Metrc API Services

Technical Design Document

# DOCUMENT REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Release Date** | **Revised by** | **Comments/ Indicate Sections Revised** |
| 1.0.0 | 22/04/2020 | Connector Development Team |  |
| 2.0.0 | 25/04/2020 | Connector Development Team | 7 more operations added |

Table of Contents

[DOCUMENT REVISION HISTORY 2](#_gjdgxs)

[A.](#_30j0zll) [INTRODUCTION 4](#_30j0zll)

1. **[Purpose](#_1fob9te)** [4](#_1fob9te)
2. **[Background](#_3znysh7)** [4](#_3znysh7)
3. **METRC****[API Service](#_1ci93xb)** [4](#_1ci93xb)
4. **[Overview](#_2et92p0)** [5](#_2et92p0)

[B.](#_3dy6vkm) MAPPING – BUSINESS FUNCTIONS CALLS to TECHNICAL OPERATIONS 5

1. **[List of operations supported by the Metrc Connector:](#_1t3h5sf)** [6](#_1t3h5sf)

[C. Configurations for Metrc Connector 17](#_4d34og8)

1. **[Authentication](#_1y810tw)** [17](#_4d34og8)

D. To connect in design center 18

E. Use cases - Connectivity with Salesforce 21

**APPENDIX A** 29

1. Install Metrc Connector In anypoint studio 29
2. Configure Metrc Connector In anypoint studio 31
3. Authentication 31
4. Configure in Anypoint Studio 31
5. About Connector Namespace and Schema 31

# **INTRODUCTION**

## **Purpose**

This specification document outlines the instructions for Participating Organizations (PO) to access the API service offered in the Metrc 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 Metrc API URL.

## **Background**

The **Metrc** (Marijuana Enforcement Tracking Reporting Compliance) Web **API** allows developers to interact with the**Metrc** regulatory compliance system. The **API** can pull information from **Metrc** into a point of sale system.

Metrc connector enables users to integrate their existing applications with Metrc REST APIs.

## **METRC** **API Service**

METRC APIs provide access to various services for the following workflows:

* + **Packages :** This service provides details on Packages.
  + **Transfers :** This service provides details on Transfers.
  + **Sales :** This service provides details on Sales like creating a sales receipt, updating a sales receipt, deleting a sales receipt, getting all sales receipts.

## **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 Metrc 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.

## **List of Operations Supported by the Metrc Connector:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Business**  **Function** | **Operations** | **Description** | **Inbound** | **Outbound** |
| **Packages** | | | | | |
| 1 | Get Active Packages | GET Request | Gets the list of all all packages which are active. | Required field is licenseNumber.  optional fields are: lastModifiedStart, lastModifiedEnd | Return an object containing list of all packages which are active. |
| 2 | Create Packages | POST Request | Creates a package | Required field is licenseNumber. | No response |
| 3 | Finish Packages | POST Request | Finishes a package | Required field is licenseNumber. | No response |
| 4 | Unfinish Packages | POST Request | Unfinishes a Package | Required field is licenseNumber. | No response |
| 5 | Adjust Packages | POST Request | Adjusts a Package | Required field is licenseNumber. | No response |
| **Transfers** | | | | | |
| 6 | Get Incoming Transfers | GET Request | Gets the list of all incoming transfers | Required field is licenseNumber.  optional fields are: lastModifiedStart, lastModifiedEnd | Return an object containing list of all incoming transfers. |
| 7 | Get outgoing Transfers | GET Request | Gets the list of all outgoing transfers | Required field is licenseNumber.  optional fields are: lastModifiedStart, lastModifiedEnd | Return an object containing list of all outgoing transfers. |
| 8 | Update Transfers Template | PUT Request | Update the existing transfer template | Required field is licenseNumber. | No response |
| 9 | Create Transfers Template | POST Request | Creates a transfer template | Required field is licenseNumber. | No response |
| **Sales** | | | | | |
| 10 | Get Sales Receipts | GET Request | Gets the list of all sales receipts | Required field is licenseNumber.  optional fields are: salesDateStart, salesDateEnd, lastModifiedStart, lastModifiedEnd | Return an object containing list of all sales receipts |
| 11 | Create Sales Receipt | POST request | Creates a new sales receipts | Required field is licenseNumber. | No response |
| 12 | Update Sales Receipt | PUT request | Update an existing sales receipt | Required field is licenseNumber. | No response |
| 13 | Delete Receipt By Id | DELETE request | deletes an existing sales receipt | Required field is licenseNumber. | No response |

# CONFIGURATION FOR METRC CONNECTOR

To use Metrc Connector, you will require the credentials , which will be required for the connector.

## **Authentication**

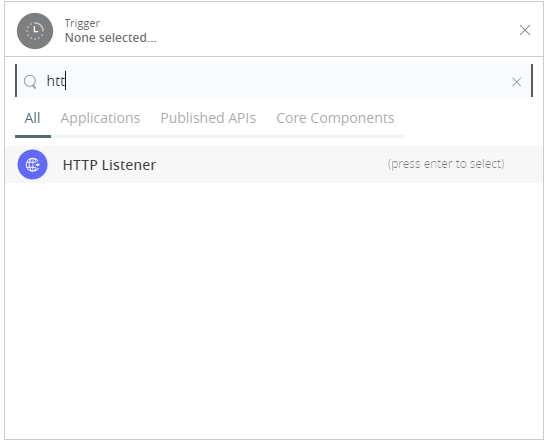
The Metrc API uses Basic Auth to provide access to an API.Vendor-key must be used as username & user-key must be used as password.

1. **Metrc connection**: This connection is used to provide an access to Metrc API.It requires below credentials:

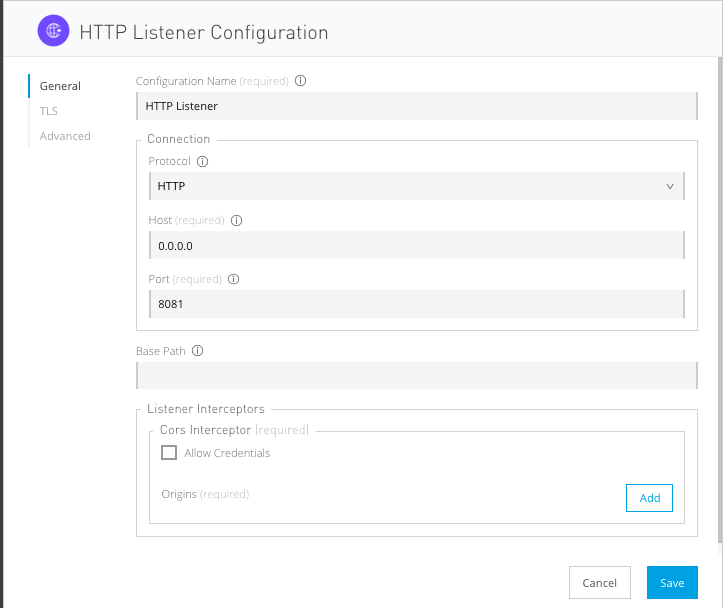
* username: <vendor-key>
* password: <user-key>

