SHOP CONNECTOR API SERVICES

Technical Design Document

# DOCUMENT REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Release Date** | **Revised by** | **Comments/ Indicate Sections Revised** |
| DRAFT | 09/09/2019 | Connector Development Team | Draft version of the Technical Design Document |

Table of Contents

[DOCUMENT REVISION HISTORY 2](#_gjdgxs)

[A.](#_30j0zll) INTRODUCTION 4

1. [**Purpose** 4](#_1fob9te)
2. [**Background of Commerce cloud OCCAPI** 4](#_3znysh7)
3. [**Commerce Cloud API Service** 5](#_tyjcwt)
4. [**Overview**](#_3dy6vkm) 5

[B.](#_1t3h5sf) MAPPING – BUSINESS FUNCTIONS CALLS to TECHNICAL OPERATIONS 5

1. [**List of operations supported by the Shop API Connector:**](#_4d34og8) 5

[C.](#_3j2qqm3) Configurations for Shop API Connector 22

[D.](#_26in1rg) Use cases 26

1. Use case withSalesForce 27

[APPENDIX A 16](#_35nkun2)

[Configure Shop API Connector In anypoint studio](#_1ksv4uv) 36

1. [**Authentication**](#_44sinio) 36
2. [**Configure in Anypoint Studio**](#_2jxsxqh) 37
3. [**About Connector Namespace and Schema**](#_z337ya) 38

# INTRODUCTION

## **Purpose**

This specification document outlines the instructions for Participating Organizations (PO) to access the API service offered in the Commerce Cloud OCCAPI systems. This document also serves as a reference for the service request parameters, service responses and other information pertinent to the dissemination of Lending information through the API. While there are similar parameters and commands to add, update, and delete services from a Commerce Cloud API endpoints.

## **Background of Commerce Cloud Occapi**

The Open Commerce API (OCAPI) enables you to access resources using HTTP requests and HTTP responses. Each resource is addressed by its unique URL, which includes the API version. Data is transported using request or header parameters, or within the request body as a JSON document with a defined structure. OCAPI is divided into different APIs: the Shop API allows a client to interact with the system as a shop customer or as an agent shopping on behalf of a customer, the Data API gives Create, Read, Update and Delete access to system resources, and the Meta API is used to retrieve a formal description of the Open Commerce API, including custom attributes.

## **Commerce cloud** **API Service**

COMMERCE CLOUD SHOP APIs provide access to various services for the following workflows:

* + **Products:**
  + **Categories**
  + **Customers**
  + **Sessions**
  + **Baskets**
  + **Product Search**
  + **Orders**
  + **Order Search**
  + **Promotions**
  + **Search Suggestion**
  + **Site**
  + **Stores**
  + **Folders**
  + **Content**
  + **Content Search**

## **Overview**

A Web Service is a method of communication between two electronic devices over a network. Web services allow organizations to communicate data without close knowledge of each other's IT systems. These web services are often called, API or Application Programming Interface.

*A high-level diagram of the architecture is shown in Figure 1.*

|  |
| --- |
|  |

Figure 1: Mule Commerce Cloud Connector Integration Architecture

# MAPPING – BUSINESS FUNCTIONS CALLS to TECHNICAL OPERATIONS

These operations are available for consumption in a manner which is suitable to the business needs of the Mule application developer. The first column represents the business functions. The next column denotes the corresponding operations and it’s descriptions.

