

# 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
A. INTRODUCTION.....	4
1. Purpose.....	4
2. Background.....	4
3. METRC API Service.....	4
4. Overview.....	4
B. MAPPING – BUSINESS FUNCTIONS CALLS to TECHNICAL OPERATIONS.....	5
1. List of Operations Supported by the Metrc Connector:.....	6
C. CONFIGURATION FOR METRC CONNECTOR.....	7
1. Authentication.....	7
D. TO CONNECT IN DESIGN CENTER.....	7
E. Use cases – Connectivity with salesforce.....	11
APPENDIX A.....	20
Install Metrc Connector in Anypoint Studio.....	20
Configure Metrc Connector In anypoint studio.....	21
1. Authentication.....	21
2. Configure in Anypoint Studio.....	21
3. About Connector Namespace and Schema.....	23

## A. INTRODUCTION

### 1. 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.

### 2. 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.

### 3. 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.

### 4. 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.

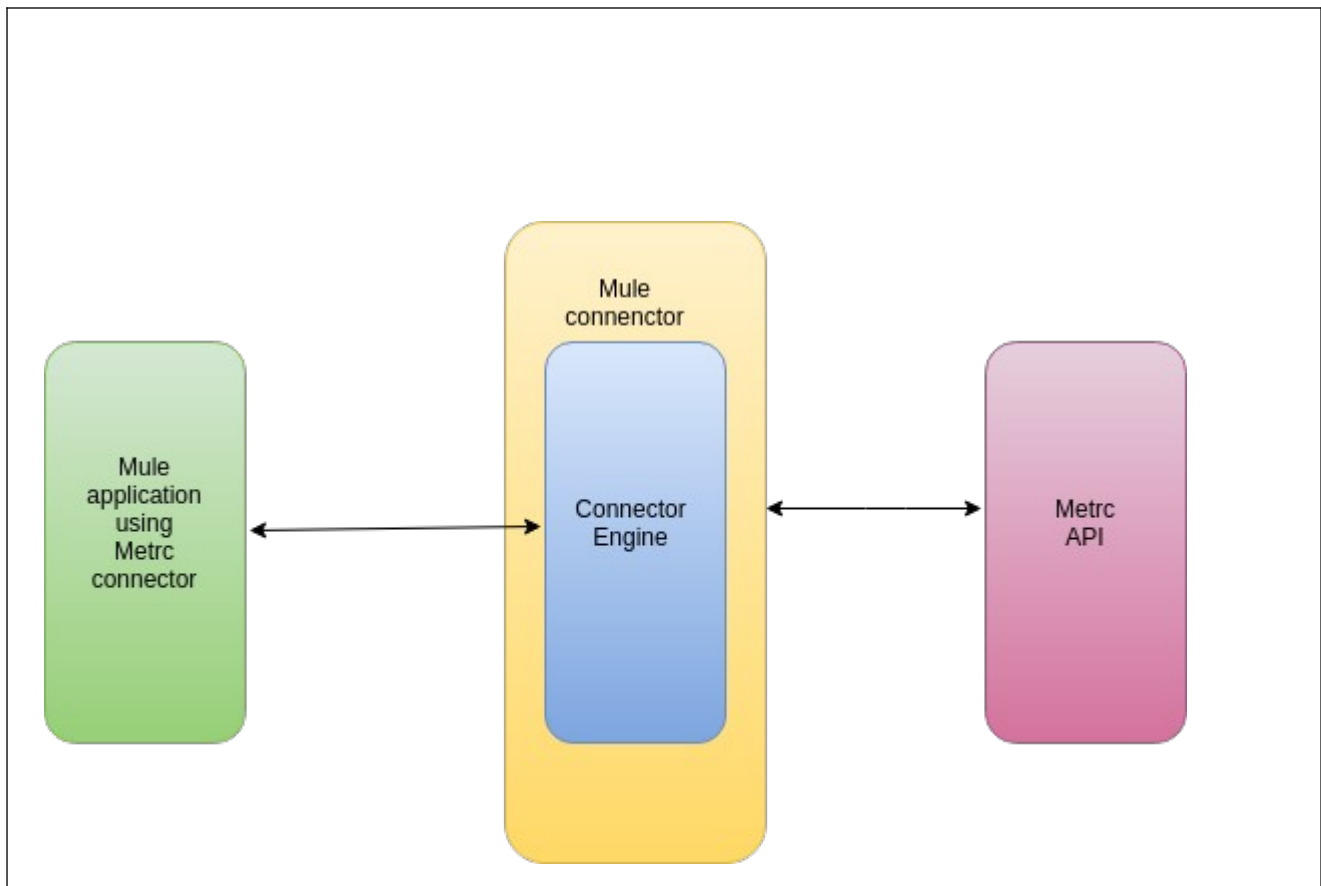


Figure 1: Mule Metrc Connector Integration Architecture

## B. 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.

**1. LIST OF OPERATIONS SUPPORTED BY THE METRC CONNECTOR:**

Sr.No	Business Function	Operations	Description	Inbound	Outbound
<b>Packages</b>					
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
<b>Transfers</b>					
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	POST Request	Creates a transfer	Required field is	No response