# TO CONNECT IN DESIGN CENTER

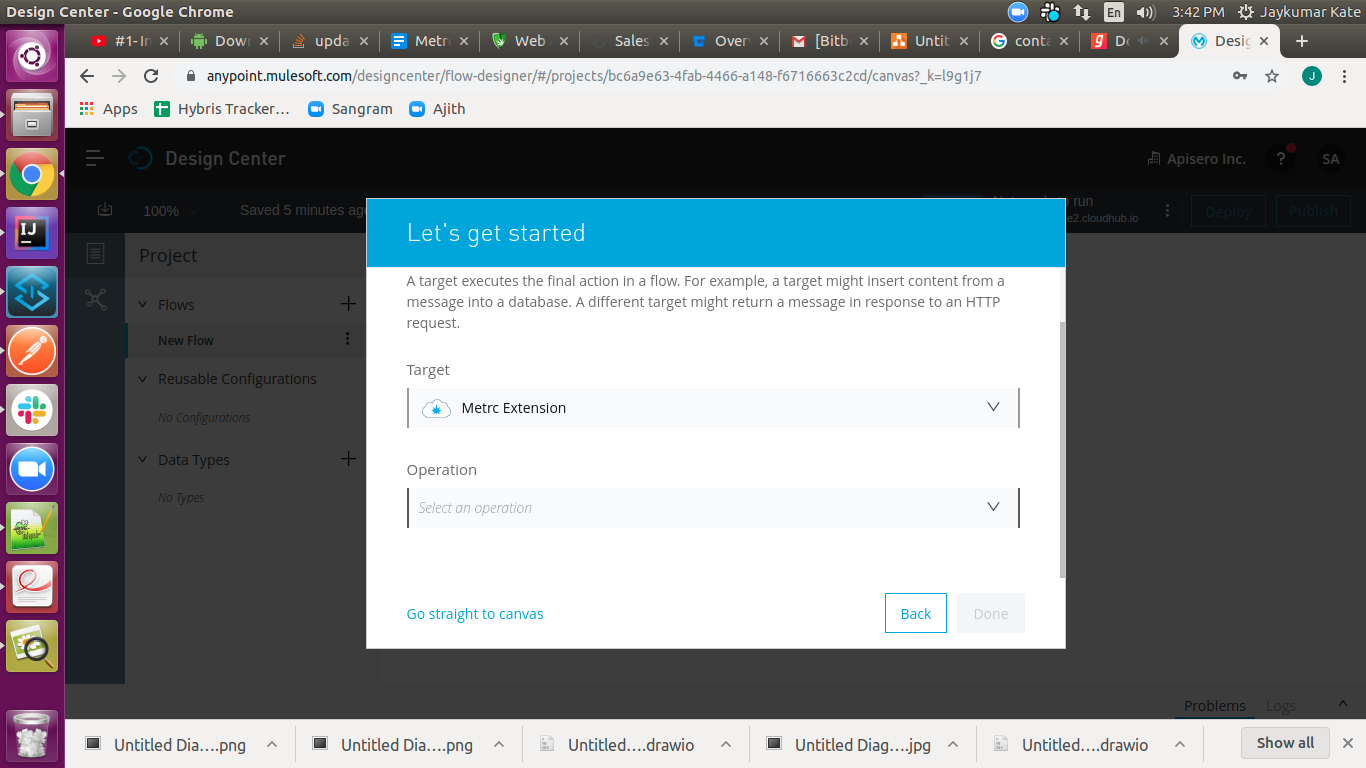
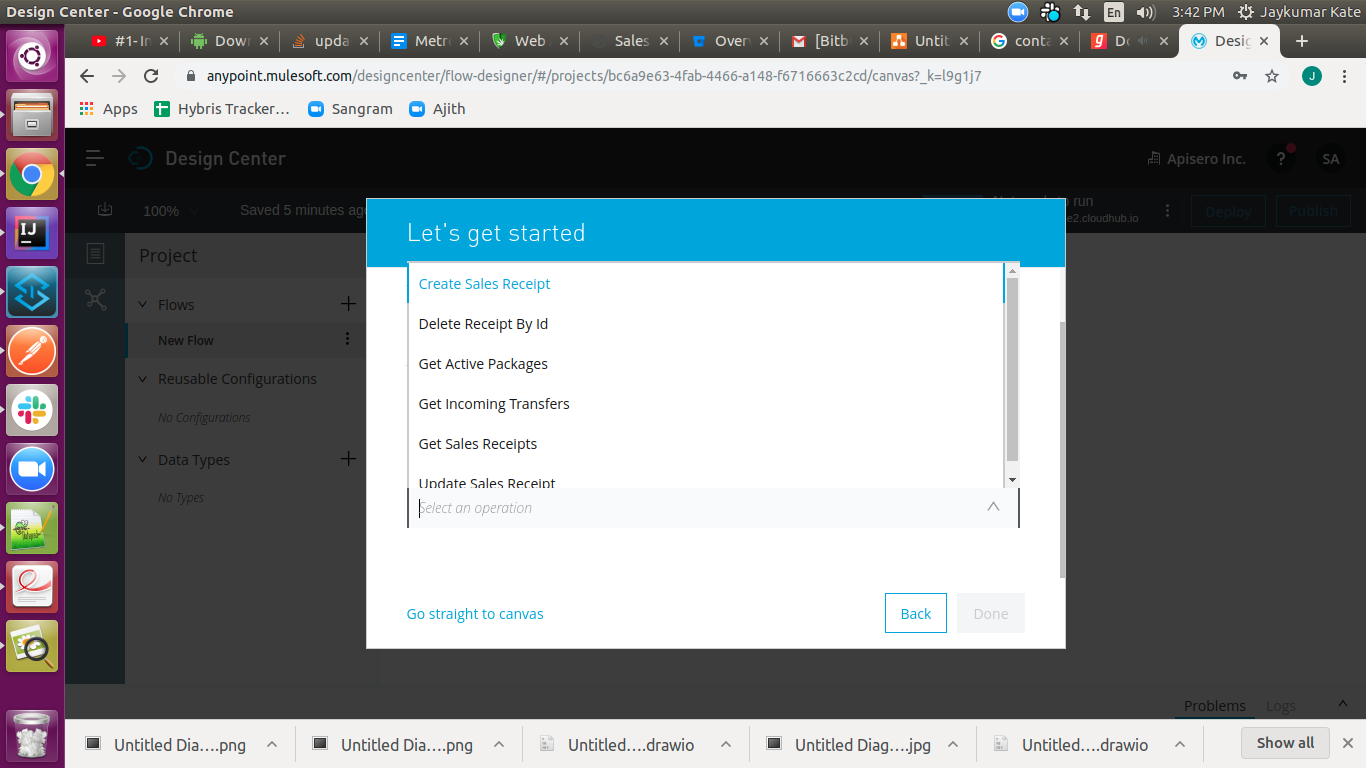
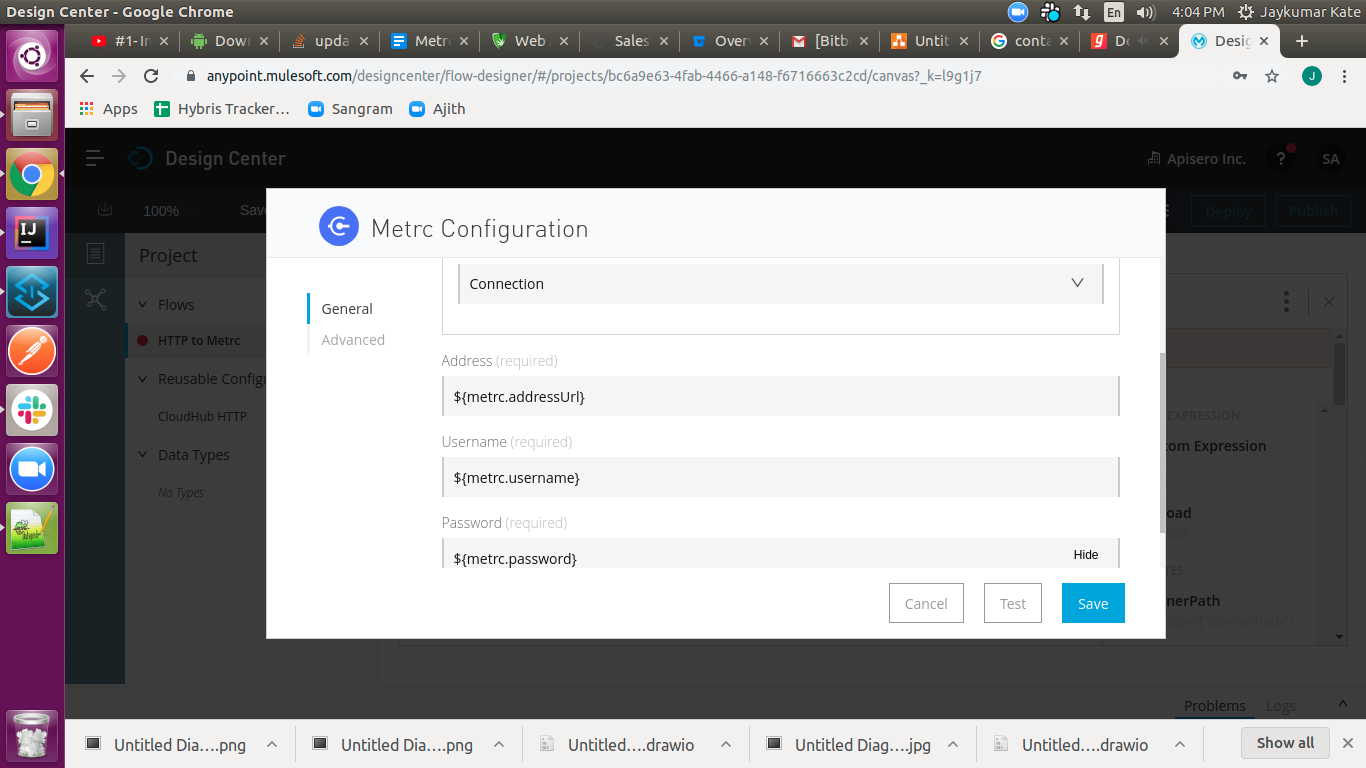
1. In Design Centre, click Create and choose Mule Application.
2. Click a trigger such as an HTTP Listener or the Scheduler trigger.



