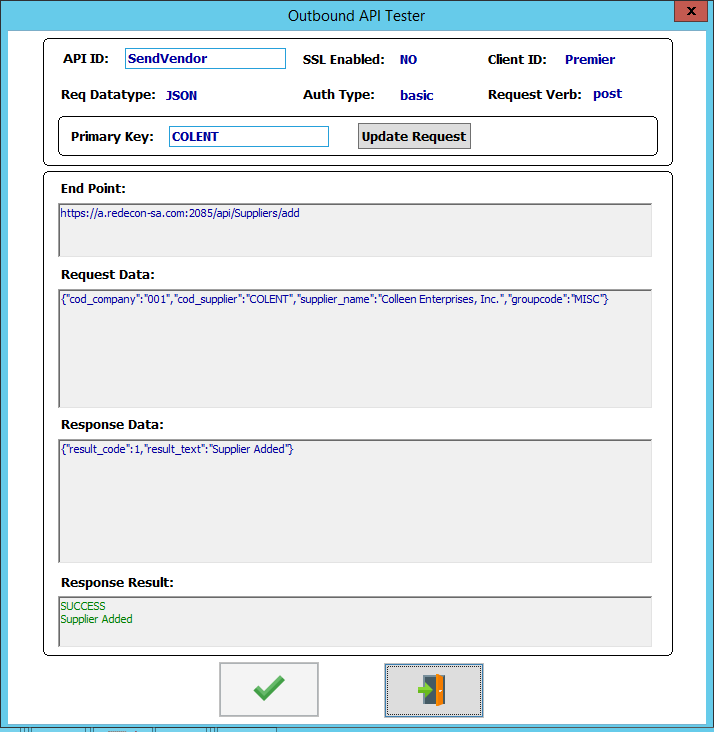
**“SendVendor”:**

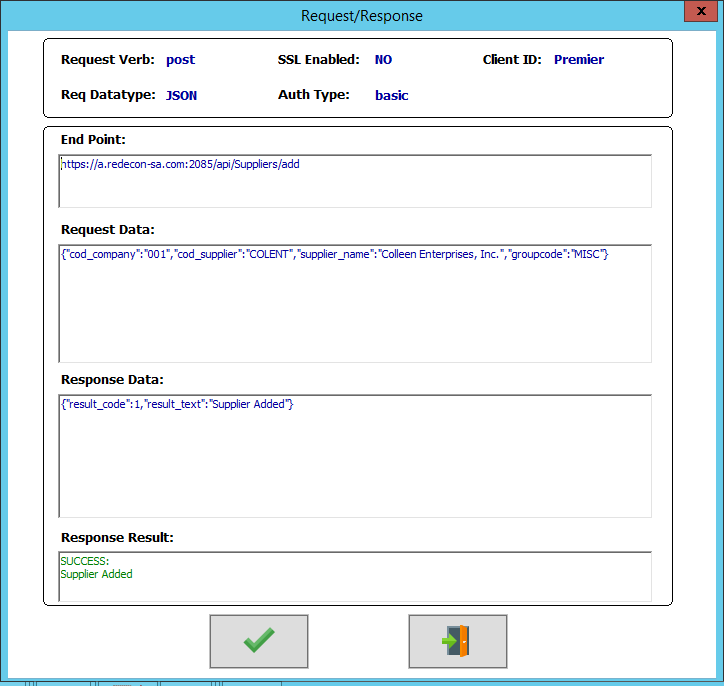
1. Enter “SendVendor” in the “API ID” field or select from “F1” lookup on “API ID” field. “F1” lookup on the “Primary Key” will now launch the “Vendor” lookup screen.



1. Selecting a row from the “Vendor” lookup screen will populate the request data for the selected record. User can also manually enter a valid vendor number in the “Primary Key” and click the “Update Request” button to populate the request data for the vendor record. User can further modify the request data manually before calling the endpoint.