## **List of operations supported by the Commerce Cloud Connector SHOP API:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Business  Function | Operations | Description | Inbound | Outbound |
| PRODUCT | | | | |
| Get Products By IDs | GET Request | Allows to access multiple products by a single request. This convenience resource should be used instead of making a separate requests. | Please refer an API specification for details on the URI parameters and query parameters.  Ids are required parameters in this operation. | Return a JSON response with an array of Product details. |
| Get Products By ID | GET Request | To access single products resource, requires you to specify an Id (typically a SKU) for a product. | Please refer an API specification for details on URI parameters and query parameters. | Returns the Response JSON with product details |
| Get Products By IDAvailability | GET Request | Access product availability information of products that are online and assigned to site catalog. | Please refer an API specification for details on URI parameters and query parameters. | Return the Response JSON with details of product. |
| Get Products By IDBundled Products | GET Request | Access bundled product information of products that are online and assigned to site catalog. | Please refer an API specification for details on URI parameters and query parameters. | Return the Response JSON with details of product. |
| Get Products By IDImages | GET Request | Access product image information of products that are online and assigned to site catalog. Filter the result by view type and variation values. | Required field is ID of the product to retrieve | Return the Response JSON with details of product. |
| Get Products By IDLinks | GET Request | Access product link information of products that are online and assigned to site catalog. Filter the result by link type and link direction. | Required field is ID of the product to retrieve | Return the Response JSON with details of product. |
| Get Products By IDOptions | GET Request | Access product option information of products that are online and assigned to site catalog. | Required field is ID of the product to retrieve | Return the Response JSON with details of product. |
| Get Products By IDPrices | GET Request | Access product price information of products that are online and assigned to site catalog. | Required field is ID of the product to retrieve | Return the Response JSON with details of product. |
| Get Products By IDPromotions | GET Request | Access product promotion information of products that are online and assigned to site catalog. | Required field is ID of the product to retrieve | Return the Response JSON with details of product. |
| Get Products By IDRecommendations | GET Request | Access product recommendation information of products that are online and assigned to site catalog. | Required field is ID of the product to retrieve | Return the Response JSON with details of product. |
| Get Products By IDSet Products | GET Request | Access product set information of products that are online and assigned to site catalog. | Required field is ID of the product to retrieve | Return the Response JSON with details of product. |
| Get Products By IDShipping Methods | GET Request | Retrieves the applicable shipping methods for a certain product. | Required field is ID of the product to retrieve | Return the Response JSON with details of product. |
| Get Products By IDVariations | GET Request | Access product variation information of products that are online and assigned to site catalog. | Required field is ID of the product to retrieve | Return the Response JSON with details of product. |
| CATEGORIES | | | | |
| Get Categories By IDs | GET Request | The server returns multiple categories (a result object of category documents). You can use this service for obtaining multiple categories in a single request. | Please refer an API specification for details on URI parameters and query parameters. You can pass a comma separated list of category ids. | Return the Response JSON with details of category. |
| Get Categories By ID | GET Request | The server returns a category identified by its id; by default, the server also returns the first level of subcategories, but you can specify another level by setting the levels parameter. | Please refer an API specification for details on URI parameters and query parameters. Category id is a required parameter | Return the Response JSON with details of category. |
| CUSTOMERS | | | | |
| Post Customers | POST Request | Registers a customer. | The mandatory data are the credentials and profile last name and email. When using OAuth the password in the request must not be set, otherwise an Invalid Password Exception will be thrown. | Return the Response JSON with details of created Customer. |
| Delete Customers Auth | DELETE Request | Invalidates the JWT provided in the header. | Pass the Authorization as a header parameter. For more details refer the API specification | Response id Success status code. |
| Post Customers Auth | POST Request | Obtains a new JWT (JSON Web Token) for a guest or registered customer. Tokens are returned as HTTP Authorization: Bearer response header entry. | Please refer the API document for the details of Request body and parameters | Return the Response JSON with details of Customer. |
| Post Customers password Reset | POST Request | Before POST hook is called first. After that the validation of the customer information provided in the password reset document is performed. Then a password reset token is generated and together with the resolved customer is passed to a after POST hook. | Please refer an API specification for details on body parameters. | Return the status code |
| Get Customers By ID | GET Request | Gets a customer | Please refer the API specification for the URI parameter and query parameters | Returns the response JSON with Customer data. |
| Patch Customers By ID | PATCH Request | Updates a customer | Please refer the API specification for the request body parameters | Returns the response JSON with Customer data. |
| Get Customers By IDAddresses | GET Request | Returns a sorted pageable list of all customer addresses in the address book. The default page size is 10 customer addresses. The addresses are sorted so that the preferred address is always sorted first. The remaining addresses are sorted alphabetically by ID. | Please refer the API specification for the URI parameters and query parameters | Returns the response JSON with Customer Address data. |
| Post Customers By IDAddresses | POST Request | Creates a new address with the given name for the given customer | Please refer the API specification for request body parameters. | Returns the response JSON with Customer Address data. |
| Get Customers By IDAddresses By ID | GET Request | Retrieves a customer's address by address name. | Please refer the API specification for the URI parameters and query parameters | Returns the response JSON with Customer Address data. |
| Delete Customers By IDAddresses By ID | DELETE Request | Deletes a customer's address by address name. | Please refer the API specification for the URI parameters and query parameters | Returns the status code |
| Patch Customers By IDaddresses By ID | PATCH Request | Updates a customer's address by address name. | Please refer the API specification for the request body parameter | Returns the response JSON with Customer Address data. |
| Post Customers By IDAuth | POST Request | Obtains a new agent on behalf token for a registered customer. Token is returned as a HTTP Authorization:Bearer response header entry. | Please refer the API specification for the URI parameters and query parameters | Returns the response JSON with Customer data. |
| Get Customers By IDBaskets | GET Request | Gets the baskets of a customer | Customer id is the required parameter | Returns the response JSON with Baskets result. |
| Get Customers By IDOrders | GET Request | Returns a pageable list of all customer's orders. The default page size is 10. | Please refer the API specification for URI and Query parameters | Returns the response JSON with Customer order result. |
| Put Customers By IDpassword | PUT Request | Updates the customer’s password | Refer API specification for passing request body parameter | Returns status code |
| Post Customers By IDPassword Reset | POST Request | Starts a password reset process. A password reset token is generated and passed together with the customer resolved by the id provided as path parameter to a afterPOST hook. | Refer API specification for URI and Query parameters. | Returns status code. |
| Get Customers By IDPayment Instruments | GET Request | Gets customer payment instruments for a customer. | Refer API specification for URI and Query parameters. | Returns the response JSON with Customer payments instrument result. |
| Post Customers By IDPayment Instruments | POST Request | Adds a payment instrument to a customer information. | Refer API specification for Request body parameters. | Returns the response JSON with Customer payments instrument. |
| Get Customers By IDPayment Instruments By ID | GET Request | Retrieves a customer's payment instrument by its id. | Refer API specification for URI and Query parameters. | Returns the response JSON with Customer payments instrument result. |
| Delete Customers By IDPayment Instruments By ID | DELETE Request | Deletes a customer’s payment instrument. | Refer API specification for URI and Query parameters. | Returns the status code. |
| Get Customers By IDProduct Lists | GET Request | Returns all customer product lists. | Refer API specification for URI and Query parameters. | Returns the response JSON with Customer product list result. |
| Post Customers By IDProduct Lists | POST Request | Creates a customer product list | Refer API specification for URI and Request body parameters. | Returns the status code. |
| Get Customers By IDProduct Lists By ID | GET Request | Returns customer product list of the given customer | Refer API specification for URI and Query parameters. | Returns the response JSON with Customer product list details. |
| Patch Customers By IDproduct Lists By ID | PATCH Request | Changes a product list. Changeable properties are the name, description and if the list is public. | Refer API specification for URI and request body parameters. | Returns the response JSON with Customer product list details. |
| Delete Customers By IDProduct Lists By ID | DELETE Request | Deletes a customer product list. | Refer API specification for URI and Query parameters. | Returns the status code. |
| Get Customers By IDProduct Lists By IDItems | GET Request | Returns a pageable list of all items of a customer's product list. The default page size is 10. | Refer API specification for URI and Query parameters. | Returns a response JSON with Customer product list item result. |
| Post Customers By IDProduct Lists By IDItems | POST Request | Adds an item to the customer's product list. | Refer API specification for URI and request body parameters. | Returns a response JSON with Customer product list item result. |
| Get Customers By IDProduct Lists By IDItems By ID | GET Request | Returns an item of a customer product list | Refer API specification for URI and Query parameters. | Returns a response JSON with Customer product list item result. |
| Delete Customers By IDProduct Lists By IDItems By ID | DELETE Request | Removes an item from a customer product list. | Refer API specification for URI and query parameters. | Returns the status code. |
| Patch Customers By IDproduct Lists By IDitems By ID | PATCH Request | Updates an item of a customer's product list | Refer API specification for URI and request body parameters. | Returns a response JSON with Customer product list item result. |
| Get Customers By IDProduit Lists By IDItems By IDPurchases | GET Request | Returns a list of all purchases of an item from a customer’s product list. | Refer API specification for URI and query parameters. | Returns a response JSON with Customer product list item purchase result. |
| Post Customers By IDProduct Lists By IDItems By IDPurchases | POST Request | Adds a purchase to an item in the customer's product list. | Refer API specification for URI and request body parameters. | Returns a response JSON with Customer product list item purchase result. |
| Get Customers By IDProduct Lists By IDItems By IDPurchases By ID | GET Request | Returns a purchase of an item from a customer’s product list. | Refer API specification for URI and query parameters. | Returns a response JSON with Customer product list item purchase result. |
| Patch Customers By IDproduct Lists By IDitems By IDpurchases By ID | PATCH Request | Updates a purchase of an item from a customer's product list. | Refer API specification for URI and request body parameters. | Returns a response JSON with Customer product list item purchase result. |
| ORDERS | | | | |
| Post Orders | POST Request | Submits an order based on a prepared basket. | Refer API specification for URI and request body parameters. | Returns a response JSON with Order data. |
| Put Orders By ID | PUT Request | Submits an order with a given order number, based on a prepared basket. | Refer API specification for URI and request body parameters. | Returns a response JSON with Order data. |
| Get Orders By ID | GET Request | Gets information for an order. | Order number is the required URI parameter | Returns a response JSON with Order data |
| Patch Orders By ID | PATCH Request | Update an order.  Considered fields for update are status and custom properties. | Refer API specification for URI and request body parameters. | Returns a response JSON with Order data. |
| Post Orders By IDNotes | POST Request | Adds a note to an existing order | Refer API specification for URI and request body parameters. | Returns a response JSON with Order data. |
| Get Orders By IDNotes | GET Request | Retrieves notes for an order. | Order number is the required URI parameter | Returns a response JSON with Notes Result. |
| Delete Orders By IDNotes By ID | DELETE Request | Removes an order note | Note id and Order no are the required URI parameters. | Returns a response JSON with Order data. |
| Post Orders By IDPayment Instruments | POST Request | Adds a payment instrument to an order. It is possible either to supply the full payment information or only a customer payment instrument id and amount. | Refer API specification for URI and request body parameters. | Returns a response JSON with Order data. |
| Patch Orders By IDpayment Instruments By ID | PATCH Request | Updates a payment instrument of an order and passes the order and updated payment instrument to the correct payment authorize Credit card or authorize hook. | Refer API specification for URI and request body parameters. | Returns a response JSON with Order data. |
| Delete Orders By IDPayment Instruments By ID | DELETE Request | Removes a payment instrument of an order | Refer API specification for URI and query parameters. | Returns a response JSON with Order data. |
| Get Orders By IDPayment Methods | GET Request | Gets the applicable payment methods for an existing order considering the open payment amount only | Refer API specification for URI and query parameters. | Returns a response JSON with Payment method result. |
| PRODUCT SEARCH | | | | |
| Get Product Search | GET Request | Provides keyword and refinement search functionality for products. Only returns the product id, link and name in the product search hit. | Refer API specification for URI and query parameters. | Returns a response JSON with Product search result. |
| Get Product Search Availability | GET Request | Provides keyword and refinement search functionality for products. Only returns the product id, link, name and availability information in the product search hit. | Refer API specification for URI and query parameters. | Returns a response JSON with Product search result. |
| Get Product Search Images | GET Request | Provides keyword and refinement search functionality for products. Only returns the product id, link, name and image information in the product search hit. | Refer API specification for URI and query parameters. | Returns a response JSON with Product search result. |
| Get Product Search Prices | GET Request | Provides keyword and refinement search functionality for products. Only returns the product id, link, name and image information in the product search hit. | Refer API specification for URI and query parameters. | Returns a response JSON with Product search result. |
| Get Product Search Represented Products | GET Request | Provides keyword and refinement search functionality for products. Only returns the product id, link, name and image information in the product search hit. | Refer API specification for URI and query parameters. | Returns a response JSON with Product search result. |
| Get Product Search Variations | GET Request | Provides keyword and refinement search functionality for products. Only returns the product id, link, name and image information in the product search hit. | Refer API specification for URI and query parameters. | Returns a response JSON with Product search result. |
| PROMOTIONS | | | | |
| Get Promotions | GET Request | Handles get promotion by filter criteria Returns an array of enabled promotions matching specified filter criteria. | Refer API specification for URI and query parameters.  You must provide a campaign\_id parameter, and you can optionally specify a date range by providing start\_date and end\_date parameters. Both parameters are required to specify a date range: omitting one causes the server to return a MissingParameterException fault. Each request returns only enabled promotions; the server does not consider promotion qualifiers or schedules. | Returns a response JSON with Promotions result. |
| Get Promotions By IDs | GET Request | Returns an array of enabled promotions for a list of specified ids | Refer API specification for URI and query parameters.  In the request URL, you can specify up to 50 ids. You must enclose the list of ids in parentheses. | Returns a response JSON with Promotions result. |
| Get Promotions By ID | GET Request | Returns an enabled promotion using a specified id. Each request returns a response only for an enabled promotion; the server does not consider promotion qualifiers or schedules. | Refer API specification for URI and query parameters. | Returns a response JSON with Promotions data. |
| SITE | | | | |
| Get Site | GET Request | Access site information, like site status and site content URLs. | No parameters are required | Response JSON with Site data. |
| ORDER SEARCH | | | | |
| Post Order Search | POST Request | Searches for orders. | Refer API specification for URI, request body and query parameters.  The query attribute specifies a complex query that can be used to narrow down the search.  Note that search fields are mandatory now and no default ones are supported. | Returns a response JSON with Order search result data. |
| SESSIONS | | | | |
| Post Sessions | POST Request | Exchanges a JWT token into a new session. If the given token is valid, creates a new session, which is associated with the authenticated or anonymous customer. All *Set-Cookie* headers for handling the session are applied on the response. | No parameters are required for this operation | Returns the status code |
| SEARCH SUGGESTION | | | | |
| Search Suggestion | GET Request | Provides keyword search functionality for products, categories, content, brands and custom suggestions. Returns suggested products, suggested categories, suggested content, suggested brands and custom suggestions for the given search phrase. | Refer API specification for URI and query parameters. | Returns a response JSON with Suggestion result data. |
| STORES | | | | |
| Get Stores | GET Request | This resource retrieves a list of stores, for the given site, that are within a configured distance of a location on the earth. | Refer API specification for URI and query parameters. | Returns a response JSON with Store result data. |
| Get Stores By IDs | GET Request | To access multiple stores | Refer API specification for URI and query parameters. | Returns a response JSON with Store result data. |
| Get Stores By ID | GET Request | To access a store | Refer API specification for URI and query parameters. | Returns a response JSON with Store data. |
| FOLDERS | | | | |
| Get Folders By IDs | GET Request | To access one or more content folder | Refer API specification for URI and query parameters. | Returns a response JSON with Content folder result data. |
| Get Folders By ID | GET Request | To access a content folder | Refer API specification for URI and query parameters. | Returns a response JSON with Content folder data. |
| CONTENT | | | | |
| Get Content By IDs | GET Request | To access multiple content asset | Refer API specification for URI and query parameters. | Returns a response JSON with Content result data. |
| Get Content By ID | GET Request | To access content asset | Refer API specification for URI and query parameters. | Returns a response JSON with Content data. |
| CONTENT SEARCH | | | | |
| Get Content Search | GET Request | Provides keyword and refinement search functionality for content assets. The search result contains only content that is online and assigned to a folder. | Refer API specification for URI and query parameters. | Returns a response JSON with Content search result. |
| BASKETS | | | | |
| Post Baskets | POST Request | Creates a new basket. The created basket is initialized with default values. Data provided in the body document will be populated into the created basket. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with Basket data. |
| Post Baskets Reference | POST Request | Creates a new basket based on a basket reference. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with Basket data. |
| Get Baskets By ID | GET Request | Gets a basket | Refer API specification for URI and query parameters. | Returns a response JSON with Basket data. |
| Patch Baskets By ID | PATCH Request | Updates a basket. Only the currency of the basket, source code, and the custom properties of the basket and of the shipping items will be considered. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with Basket data. |
| Delete Baskets By ID | DELETE Request | Removes a basket | Refer API specification for URI and query parameters. | Returns the status code |
| Put Baskets By IDagent | PUT Request | Marks a basket as an agent basket | Refer API specification for URI and query parameters. | Returns a response JSON with Basket data. |
| Get Baskets By IDApproaching Discounts | GET Request | Gets the approaching discounts of a basket | Refer API specification for URI and query parameters. | Returns a response JSON with Approaching discount result data. |
| Put Baskets By IDbilling Address | PUT Request | Sets the billing address of a basket. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with Approaching discount result data. |
| Post Baskets By IDCoupons | POST Request | Adds a coupon to an existing basket | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with Coupon Item. |
| Delete Baskets By IDCoupons By ID | DELETE Request | Removes a coupon from the basket | Refer API specification for URI and query parameters. | Returns a response JSON with basket data. |
| Put Baskets By IDcustomer | PUT Request | Sets customer information for an existing basket | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Post Baskets By IDGift Certificate Items | POST Request | Adds a gift certificate item to an existing basket. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Delete Baskets By IDGift Certificate Items By ID | DELETE Request | Deletes a gift certificate item from an existing basket. | Refer API specification for URI and query parameters. | Returns a response JSON with basket data. |
| Patch Baskets By IDGift Certificate Items By ID | PATCH Request | Updated a gift certificate of an existing basket | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Post Baskets By IDItems | POST Request | Adds new items to a basket. The added items are associated with the specified shipment. If no shipment id is specified, the added items are associated with the default shipment. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Patch Baskets By IDitems By ID | PATCH Request | Updates an item in a basket. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Delete Baskets By IDItems By ID | DELETE Request | Removes a product item from the basket | Refer API specification for URI parameters. | Returns a response JSON with basket data. |
| Post Baskets By IDNotes | POST Request | Adds a note to an existing basket | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Get Baskets By IDNotes | GET Request | Retrieves notes for a basket | Refer API specification for URI parameters. | Returns a response JSON with Notes result data. |
| Delete Baskets By IDNotes By ID | DELETE Request | Removes a basket note. | Refer API specification for URI parameters. | Returns a response JSON with Notes result data. |
| Post Baskets By IDPayment Instruments | POST Request | Adds a payment instrument to a basket.  Payment instruments are usually authorized after order creation, for example in a custom hook. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Patch Baskets By IDpayment Instruments By ID | PATCH Request | Updates a payment instrument of a basket.  Payment instruments are usually authorized after order creation, for example in a custom hook. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Delete Baskets By IDPayment Instruments By ID | DELETE Request | Removes a payment instrument of a basket | Refer API specification for URI, query parameters. | Returns a response JSON with basket data. |
| Get Baskets By IDPayment Methods | GET Request | Gets applicable payment methods for an existing basket considering the open payment amount only. | Refer API specification for URI, query parameters. | Returns a response JSON with Payment Method Result data. |
| Post Baskets By IDPrice Adjustments | POST Request | Adds a custom manual price adjustment to the basket | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Delete Baskets By IDPrice Adjustments By ID | DELETE Request | Removes a custom manual price adjustment from the basket | Refer API specification for URI, query parameters. | Returns a response JSON with basket data. |
| Patch Baskets By IDprice Adjustments By ID | PATCH Request | Updates a custom manual price adjustment on the basket. Only the following PATH values are considered for the update: discount, item\_text, reason\_code and custom properties; all other attributes are ignored. The discount type of a price adjustment cannot be updated and therefore, the value of the existing type must be passed. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Post Baskets By IDShipments | POST Request | Creates a new shipment for a basket.  The created shipment is initialized with values provided in the body document and can be updated with further data API calls. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Patch Baskets By IDshipments By ID | PATCH Request | Updates a shipment for a basket.  The shipment is initialized with values provided in the body document and can be updated with further data API calls | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Delete Baskets By IDShipments By ID | DELETE Request | Removes a specified shipment and all associated product, gift certificate, shipping and price adjustment line items from a basket. It is not permissible to remove the default shipment. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Put Baskets By IDshipments By IDshipping Address | PUT Request | Sets a shipping address of a specific shipment of a basket. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Put Baskets By IDshipments By IDshipping Method | PUT Request | Sets a shipping method to a specific shipment of a basket. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |
| Get Baskets By IDShipments By IDShipping Methods | GET Request | Gets the applicable shipping methods for a certain shipment of a basket. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with shipping method result. |
| Put Baskets By IDstorefront | PUT Request | Marks a basket as storefront basket. | Refer API specification for URI, Request body and query parameters. | Returns a response JSON with basket data. |

