**API Inbound cxmlorder Testing**

**(EXTERNAL USE)**

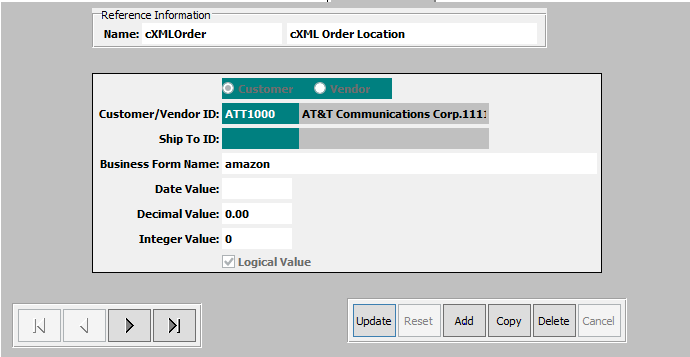
Last update date: **26-Aug-2019 |** Version: **1.0**

1. **Set up needed in ASI**

* Go to “System Control Parameters” (N-K-1) screen
* Search for name “cXMLOrder”. Here, we are assuming that “cXMLOrder” is already created
* Create new form (sys-ctrl-shipto) by following below steps

1. Select radio button “Customer”
2. Enter “Customer/Vendor ID” field for which we tag the customer name
3. Enter customer name in “Business Form Name” field. Customer name should be same as “Identity” tag value in request XML provided by customer
4. Check “Logical Value” toggle box

Find the below sample form created for customer “amazon” linked to Customer ID “ATT1000”



**2. Testing by using Postman**

* Launch Postman
* Select request type as **POST**
* Enter “http://<IPAddress>:<PORT>/api/cxmlorder” – this is the endpoint URL
* Go to “Authorization” tab and select Type as **Basic**
* Enter username and password
* Go to “Body” tab
* Select “raw” and “application/xml” as data type as shown below
* Enter the request cXML

1. **Test by executing below test cases**

**Case 1: Validation of company code and customer number**

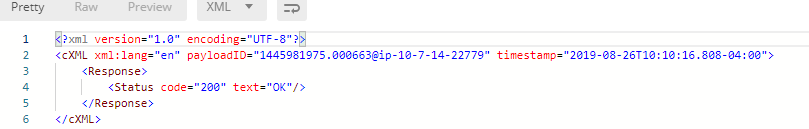
* When identity and addressID are valid then fetch company code and customer number from sys-ctrl-shipto table

Modify “cXML request data” provided by customer as below.

* Modify “orderID” attribute with a unique value and “orderDate” with current date
* Modify “Identity” and “addressID” with the valid company name and addressID present in sys-ctrl-shipto
* Sample test data:

Identity(amazon) and addressID(9523949)

* Now click on **Send** button to call API “cxmlorder”
* If API “cxmlorder” call is successful, then below XML is returned as the response



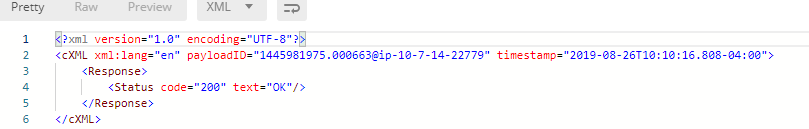
* Verify Order creation in application UI
* Go to screen OU1 (Order Maintenance)
* Enter orderID in “cust PO#” and click on “Go”
* Now, results related to that particular “orderID” will be shown as below
* When identity is invalid and addressID is valid then fetch company code and customer number from shipto table for the given addressID

Modify “cXML request data” provided by customer as below.

* Modify “orderID” attribute with a unique value and “orderDate” with current date
* Modify “Identity” and “addressID” with the Invalid company name and addressID Sample test data:

Identity(amazon) and addressID(HEW1001)

* Now click on **Send** button to call API “cxmlorder”
* If API “cxmlorder” call is successful, then below XML is returned as the response



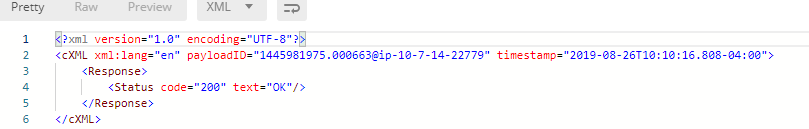
* Verify Order creation in application UI
* Go to screen OU1 (Order Maintenance)
* Enter orderID in “cust PO#” and click on “Go”
* Now, results related to that particular “orderID” will be shown as below
* When identity and addressID are Invalid/addressID is not available in shipto table then fetch company code and customer number from cust table

Modify “cXML request data” provided by customer as below.