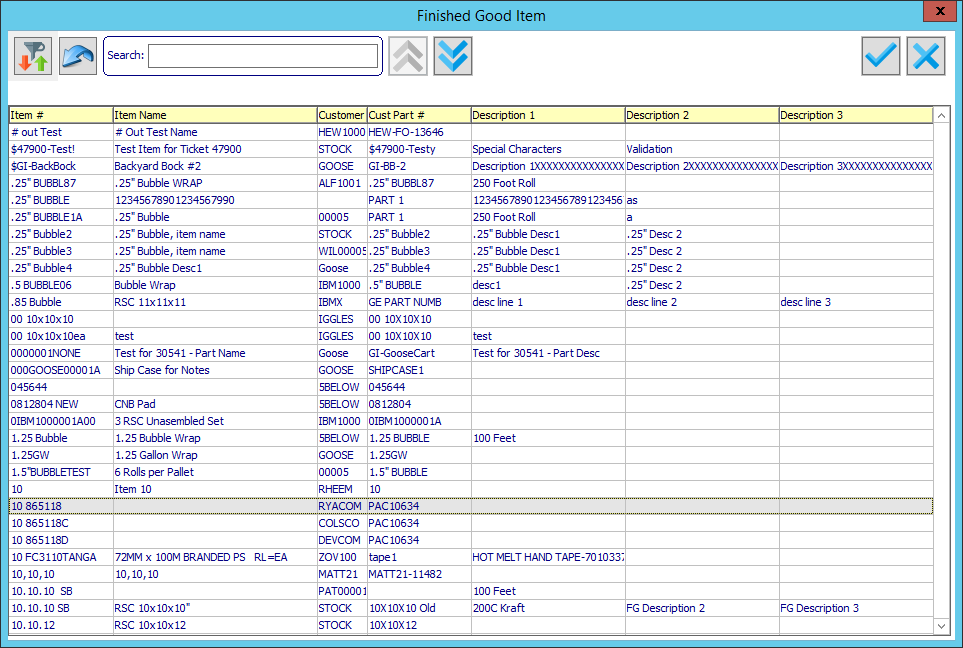




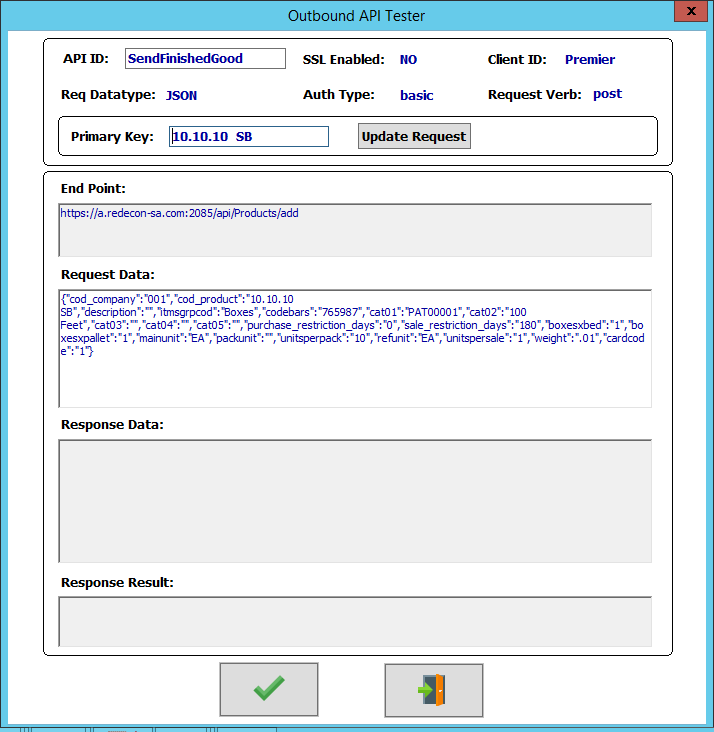


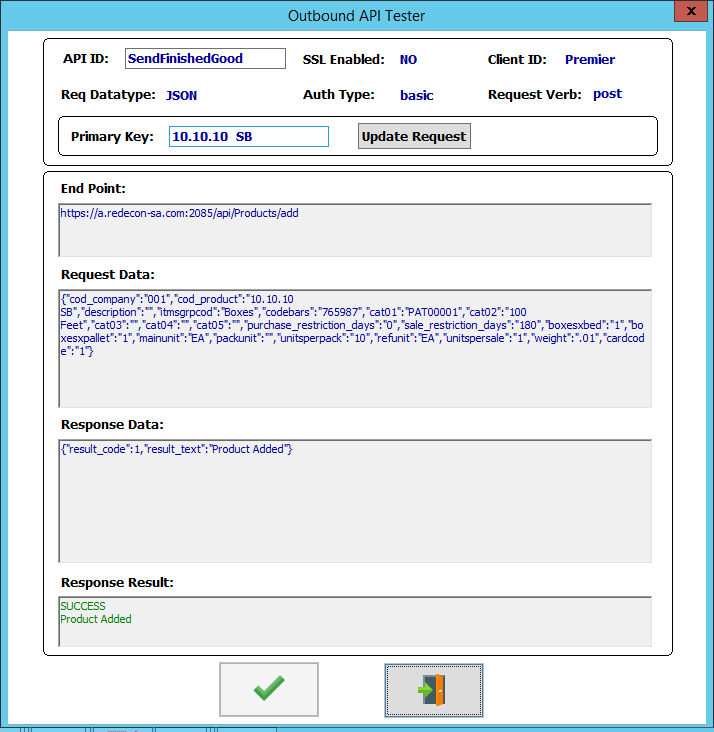
**“SendFinishedGood”:**

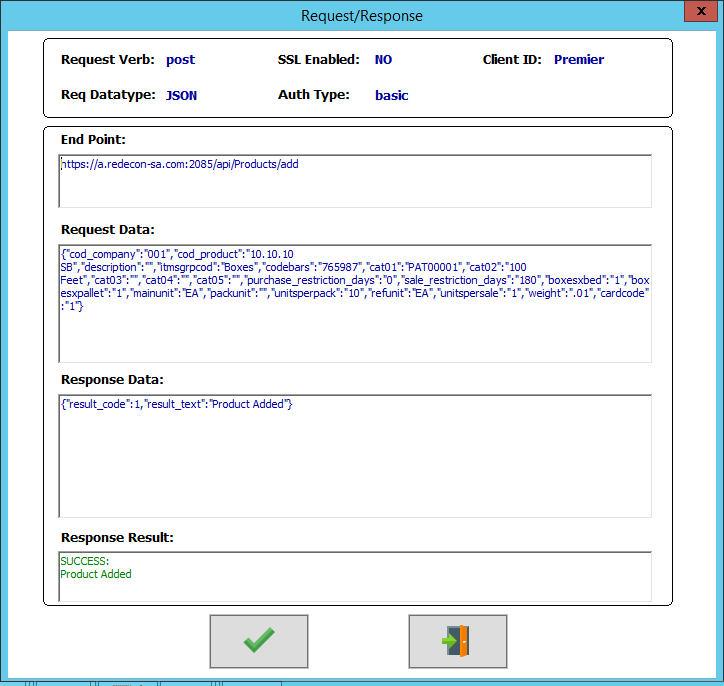
1. Enter “SendFinishedGood” in the “API ID” field or select from “F1” lookup on “API ID” field. “F1” lookup on the “Primary Key” will now launch the “Finished Good Item” lookup screen.



1. Selecting a row from the “Finished Good Item” lookup screen will populate the request data for the selected record. User can also manually enter a valid finished good item number in the “Primary Key” and click the “Update Request” button to populate the request data for the finished good record. User can further modify the request data manually before calling the endpoint.