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 Metrc connector as a component.
3. Select an operation:
4. Configure the Global element for the connector:

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Address** | URL to access Metrc API |
| **Username** | Your vendor-key to access your Metrc API. |
| **Password** | Your user-key to access your Metrc API. |
|  |  |

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

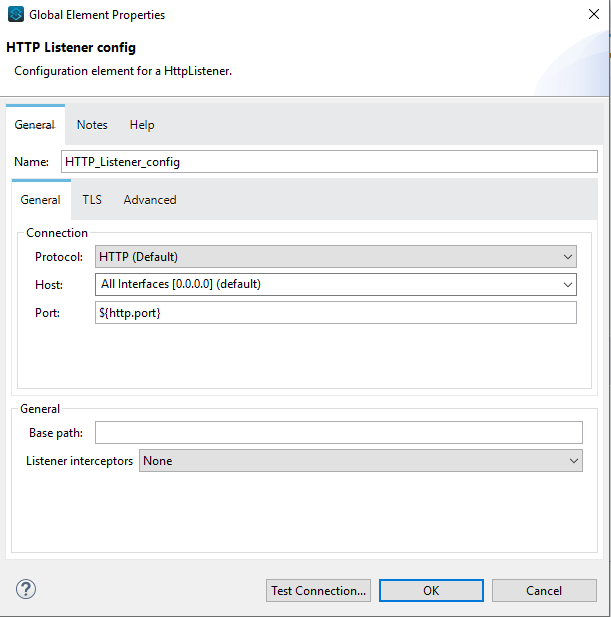
# Use cases – Connectivity with salesforce

This use-case demonstrates the interaction between Metrc and Salesforce systems using Metrc connector. It utilises POST, UPDATE, GET and DELETE operations connectors.

* The flow gets created using “Create Sales Receipts” operation,then “Get Sales Receipt” operation in Metrc systems and also creates account in salesforce.
* Once the Account is created in Salesforce for provided license number then flow will create the sales receipt for provided license number, gets all sales receipts & stores the created receipt id in object store, retrieves the receipt id from object store & updates the created sales receipt.
* In delete receipt flow The receipt will be deleted from Metrc system & salesForce.

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

1. Drag and drop an 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,

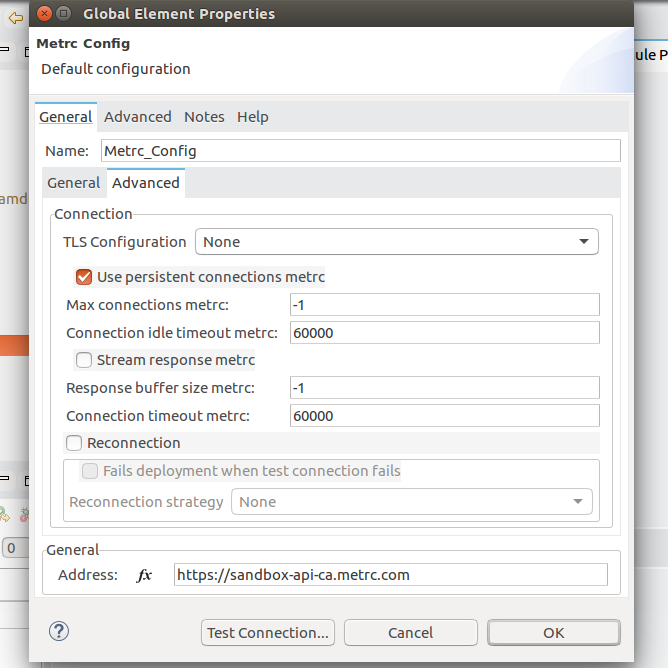
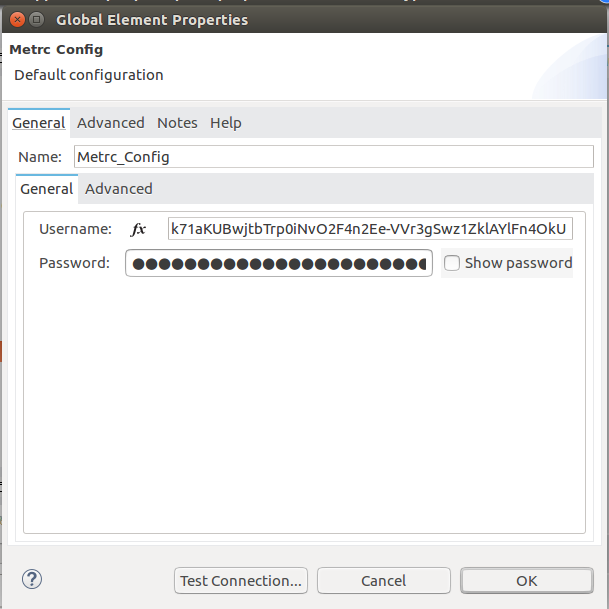


1. Test the connection and click on Okay.
2. Make sure your mule palette has Salesforce and Metrc modules. If you do not have Salesforce module in your palette, go to add module -> Salesforce and drag it to your palette.
3. Now add configurations for Metrc.
4. Go to global-configurations.xml global elements -> create -> Connector Configuration -> Metrc Configuration
5. Add following properties:

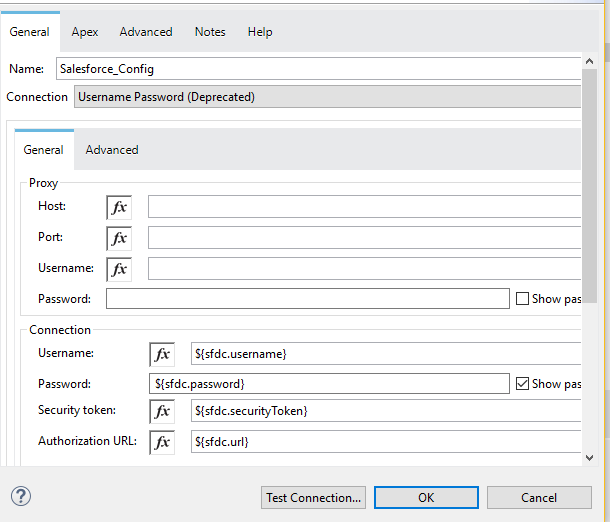
Address Url :

Username :

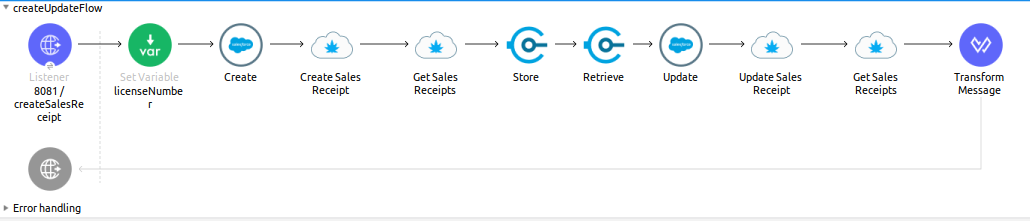
Password :

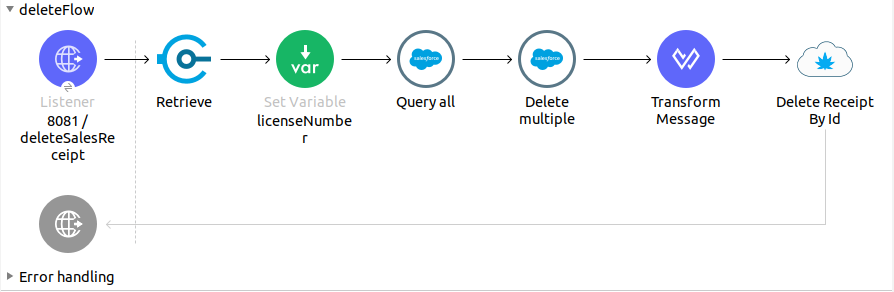


1. Add Salesforce configuration.
2. Go to global-configurations.xml -> global elements -> create -> Connector Configuration -> Salesforce Configuration
3. Add following properties