* Modify “orderID” attribute with a unique value and “orderDate” with current date
* Modify “Identity” and “addressID” with the Invalid company name and Invalid addressID Sample test data:

Identity(amaz) and addressID(9523)

* Now click on **Send** button to call API “cxmlorder”
* If API “cxmlorder” call is successful, then below XML is returned as the response



* Verify Order creation in application UI
* Go to screen OU1 (Order Maintenance)
* Enter orderID in “cust PO#” and click on “Go”
* Now, results related to that particular “orderID” will be shown as below

**Case 2: Fetching supplierID and manufacturingID based on supplierID validation**

* When supplierID is available in cust-part table then fetch supplierID and manufacturingID from cust-part table

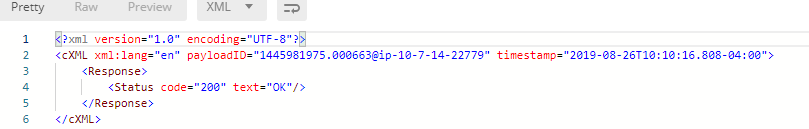
Modify “cXML request data” provided by customer as below.

* Modify “orderID” attribute with a unique value and “orderDate” with current date
* Modify “supplierID” with the valid supplierID present in cust-part table

Sample test data:

supplierID(1234)

* Now click on **Send** button to call API “cxmlorder”
* If API “cxmlorder” call is successful, then below XML is returned as the response



* Verify Order creation in application UI
* Go to screen OU1 (Order Maintenance)
* Enter orderID in “cust PO#” and click on “Go”
* Now, results related to that particular “orderID” will be shown
* When supplierID is available in itemfg table but not in cust-part table then fetch supplierID and manufacturingID from itemfg table

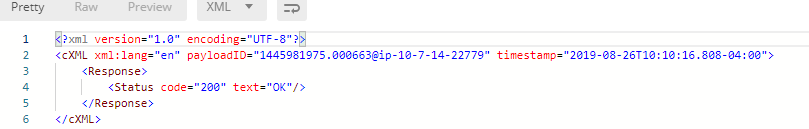
Modify “cXML request data” provided by customer as below.

* Modify “orderID” attribute with a unique value and “orderDate” with current date
* Modify “supplierID” with the valid supplierID present in cust-part table

Sample test data:

supplierID(1234)

* Now click on **Send** button to call API “cxmlorder”
* If API “cxmlorder” call is successful, then below XML is returned as the response



* Verify Order creation in application UI
* Go to screen OU1 (Order Maintenance)
* Enter orderID in “cust PO#” and click on “Go”
* Now, results related to that particular “orderID” will be shown as below
* When supplierID is invalid/not available in itemfg table then fetch supplierID and manufacturingID from the first record of itemfg table

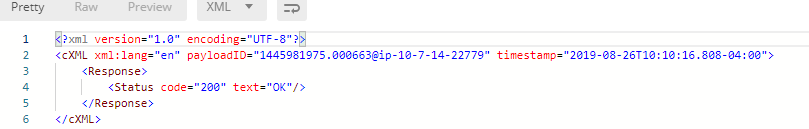
Modify “cXML request data” provided by customer as below.

* Modify “orderID” attribute with a unique value and “orderDate” with current date
* Modify “supplierID” with the valid supplierID present in cust-part table

Sample test data:

supplierID(1194)

* Now click on **Send** button to call API “cxmlorder”
* If API “cxmlorder” call is successful, then below XML is returned as the response



* Verify Order creation in application UI
* Go to screen OU1 (Order Maintenance)
* Enter orderID in “cust PO#” and click on “Go”
* Now, results related to that particular “orderID” will be shown

1. **How to call the API using a program / code**

|  |  |
| --- | --- |
| **Endpoint URL** | http://<IPAddress>:<PORT>/api/cxmlorder |
| **Verb** | POST |
| **Authentication Type** | Basic |
| **Authentication Credentials** | **Username:** <username> |
|  | **Password:** <password> |
| **Content-Type** | application/xml |
| **Request Payload** | cXML-based XML data |