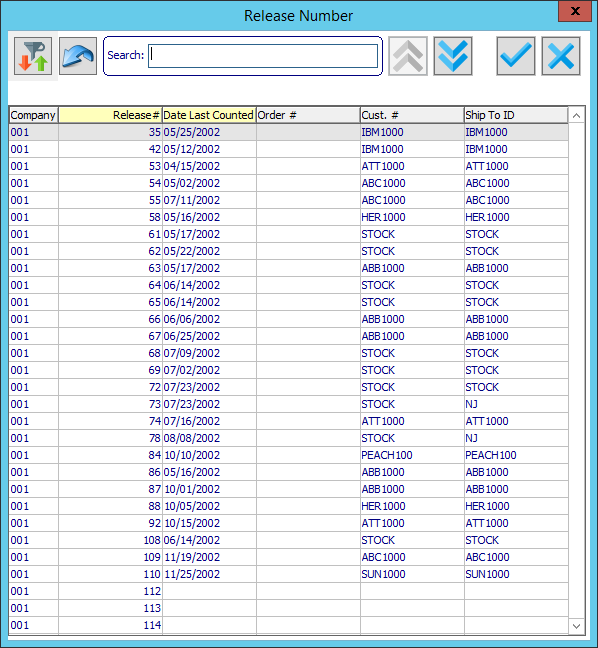




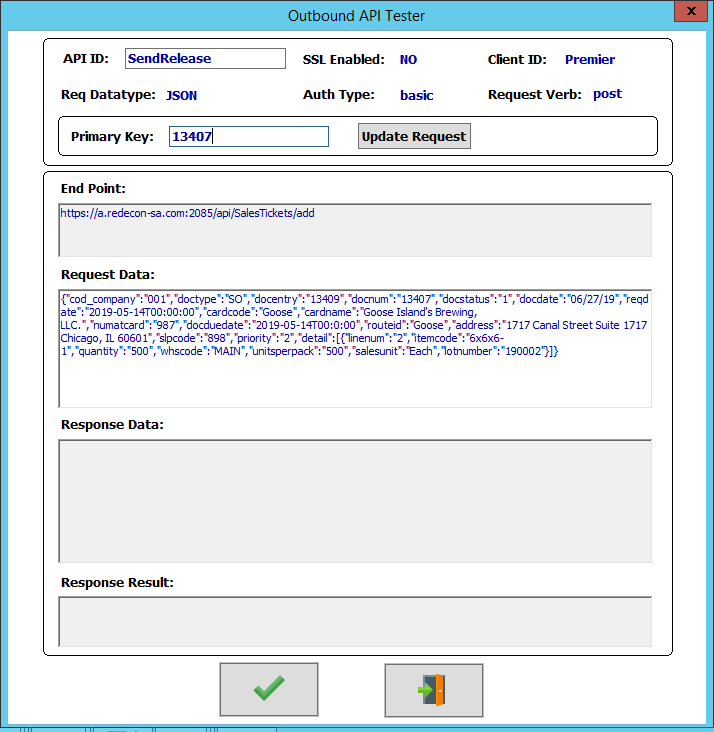


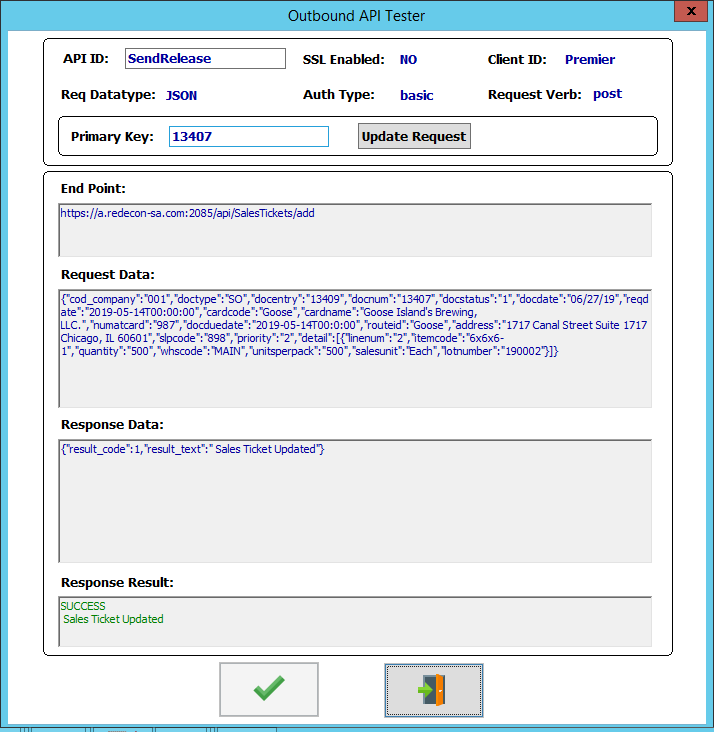
**“SendRelease”:**

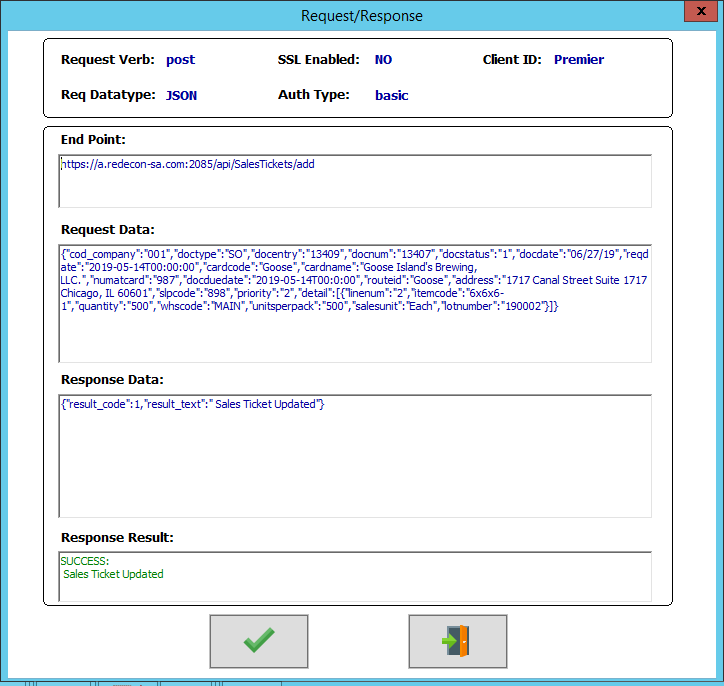
1. Enter “SendRelease” in the “API ID” field or select from “F1” lookup on “API ID” field. “F1” lookup on the “Primary Key” will now launch the “Release” lookup screen.



1. Selecting a row from the “Release” lookup screen will populate the request data for the selected record. User can also manually enter a valid release number in the “Primary Key” and click the “Update Request” button to populate the request data for the release record. User can further modify the request data manually before calling the endpoint.