# CONFIGURATION FOR SHOP CONNECTOR

To use Commerce Cloud Connector, you will also need an account on Commerce Cloud. You will require the credentials for creating your access token and refresh token, which will be required for the connector.

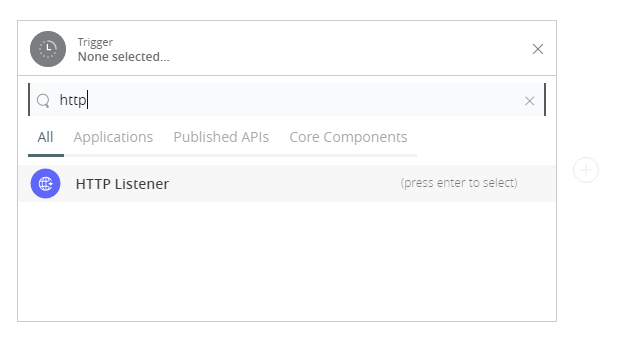
## **Authentication**

The API uses JWT or OAuth2.0 where a token needs to be generated for communicating with server. This token can be of following three types:

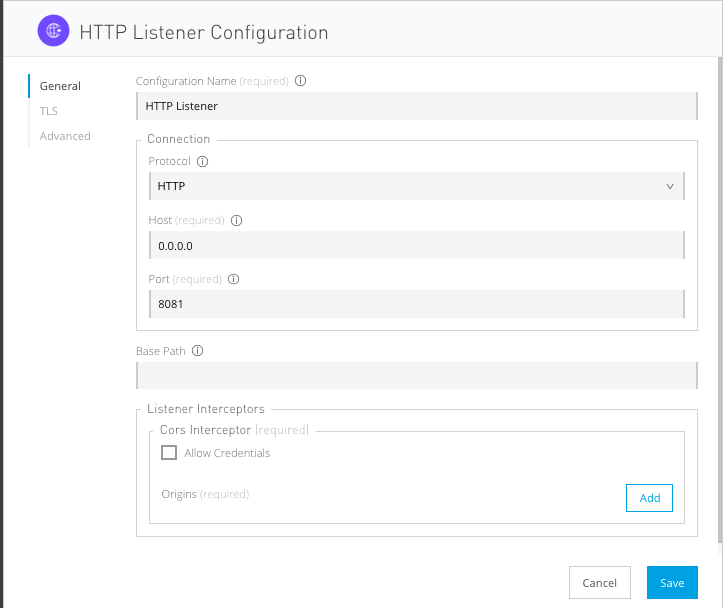
1. **Registered customer:** 
   1. base URI
   2. username: <username>
   3. password: <password>
   4. clientId: <client\_id>
2. **Guest customer:** 
   1. base URI
   2. clientId: <client\_id>
3. **Business Manager:**
   1. base URI
   2. clientid: <client\_id>
   3. clientSecret: <client\_secret>
   4. token URI
   5. username: <username>
   6. password: <password>

# TO CONNECT IN DESIGN CENTER

1. In Design Centre, click Create and choose Mule Application.
2. Click a trigger such as an HTTP Listener or the Scheduler trigger.

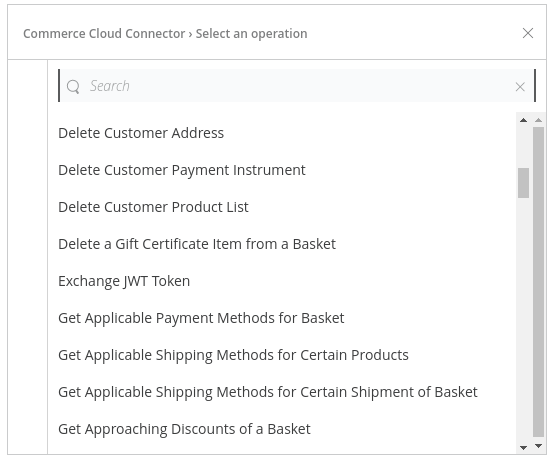


1. To create an HTTP global element for the connector, set these fields:

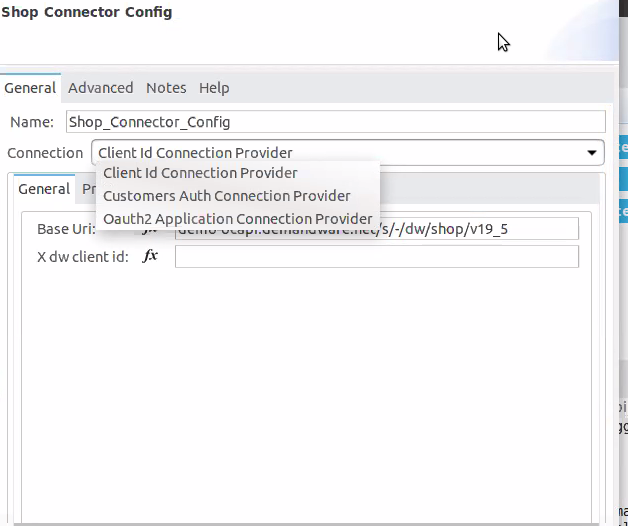


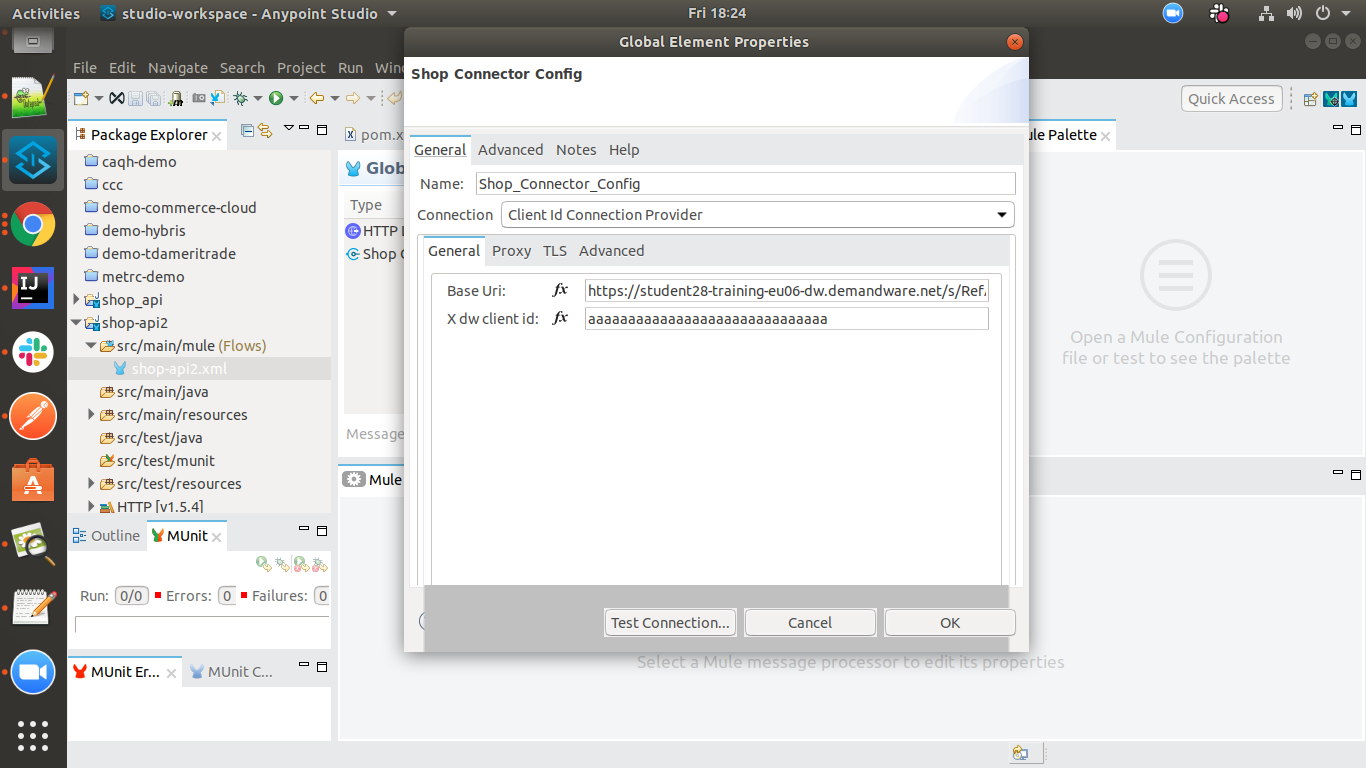
|  |  |
| --- | --- |
| **Field** | **Description** |
| **Protocol** | Protocol selected for the HTTP connector, it can be HTTP or HTTPS (secure). |
| **Host** | IP address where your Mule application listens for requests. |
| **Port** | Port address where your Mule application listens for requests. |
| **Base Path** | Path where your Mule application listens for requests. |