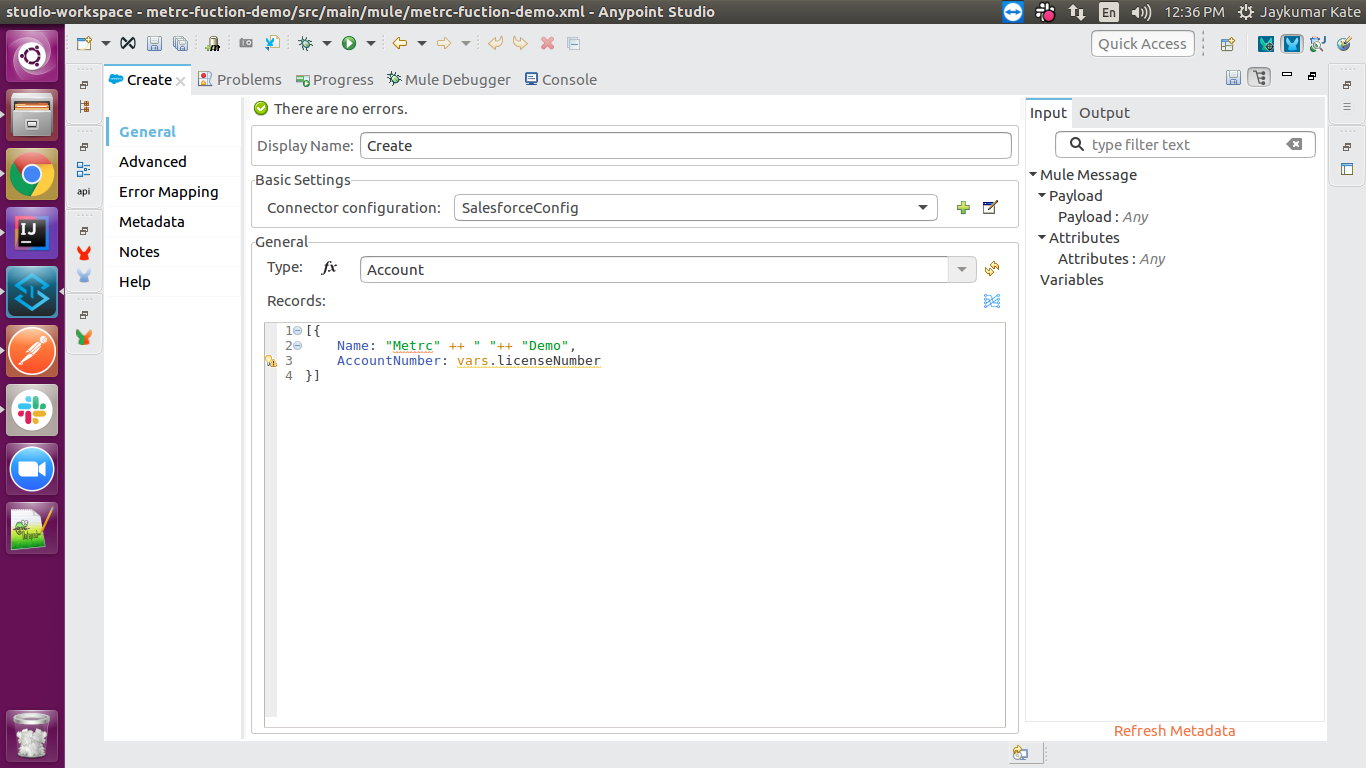
1. Create a flow with the components displayed in the image below:



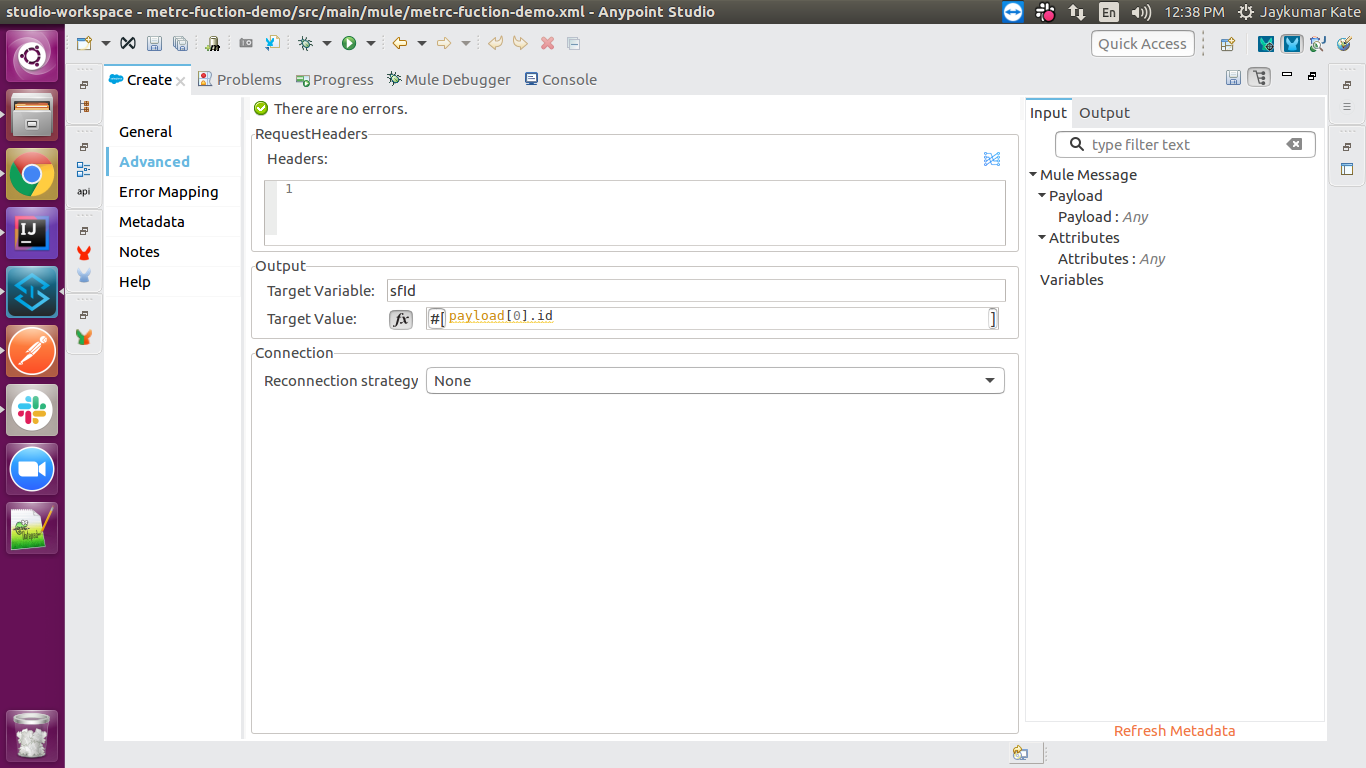


1. Individual mappings for each component are illustrated in below screenshots:

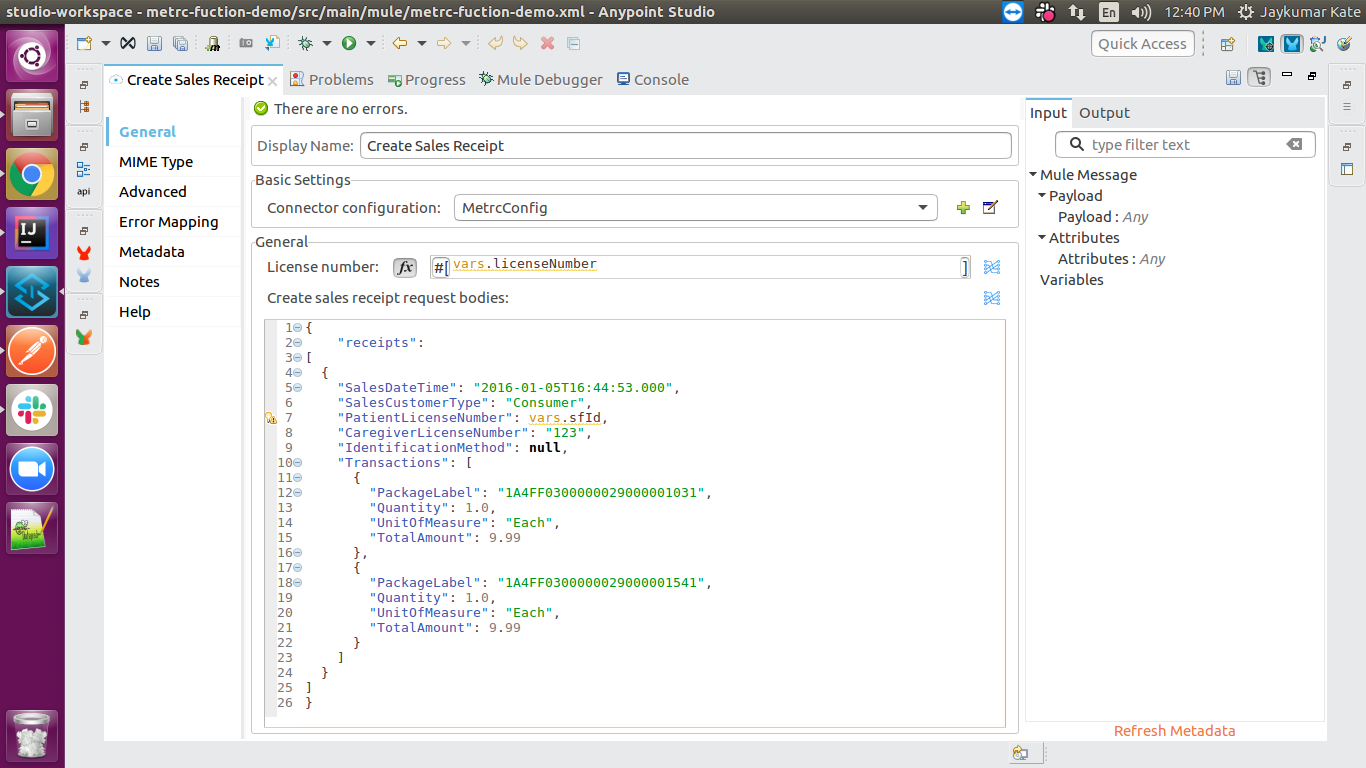
Create SalesForce account request body:



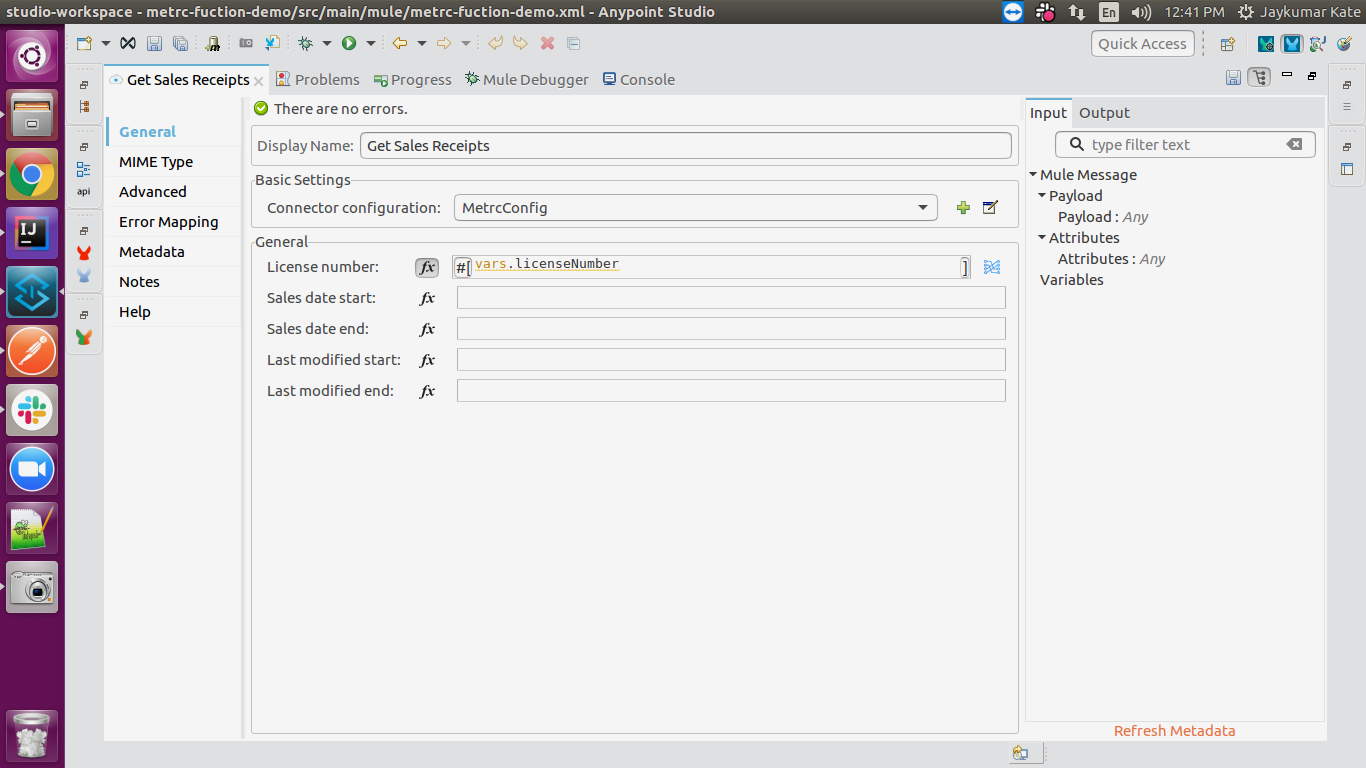
Store Salesforce account id:



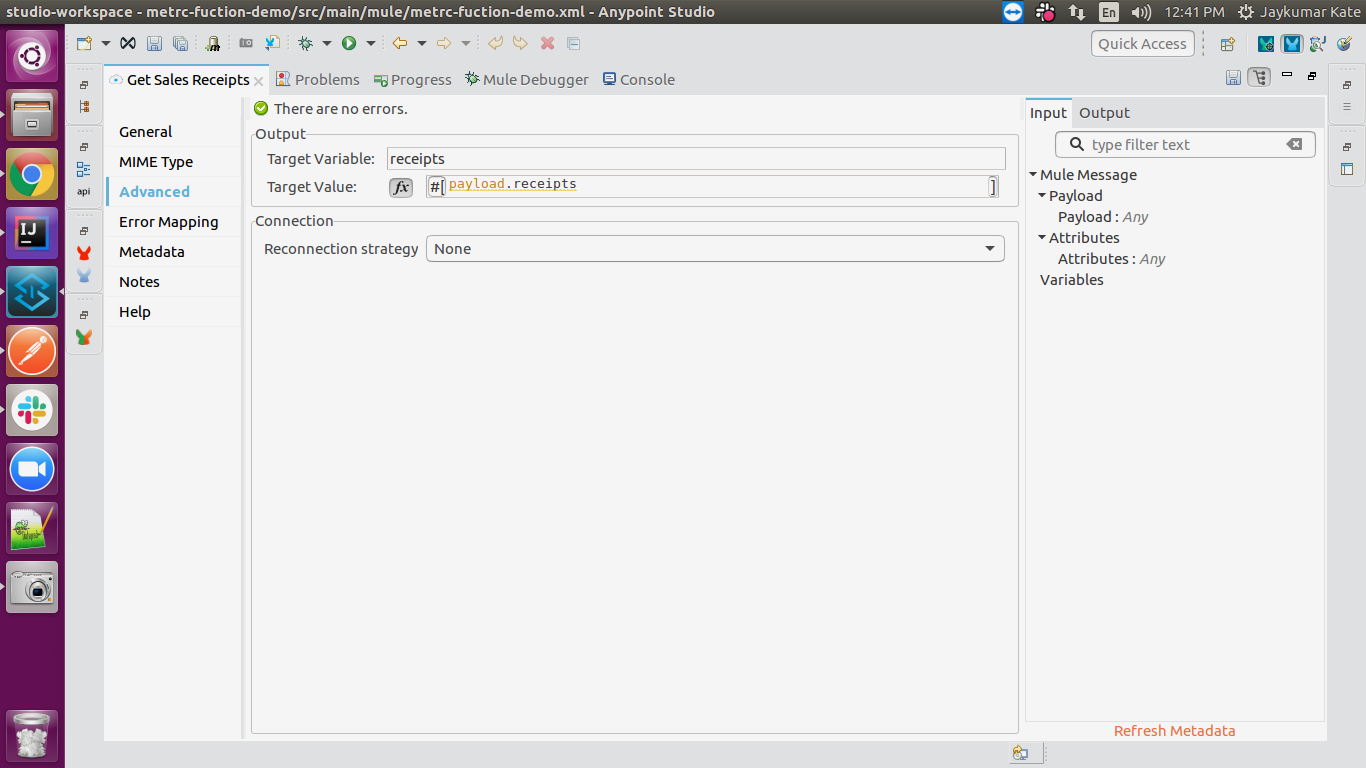
Creating sales receipt:



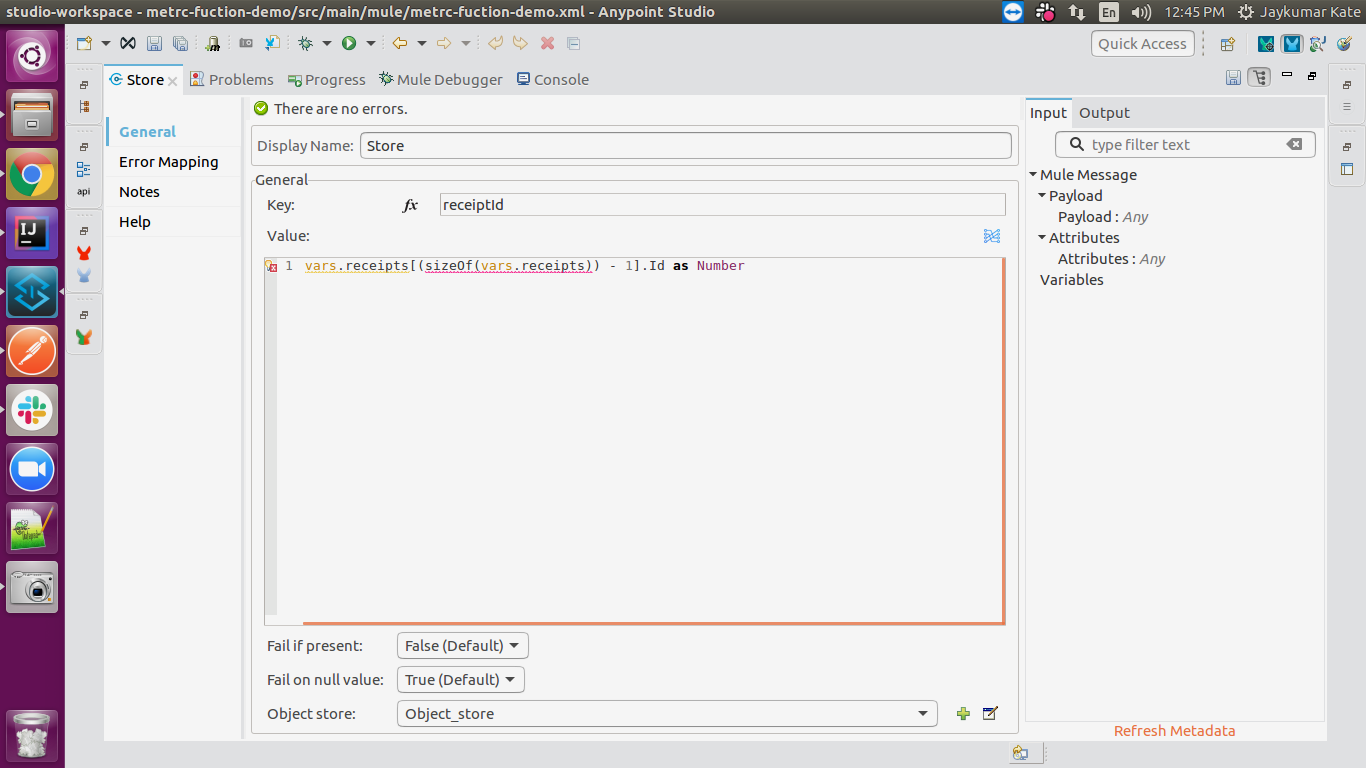
Get all sales receipts :