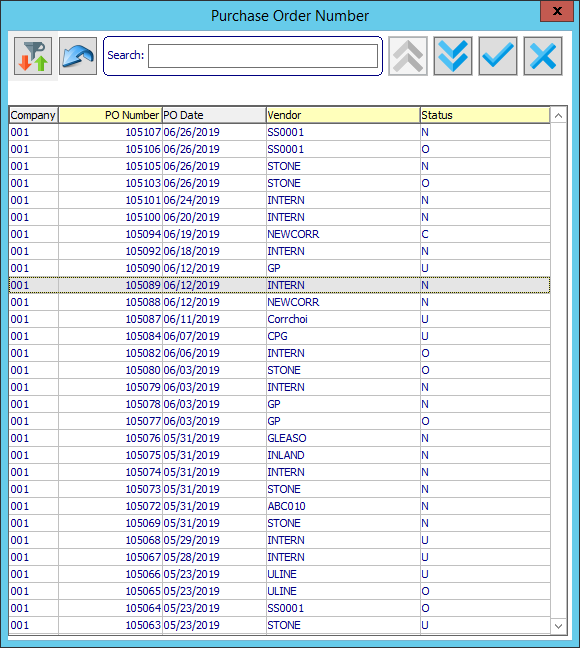






**“SendPurchaseOrder”:**

1. Enter “SendPurchaseOrder” in the “API ID” field or select from “F1” lookup on “API ID” field. “F1” lookup on the “Primary Key” will now launch the “Purchase Order” lookup screen.



1. Selecting a row from the “Purchase Order” lookup screen will populate the request data for the selected record. User can also manually enter a valid purchase order number in the “Primary Key” and click the “Update Request” button to populate the request data for the release record. User can further modify the request data manually before calling the endpoint.