1. Select the plus sign to add a component.
2. Select the Shop Connector as a component.
3. Select an operation:



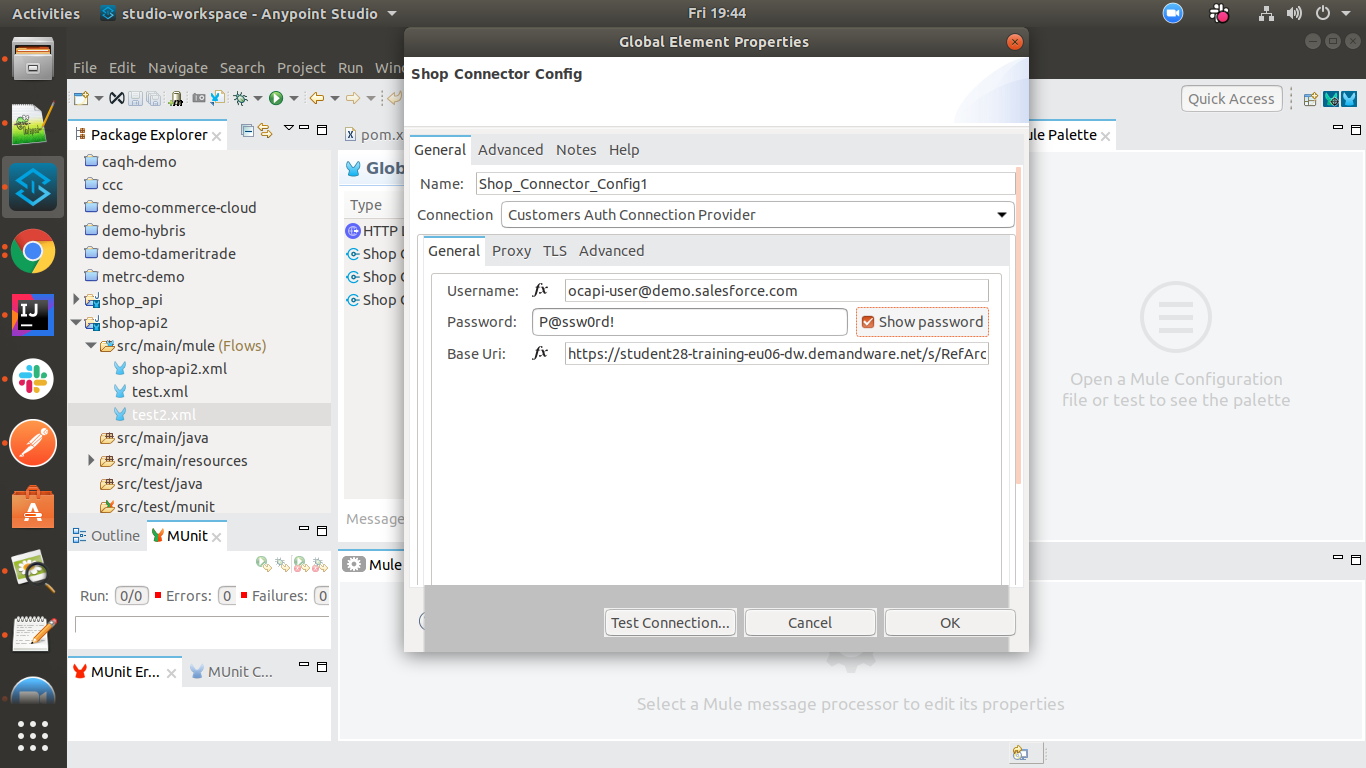
1. Configure the Global element for the connector:



**For Client Id Connection Provider**

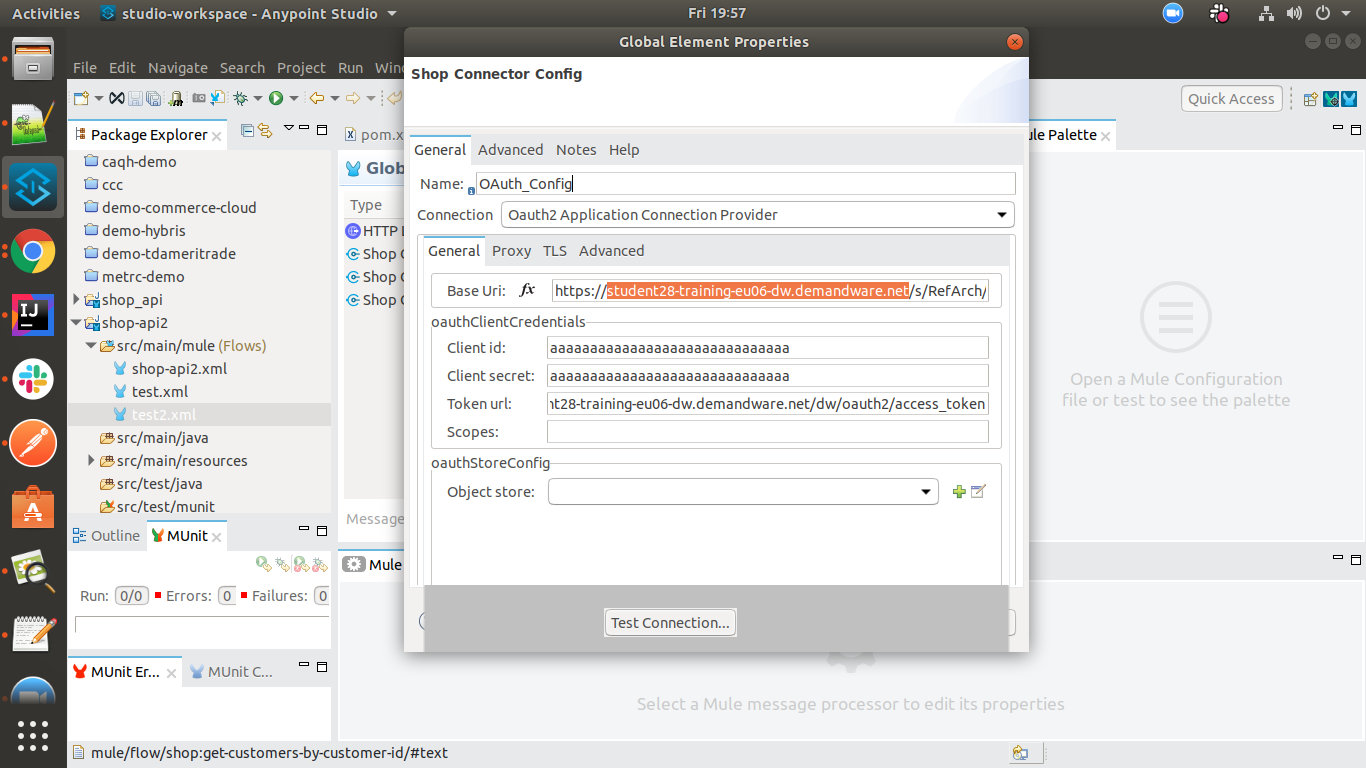
|  |  |
| --- | --- |
| **Field** | **Description** |
| **Base URI** | URL to access API |
| **Client Id** | client id for authentication |

**For Customer Auth Connection Provider**



|  |  |
| --- | --- |
| **Field** | **Description** |
| **Base URI** | URL to access API |
| **Username** | username of commerce cloud account |
| **Password** | password of commerce cloud account |
|  |  |

**For OAuth2 Application Connection Provider**

****

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Base URI** | URL to access API |
| **Username** | Username of commerce cloud account |
| **Password** | Password of commerce cloud account |
| **Token URI** | URL to access token |
| **Client Id** | Client Id of commerce cloud account |
| **Client Secret** | Client Secret of commerce cloud account |

1. Fill the required parameters (if any) for the selected operation:

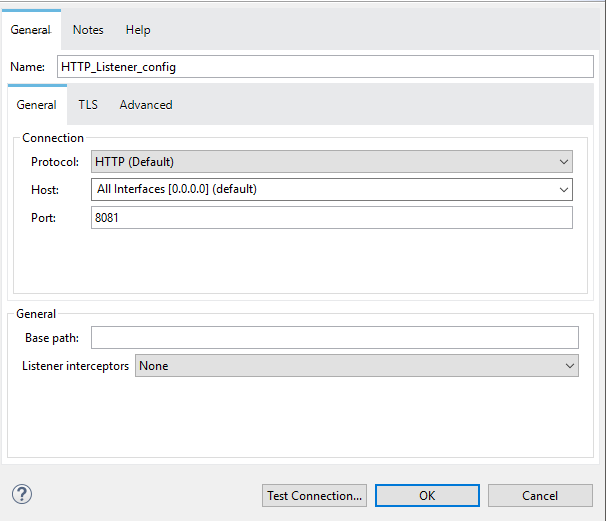
# Use cases – Connectivity with salesforce

This use-case demonstrates the interaction between Commerce cloud and Salesforce systems using commerce cloud connector. It utilises POST, GET, PUT and DELETE operations connectors.

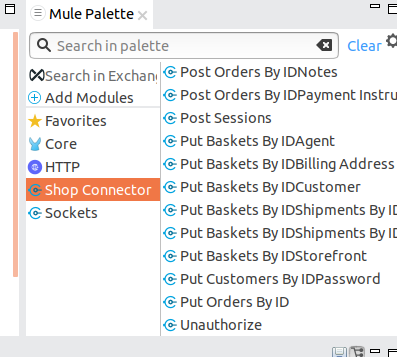
* The flow creates an account in Salesforce and using that account it will create one customer account in application.
* Once the Account created in Salesforce it will fetch the account Id and use that account Id for the remaining operations
* then it is creating basket in commerce cloud, it will save the basket id and item id after that it will update in the salesforce.
* Using the above details just post the order.