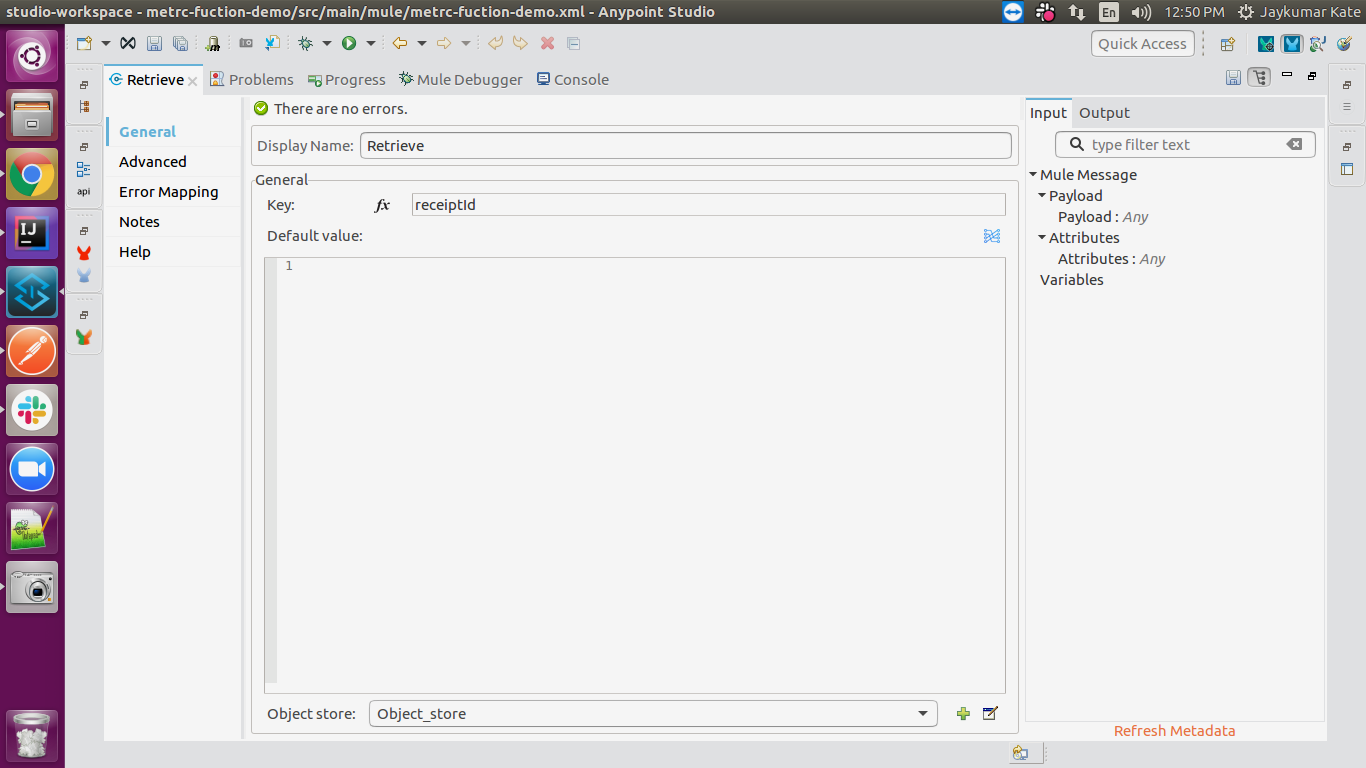
store sales receipts in variable:



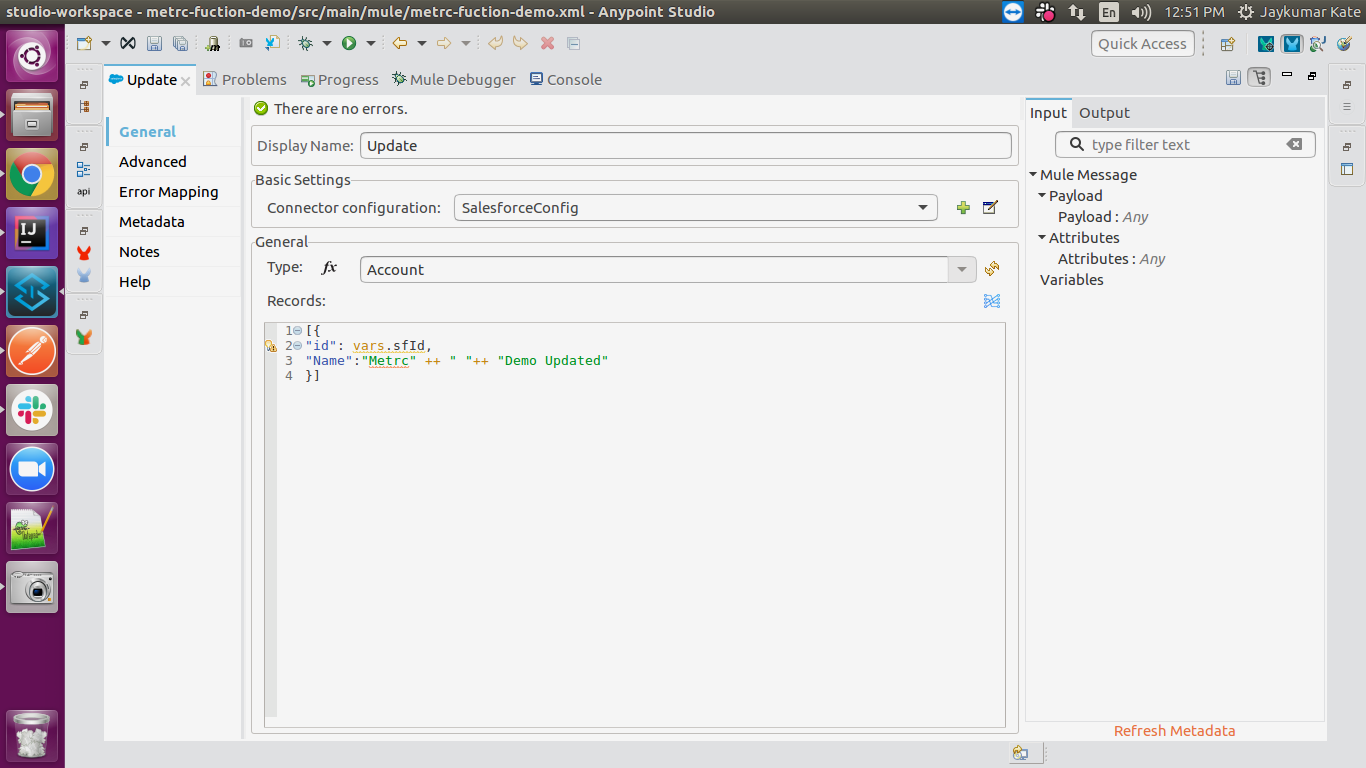
Storing receipt id in Object store:



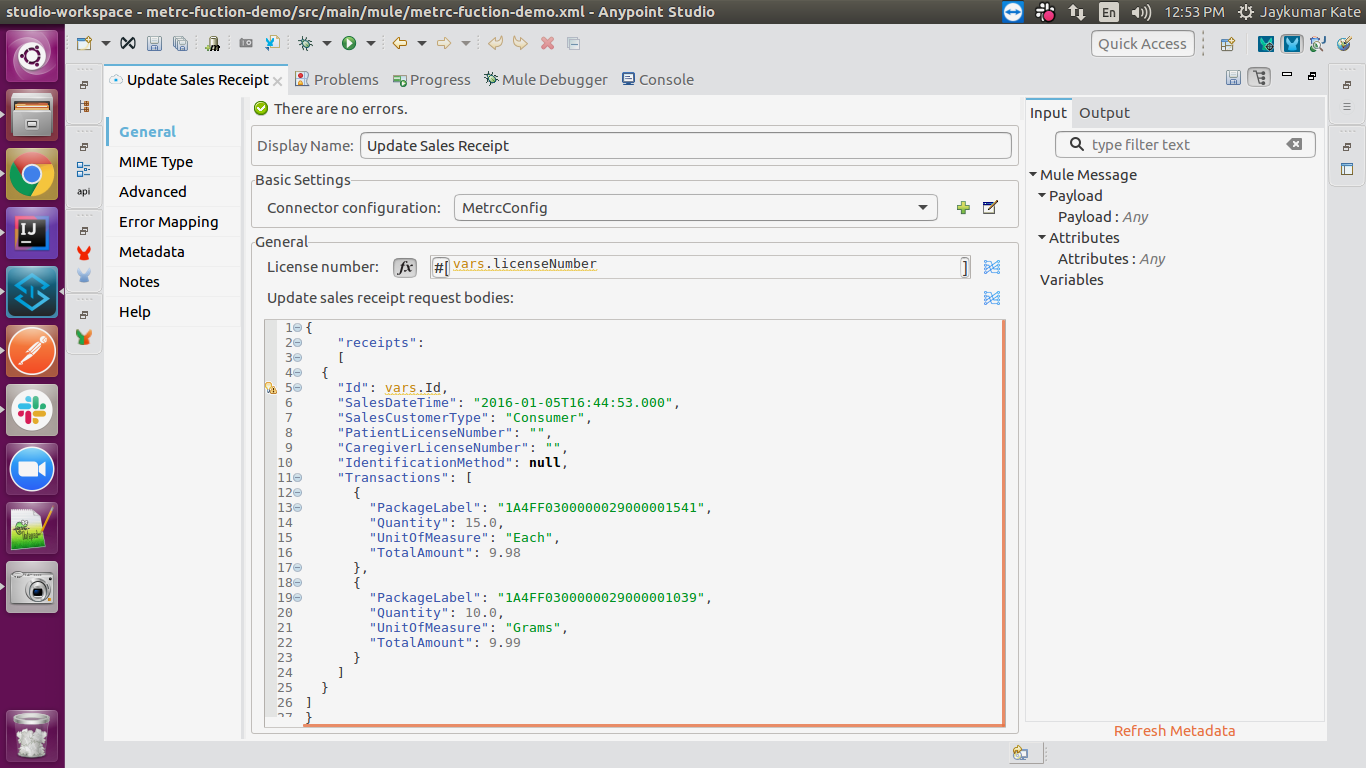
Retrieve receipt id from object store:



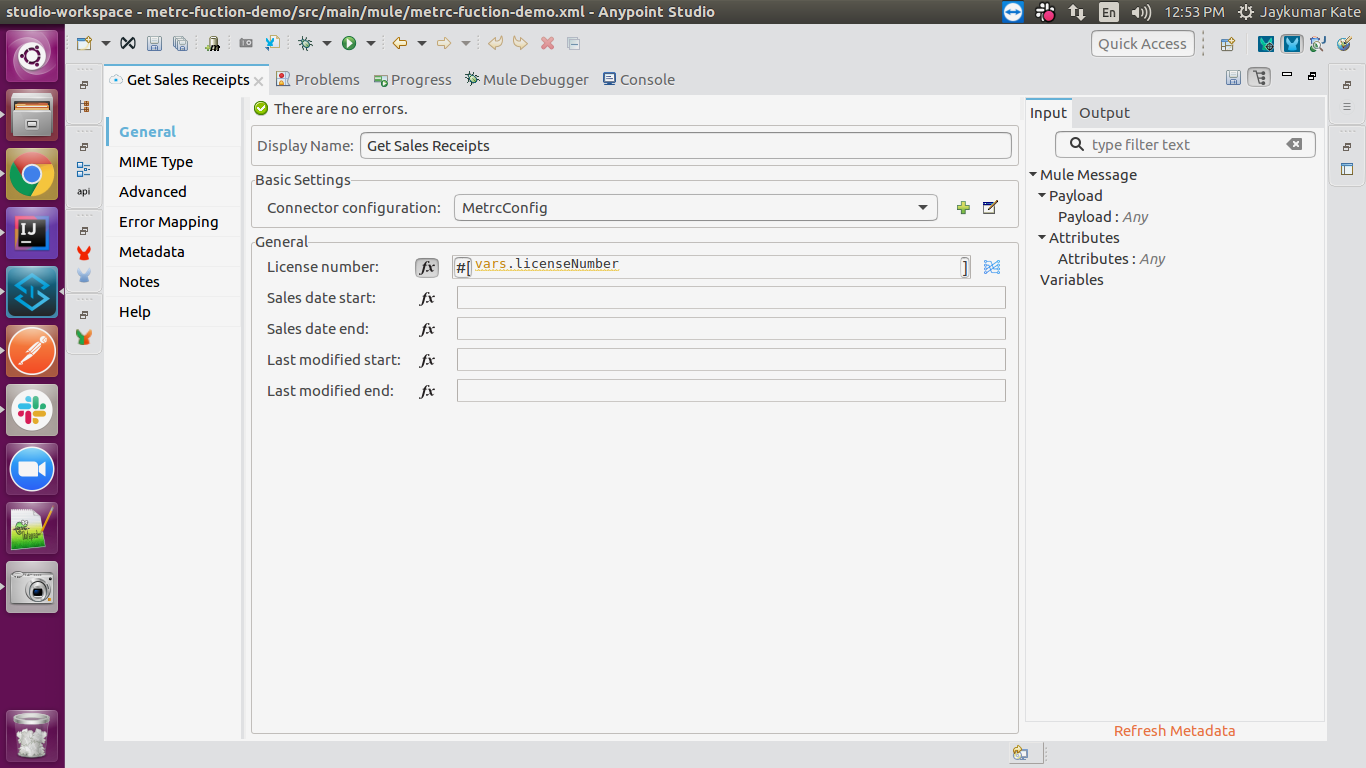
Update salesForce account:



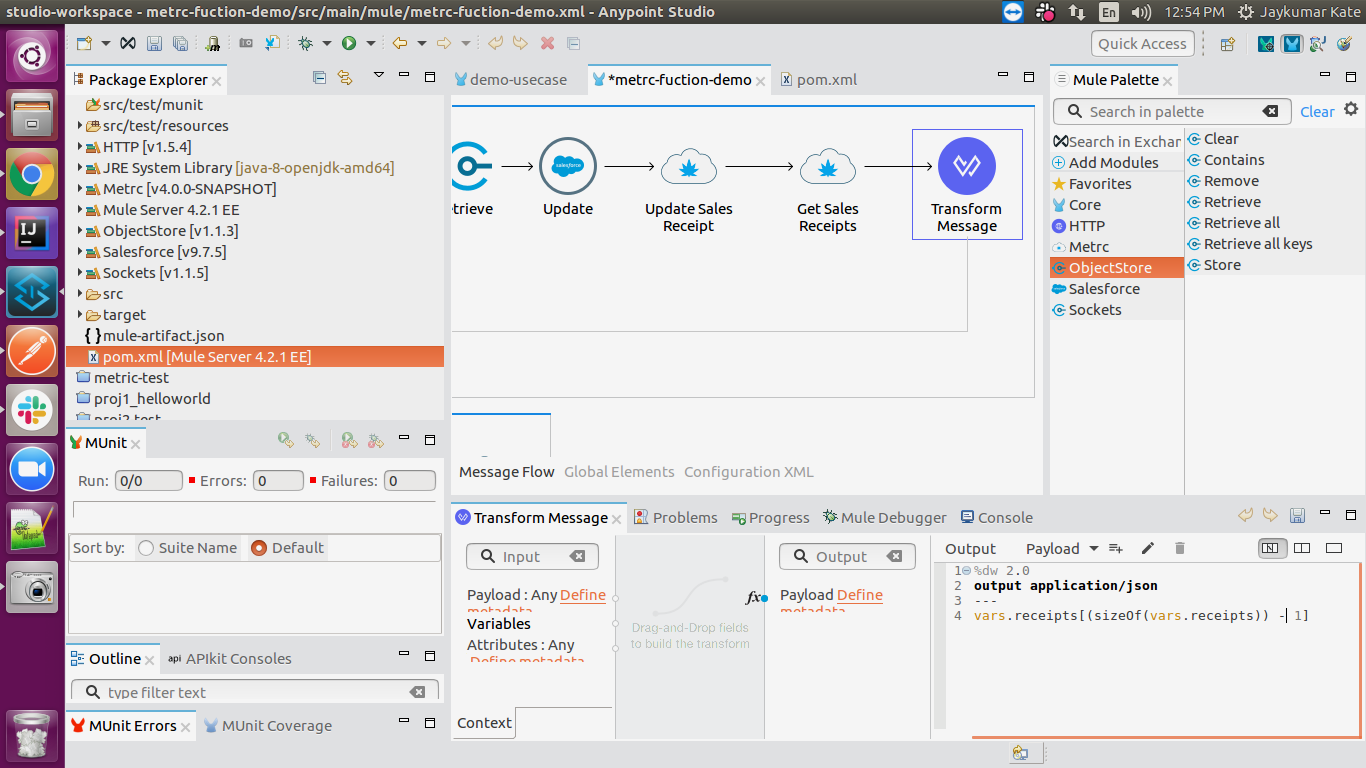
Updating sales receipt:



Get all sales receipts:



Retrieve updated sales receipt:

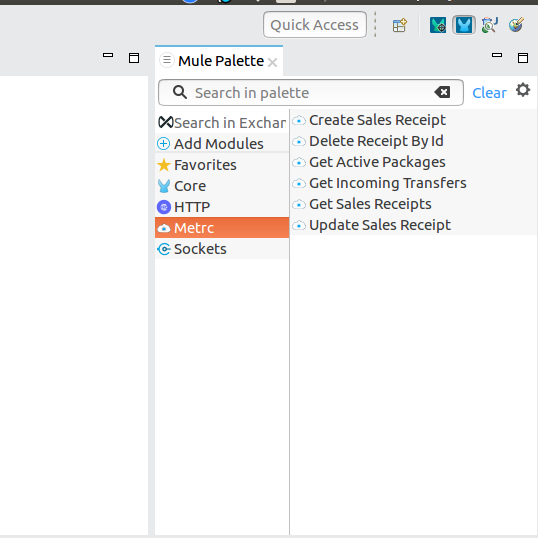


# APPENDIX A

# Install Metrc Connector in Anypoint Studio

Developers can add the Metrc connector module in Anypoint Studio, by following the steps below:

* Open you Anypoint Studio
* Search for the Metrc Connector in your mule palette



# Configure Metrc Connector In anypoint studio

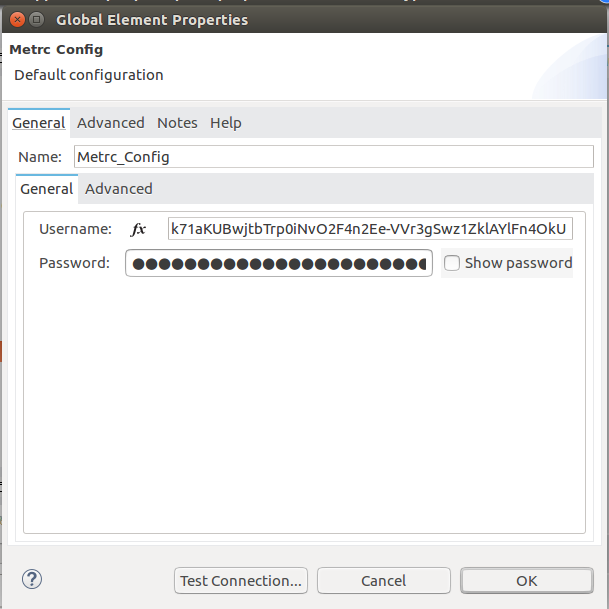
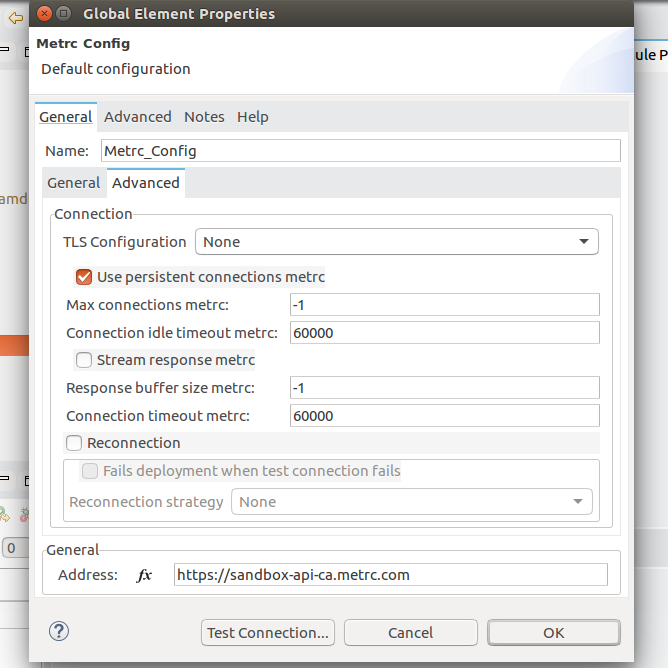
You will require the credentials for creating your access and refresh token, which will be required for the connector.

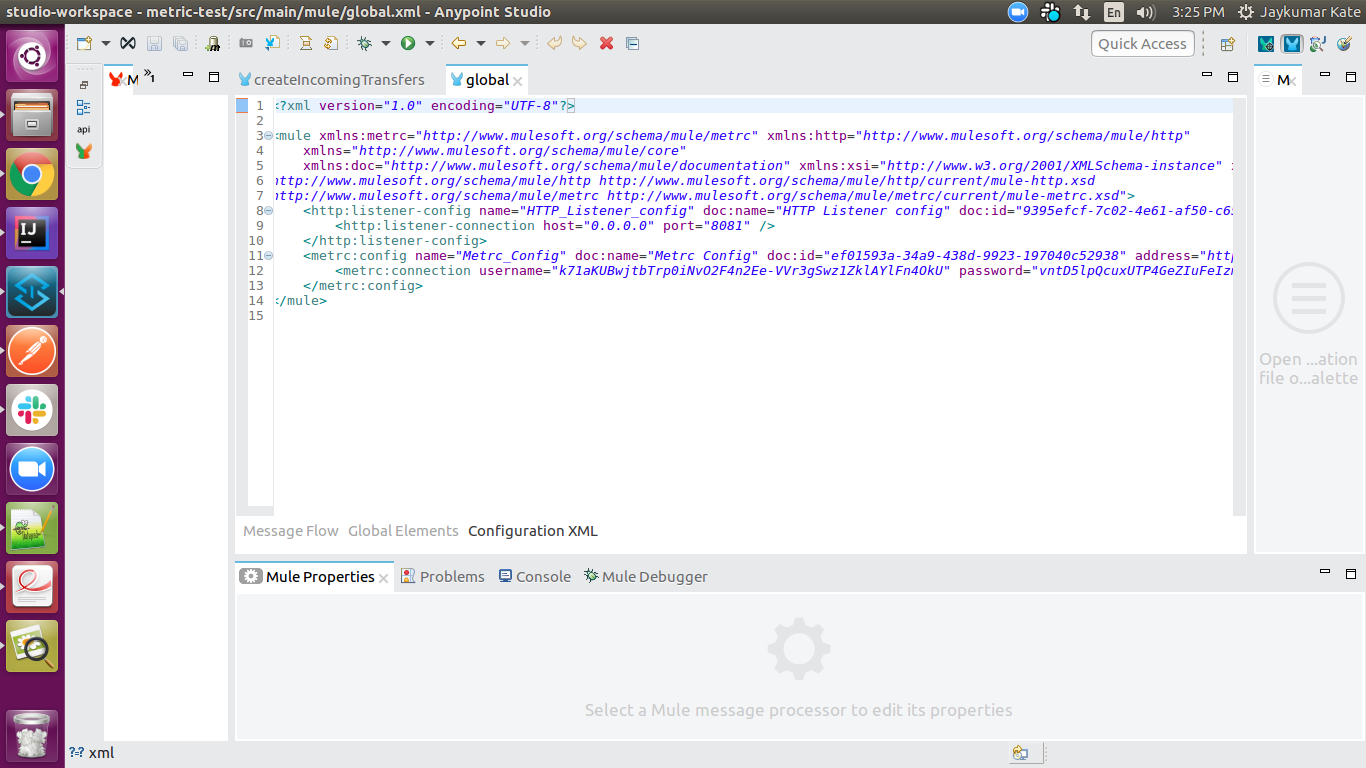
## **Authentication**

Provide following credentials in configuration file

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Address** | URL to access Metrc API |
| **Username** | Your vendor-key to access your Metrc API. |
| **Password** | Your user-key to access your Metrc API. |

## **Configure in Anypoint Studio**

* 1. First thing we need to do to configure the connector, provide above credentials 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 Metrc configuration.
  4. Provide username(vendor-key) & password(user-key). 
  5. Provide Metrc API address by clicking on Advanced tab inside General tab to access Metrc API.
  6. Once these fields are added, you can test the connection
  7. 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/metrc](http://www.mulesoft.org/schema/mule/hybris-occ)*

Schema Location: *[http://www.mulesoft.org/schema/mule/metrc/current/mule-metrc.xsd](http://www.mulesoft.org/schema/mule/hybris-occ/current/mule-hybris-occ.xsd)*