	Template		template	licenseNumber.	
<b>Sales</b>					
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

## C. CONFIGURATION FOR METRC CONNECTOR

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

### 1. AUTHENTICATION

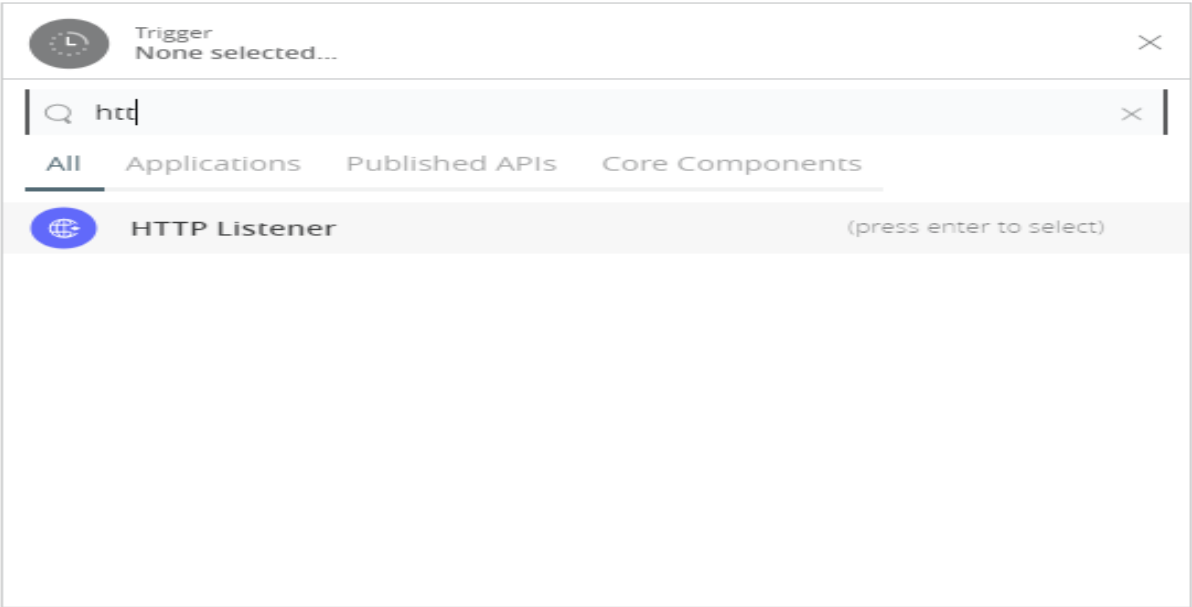
The Metrc API uses Basic Auth to provide access to an API. Vendor-key must be used as username & user-key must be used as password.

1. **Metrc connection:** This connection is used to provide an access to Metrc API. It requires below credentials:
  - username: <vendor-key>
  - password: <user-key>

## D. 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.



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

HTTP Listener Configuration

General

TLS

Advanced

Configuration Name (required) ⓘ

HTTP Listener

Connection

Protocol ⓘ

HTTP

Host (required) ⓘ

0.0.0.0

Port (required) ⓘ

8081

Base Path ⓘ

Listener Interceptors

Cors Interceptor (required)

☐ Allow Credentials

Origins (required)