For running this use-case we need the following configurations as prerequisites:

1. Drag and drop a HTTP Listener in the canvas.
2. In the Listener properties, give a path you want to use to trigger the listener.
3. Add a new Configuration as follows,



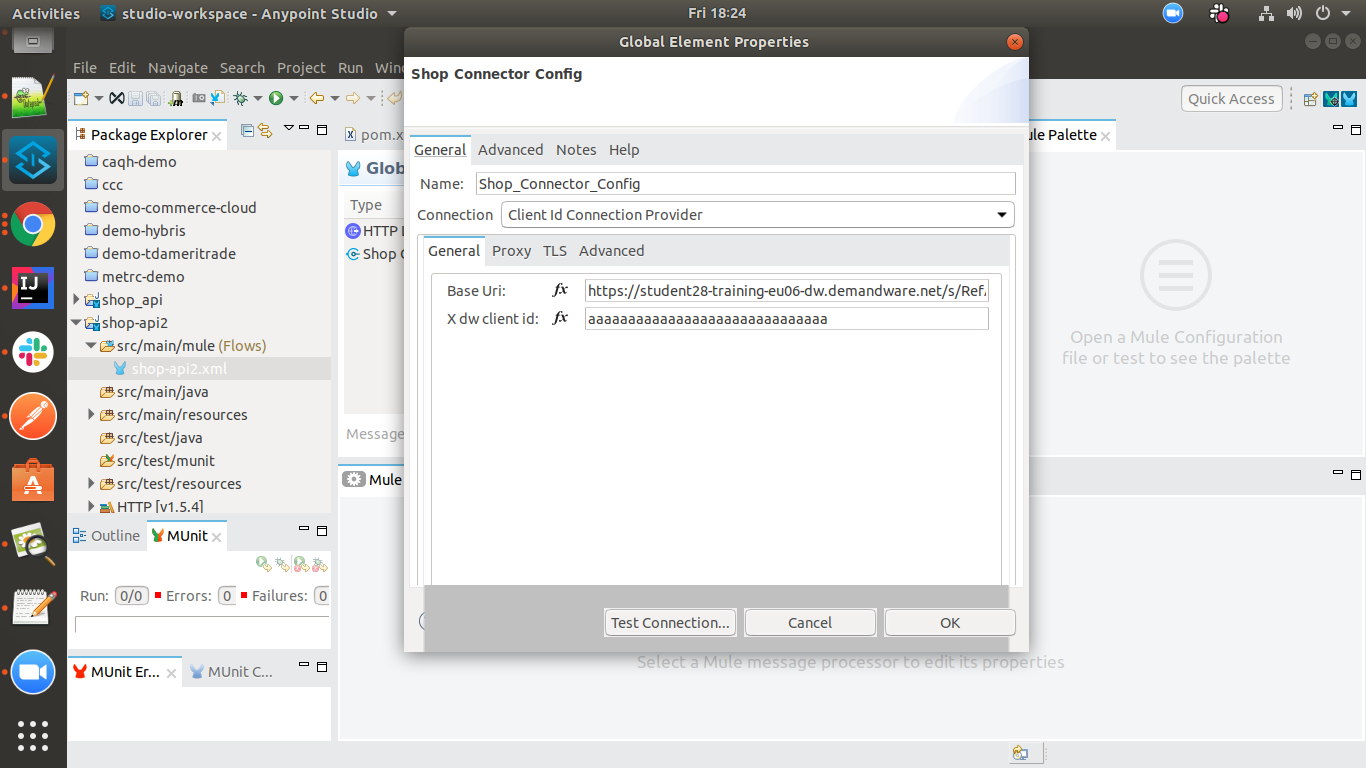
1. Test the connection and click on Okay.
2. Make sure your mule palette has Salesforce and commerce cloud modules. If you do not have Salesforce module in your palette, go to add module -> Salesforce and drag it to your palette.



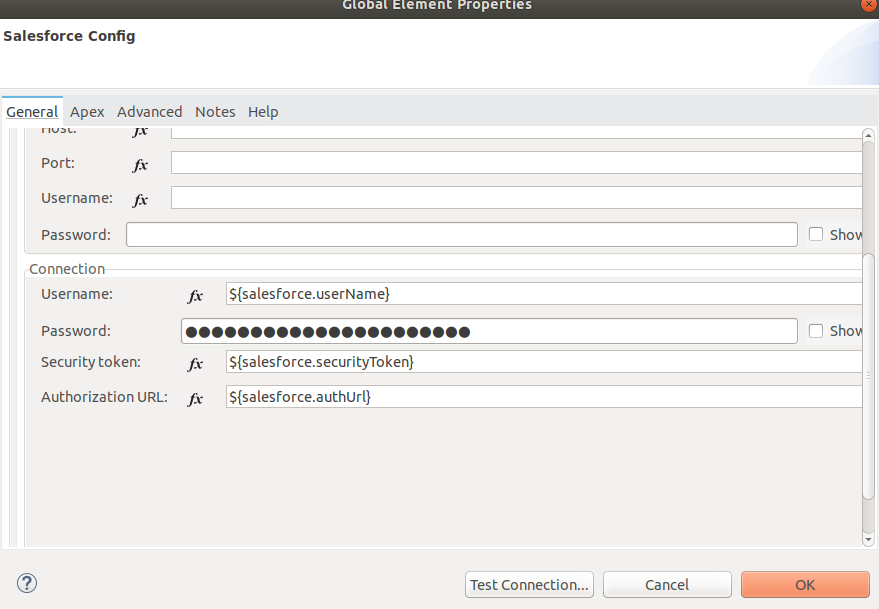
1. Now add configurations for commerce cloud.
2. Go to global-configurations.xml global elements -> create -> Connector Configuration -> Shop Connector Config : Client Id Connection Provider
3. Add following properties from application.properties

Base URI: <Shop api uri>

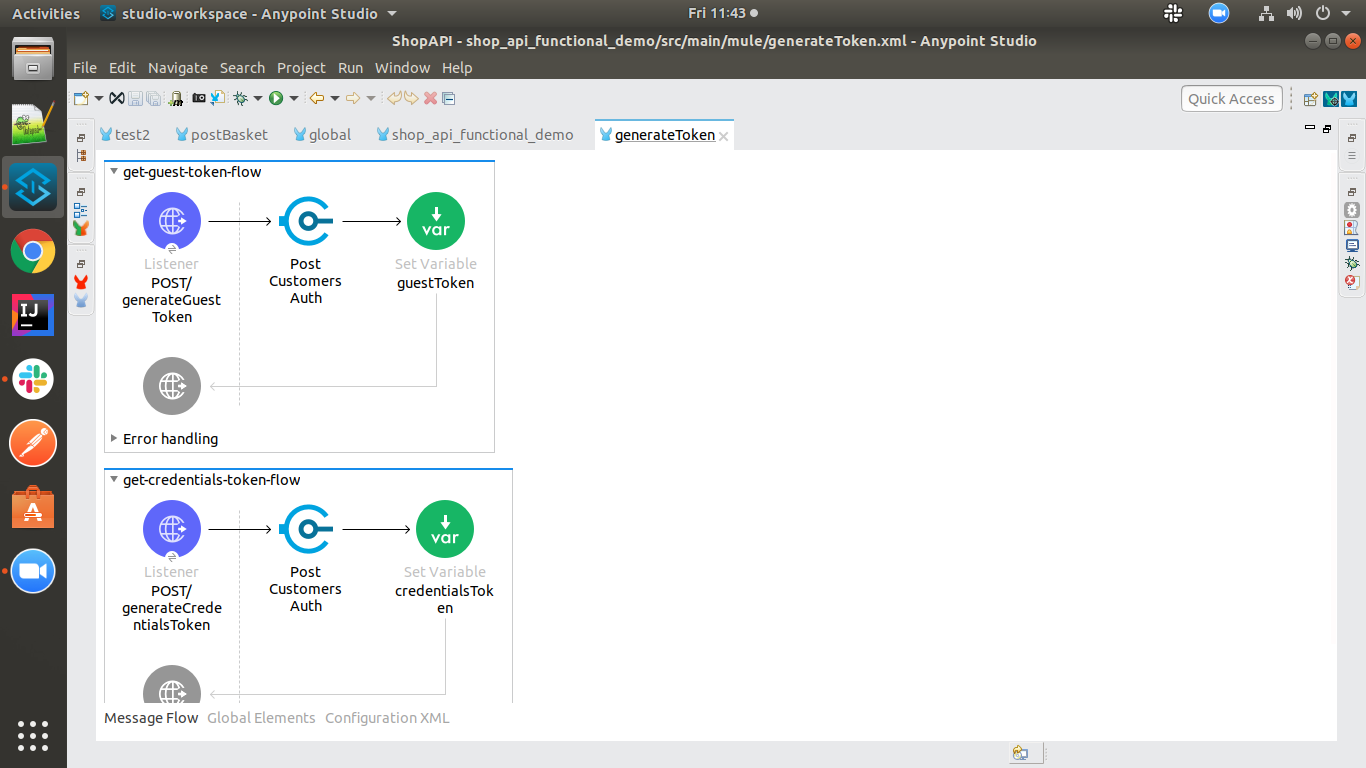
Client Id: <Client id>

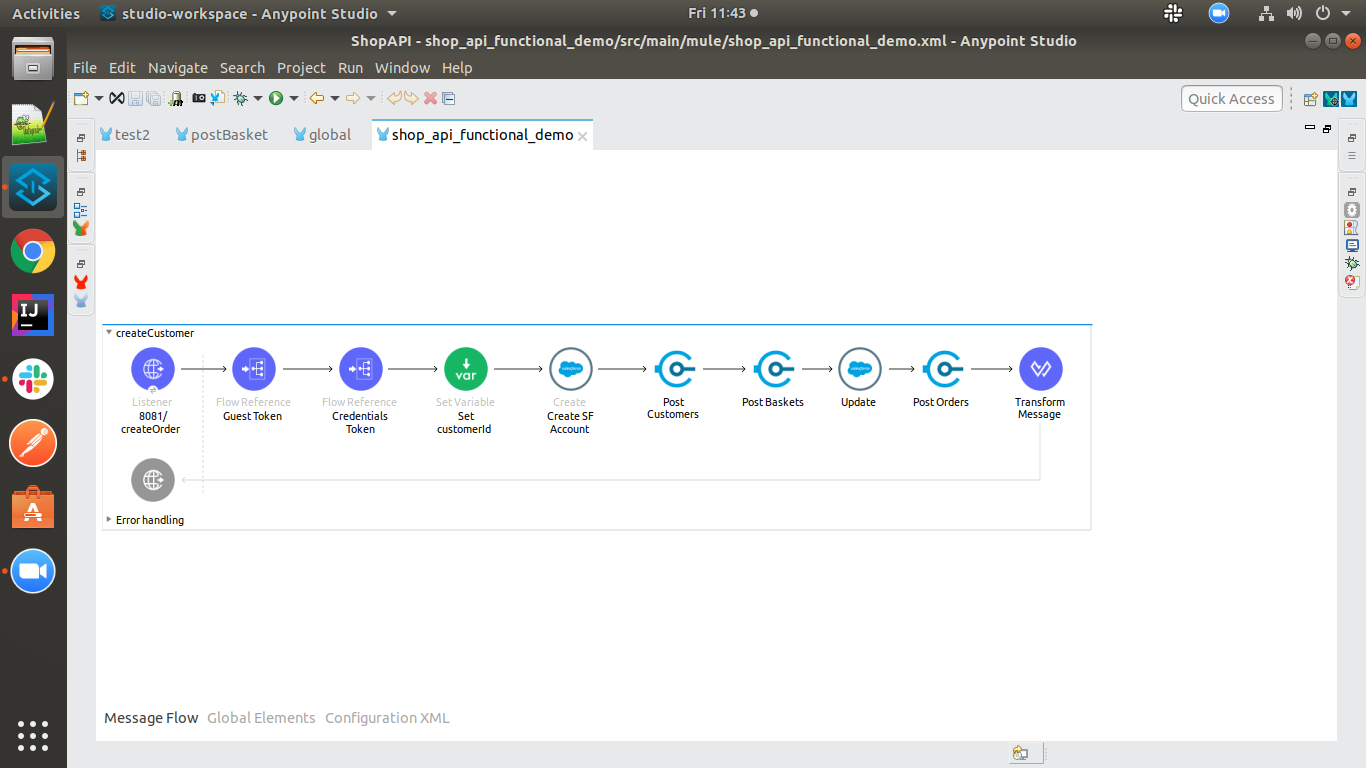


1. Add Salesforce configuration.
2. Go to global-configurations.xml -> global elements -> create -> Connector Configuration -> Salesforce Configuration
3. Add following properties from application.properties



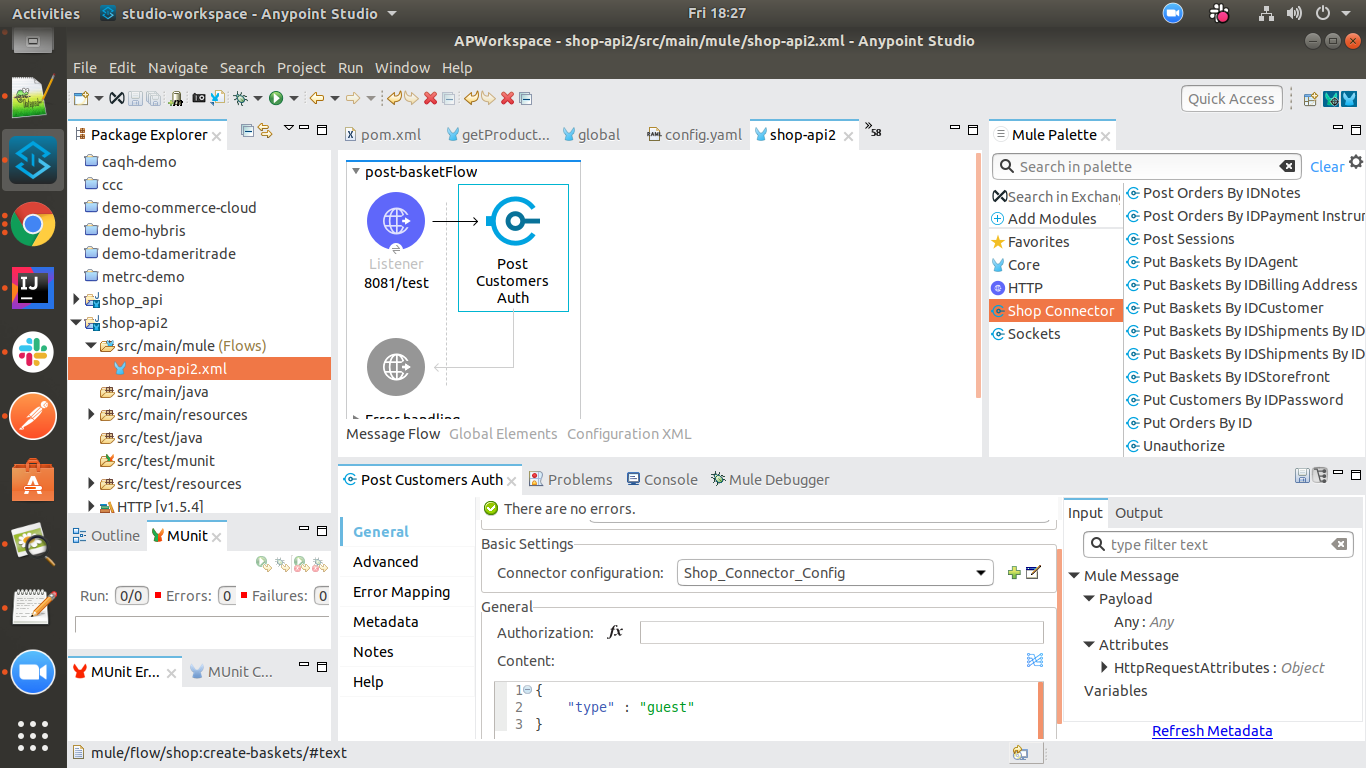
1. Create a flow with the components displayed in the image below:



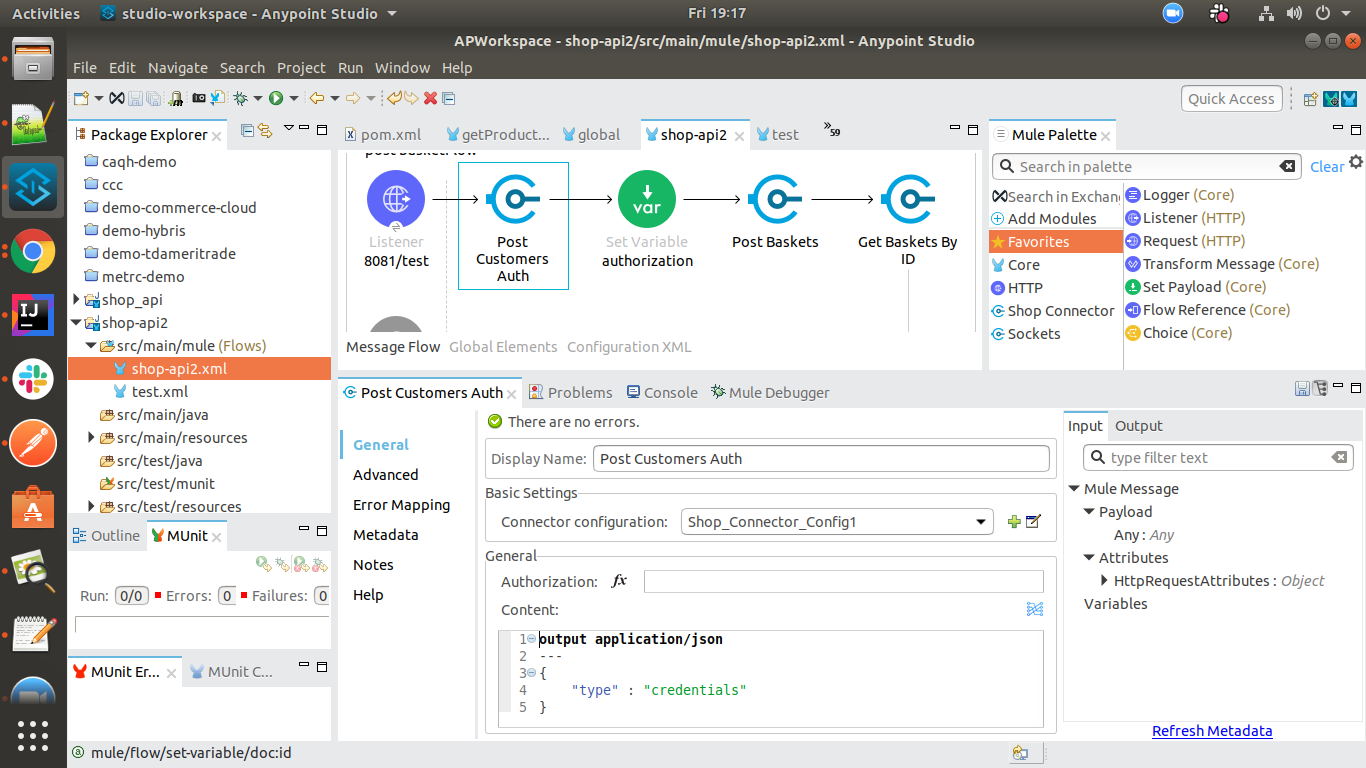


1. Individual mappings for each component are illustrated in below screenshots:

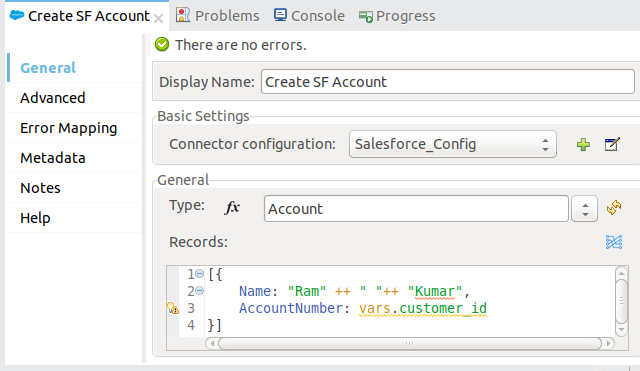
Create Guest token for a user in Shop Connector:



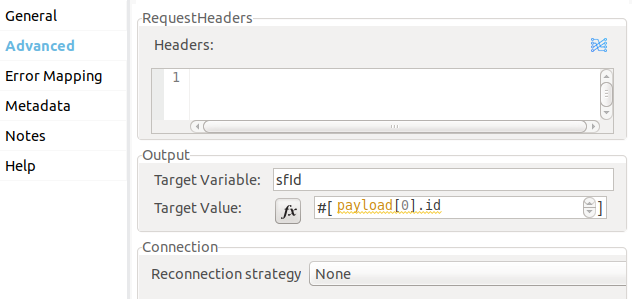
Create Customer token for a customer in Shop Connector:



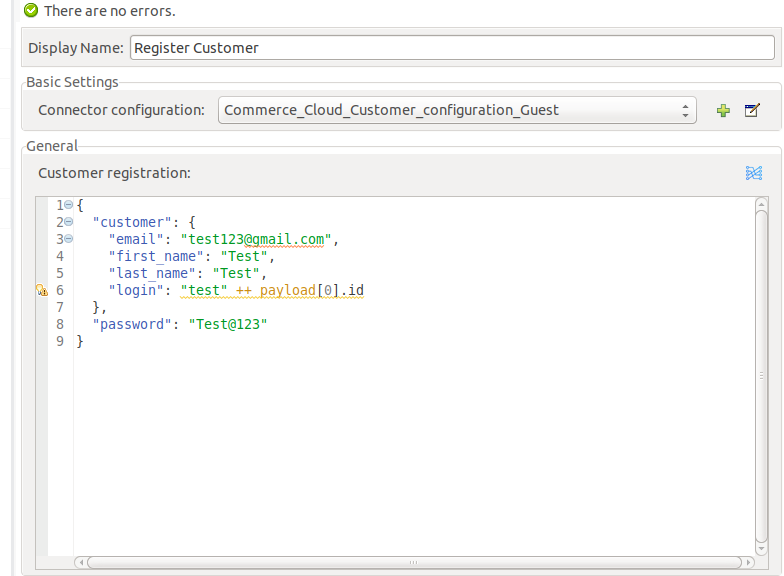
Create account in Salesforce:



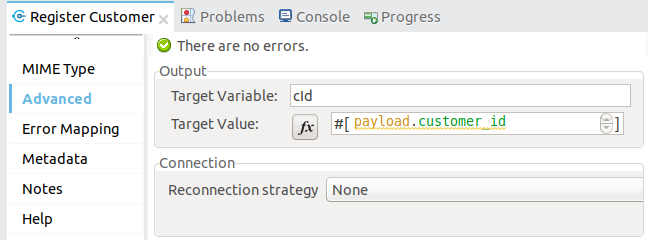
Store Salesforce account id:



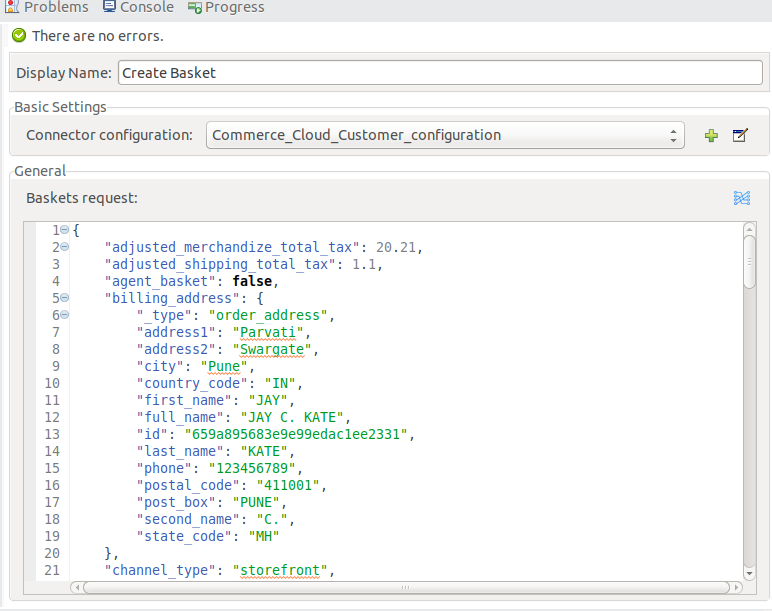
Create Commerce Cloud Customer Account:



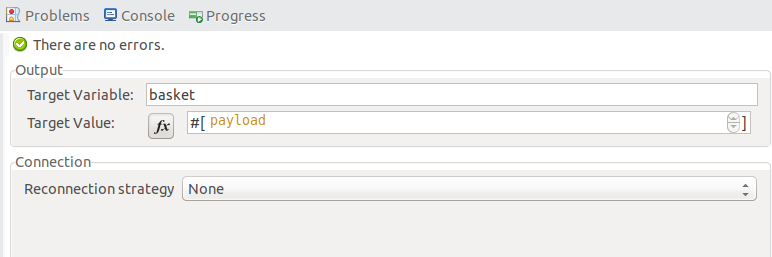
Setting customer id:



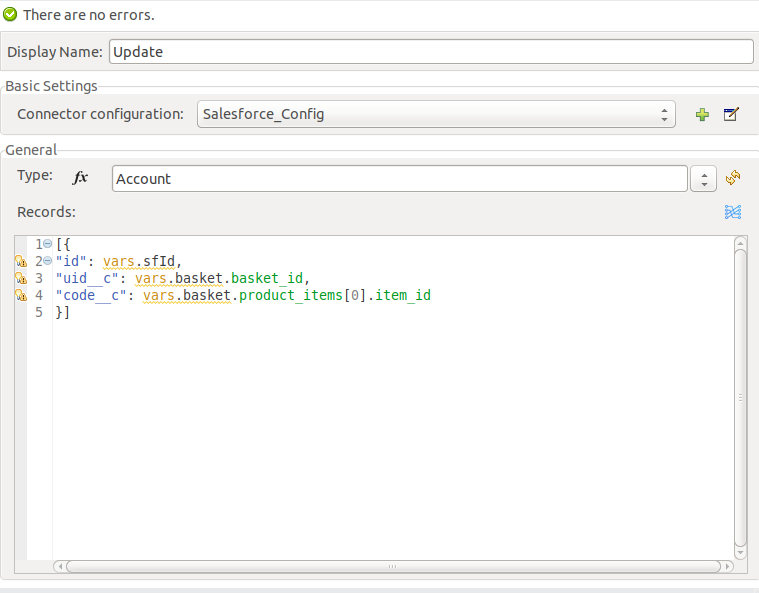
creating basket:



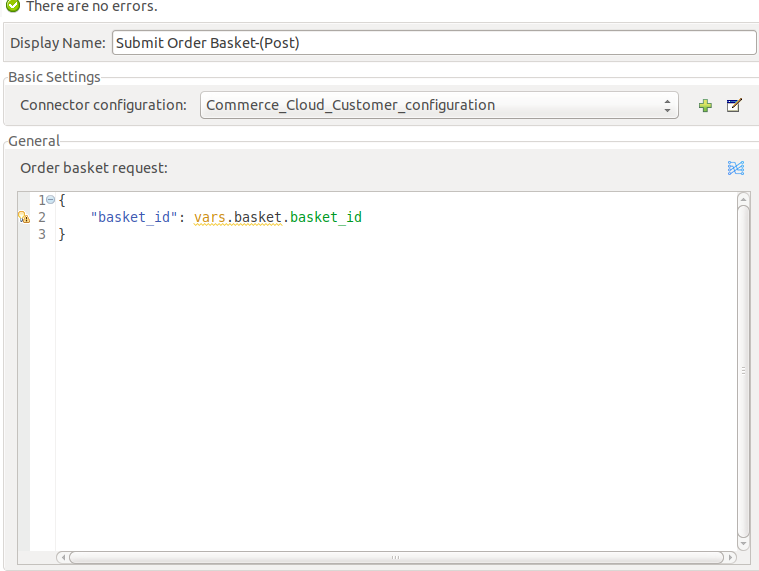
store the payload of basket:



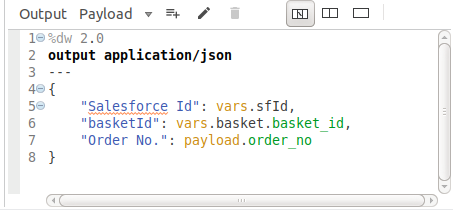
update details in SF:



Creating order:



Printing updated response:

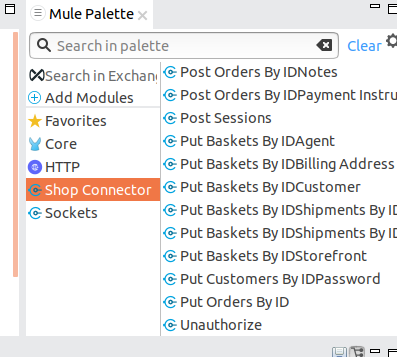


# APPENDIX A

# Install Shop Connector in Anypoint Studio

Developers can add the Shop Connector module in Anypoint Studio, by following the below steps:

* Open you Anypoint Studio
* Search for the Shop Connector in your mule pallet



# Configure Shop Connector In Anypoint Studio

You will require the credentials for creating your access and refresh token, which will be required for the connector.

## **Authentication**

Provide following credentials in configuration file

base URI : "Shop API base URI"

Username : "your user name"

Password : "your password"

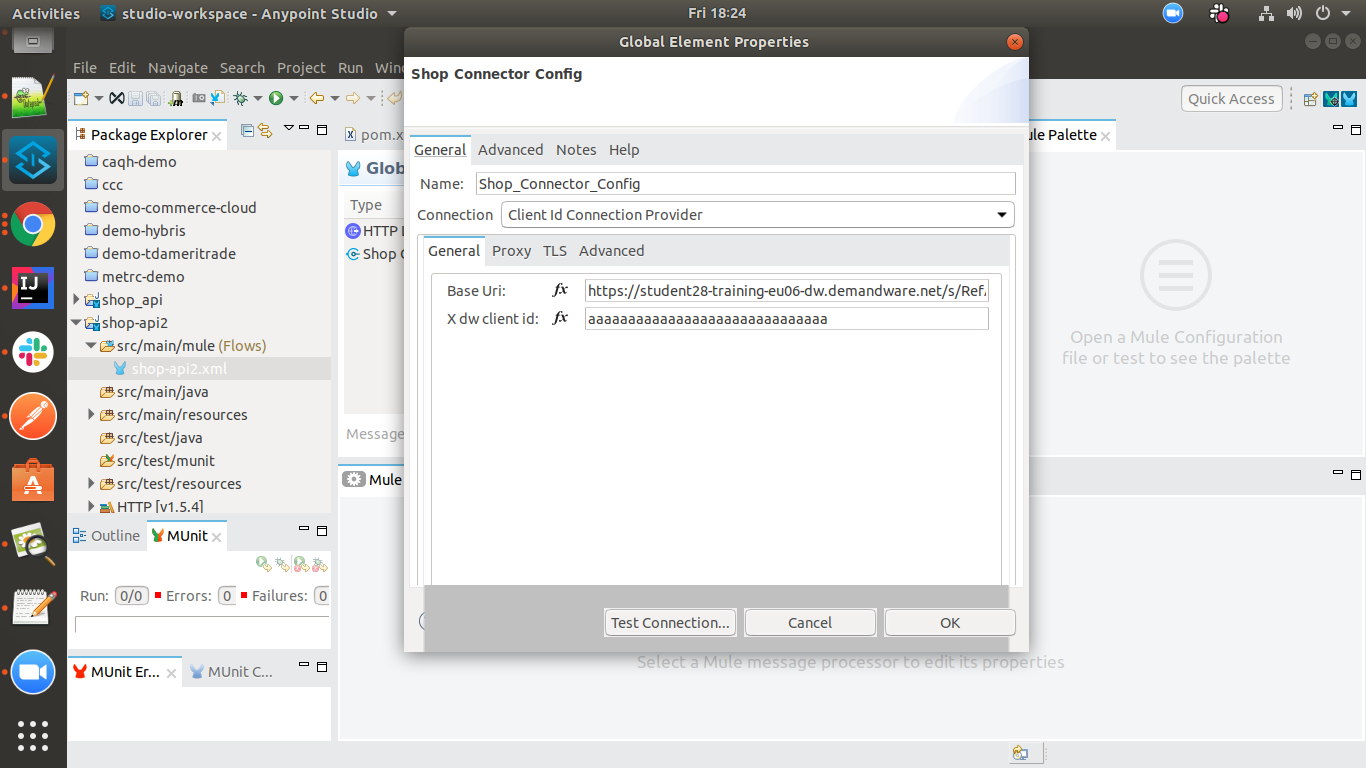
client id : "your client id"

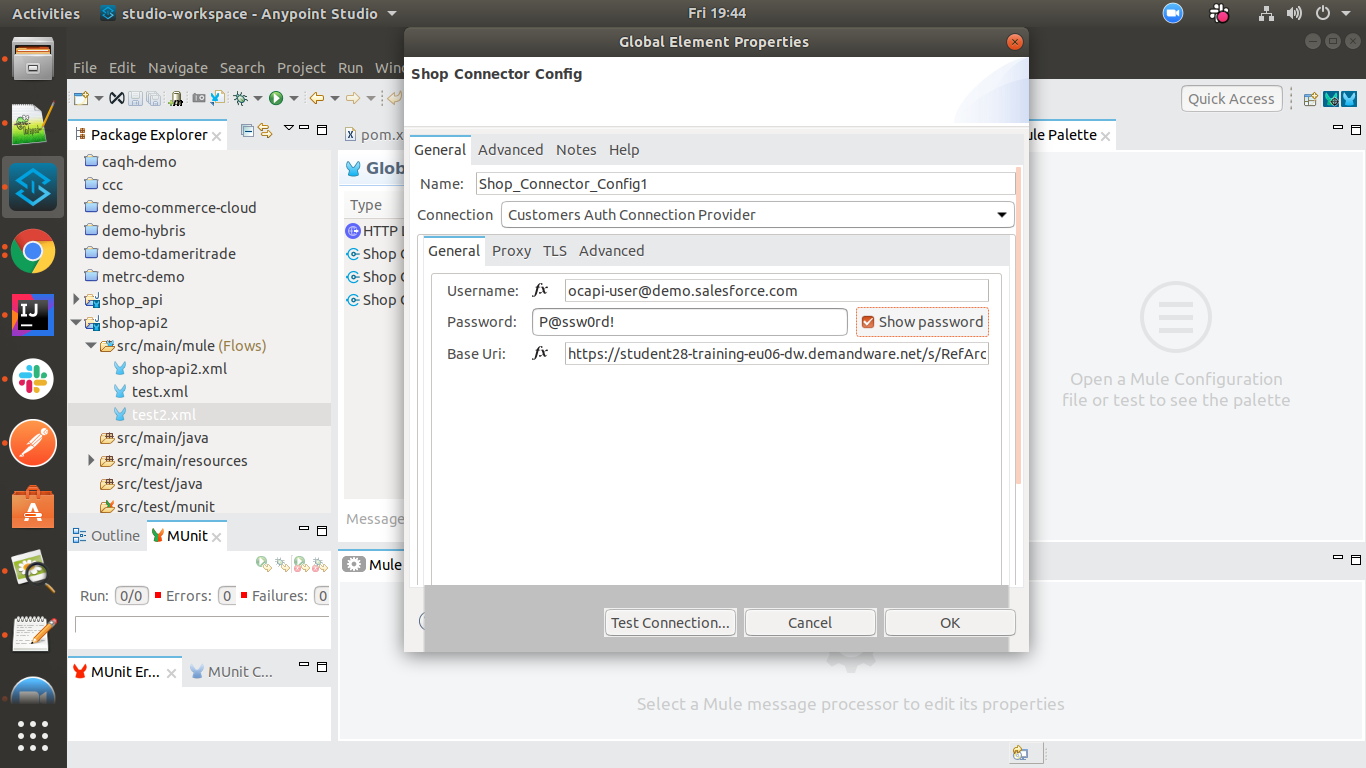
client secret : "your client secret"

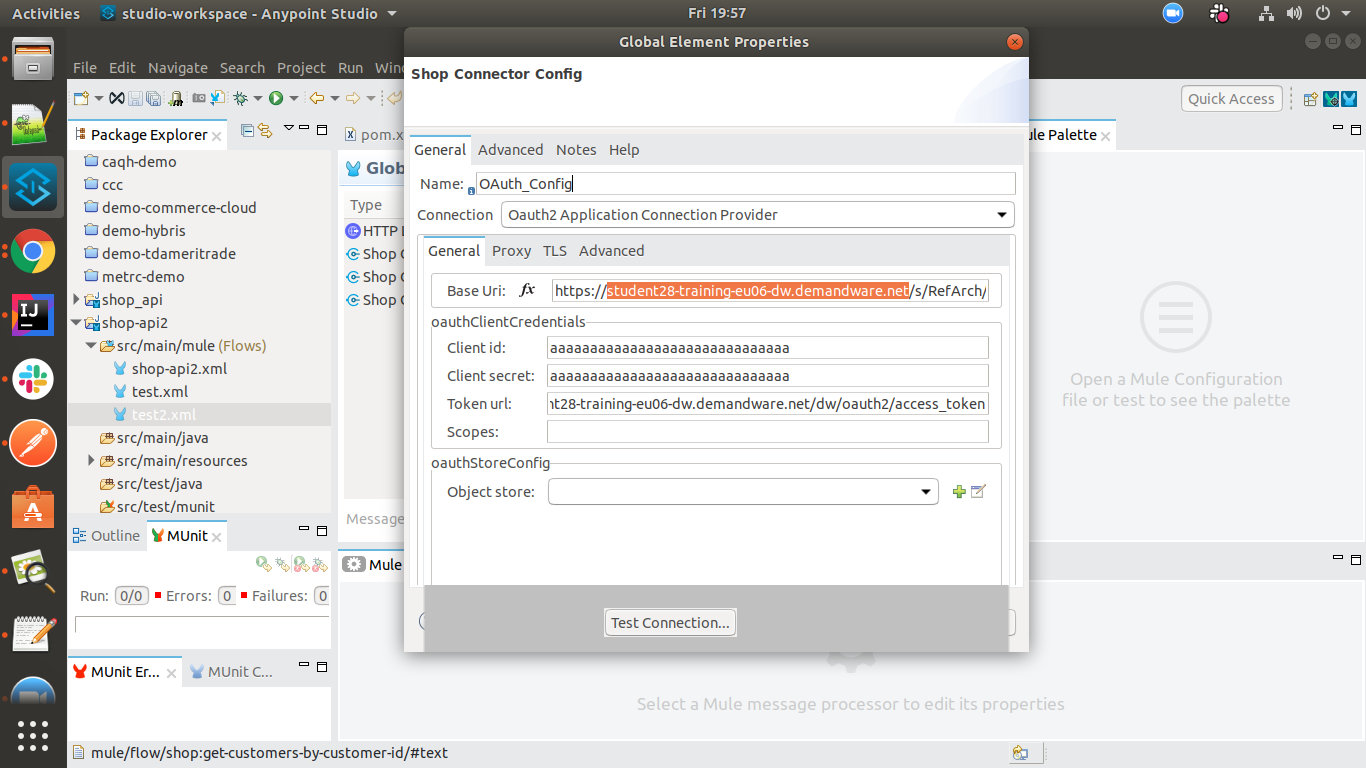
Token Url : "your client id"

## **Configure in Anypoint Studio**

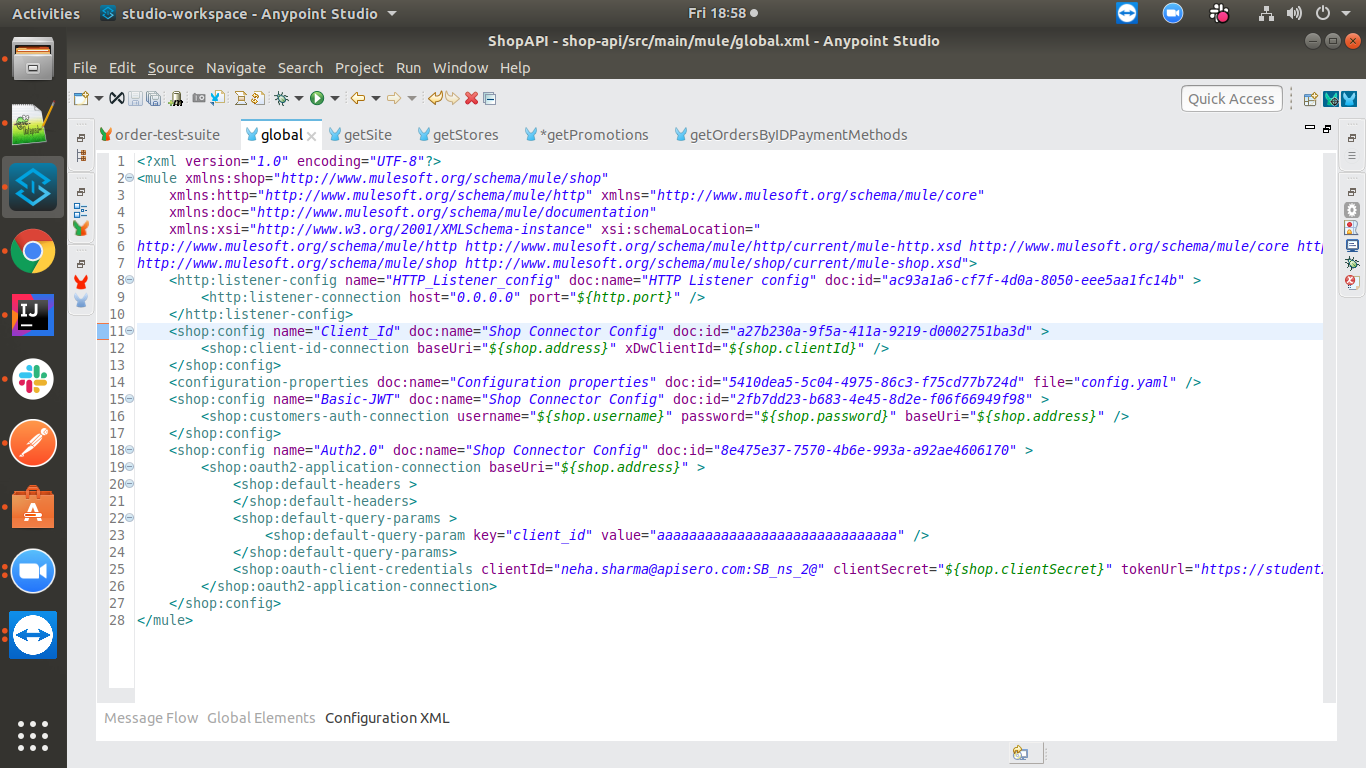
* 1. First thing we need to do to configure the connector, is to have a base URI, token URL, username, password, client id and client secret (depends on each configuration) required to access the APIs, as mentioned in this document previously
  2. Once you have your credentials, drag and drop your connector in the mule pallet and make sure you have a listener in the canvas.
  3. In the Global elements, create a new configuration and use one of the following details:
  4. Once these fields are added, you can test the connection







* 1. You don’t have to add any configurations in the XML because, when the connector configurations are created as mentioned in the document below, the xml will automatically get updated as follows:



## **About Connector Namespace and Schema**

When designing your application in Studio, drag and drop the connector in your canvas and the Namespace and schema get populated in the config file as below,

Namespace: *http://www.mulesoft.org/schema/mule/shop*

Schema Location: *http://www.mulesoft.org/schema/mule/core http://www.mulesoft.org/schema/mule/core/current/mule.xsd*