Add

Cancel

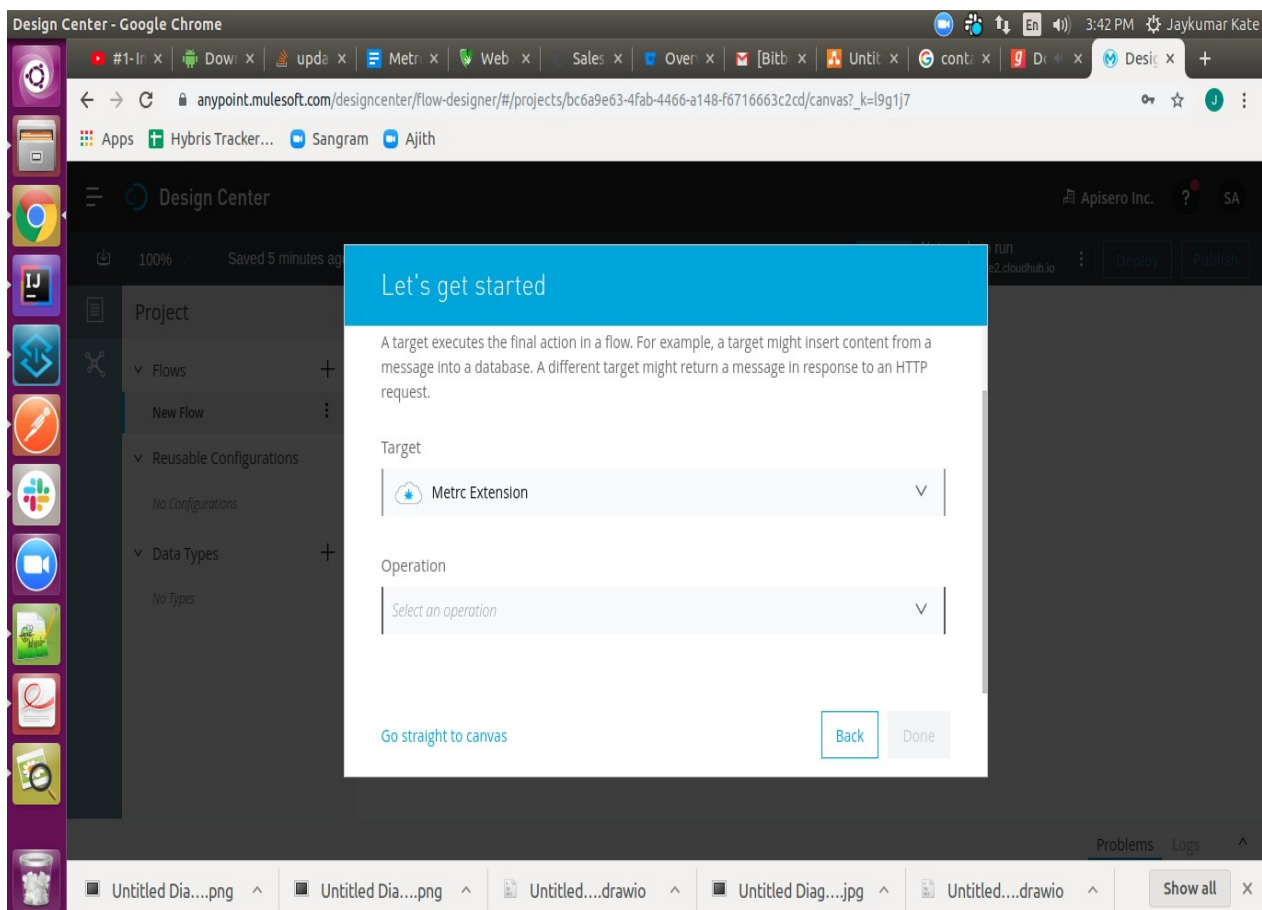
Save

Field	Description
-------	-------------

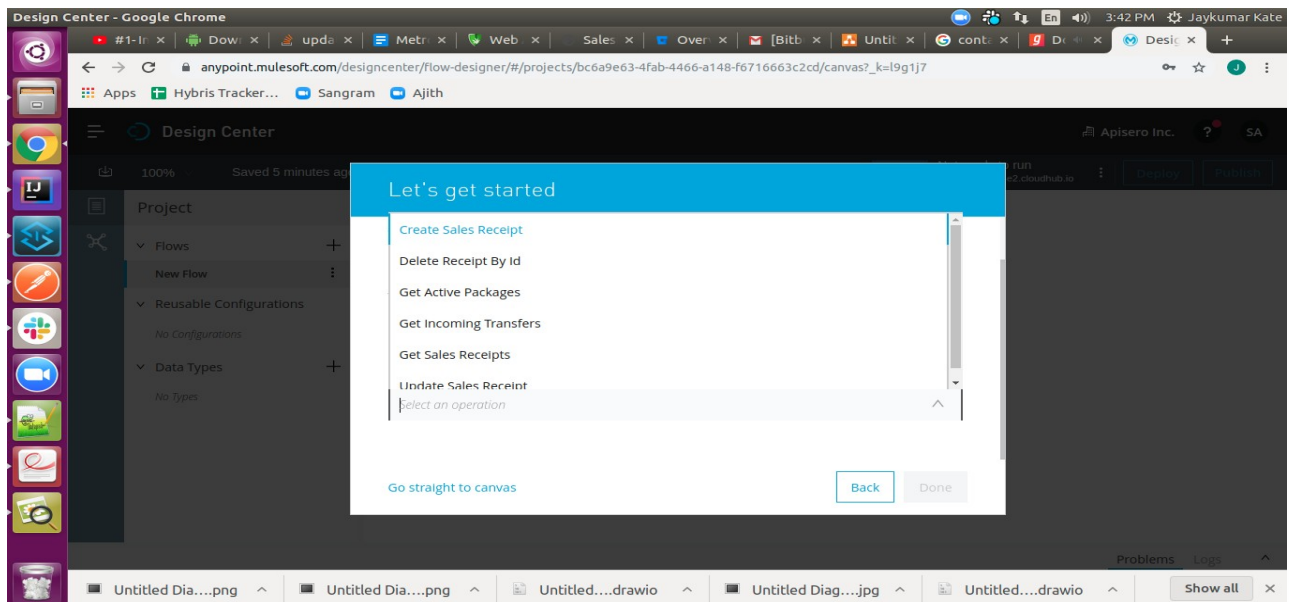


<b>Protocol</b>	Protocol selected for the HTTP connector, it can be HTTP or HTTPS (secure).
<b>Host</b>	IP address where your Mule application listens for requests.
<b>Port</b>	Port address where your Mule application listens for requests.
<b>Base Path</b>	Path where your Mule application listens for requests.

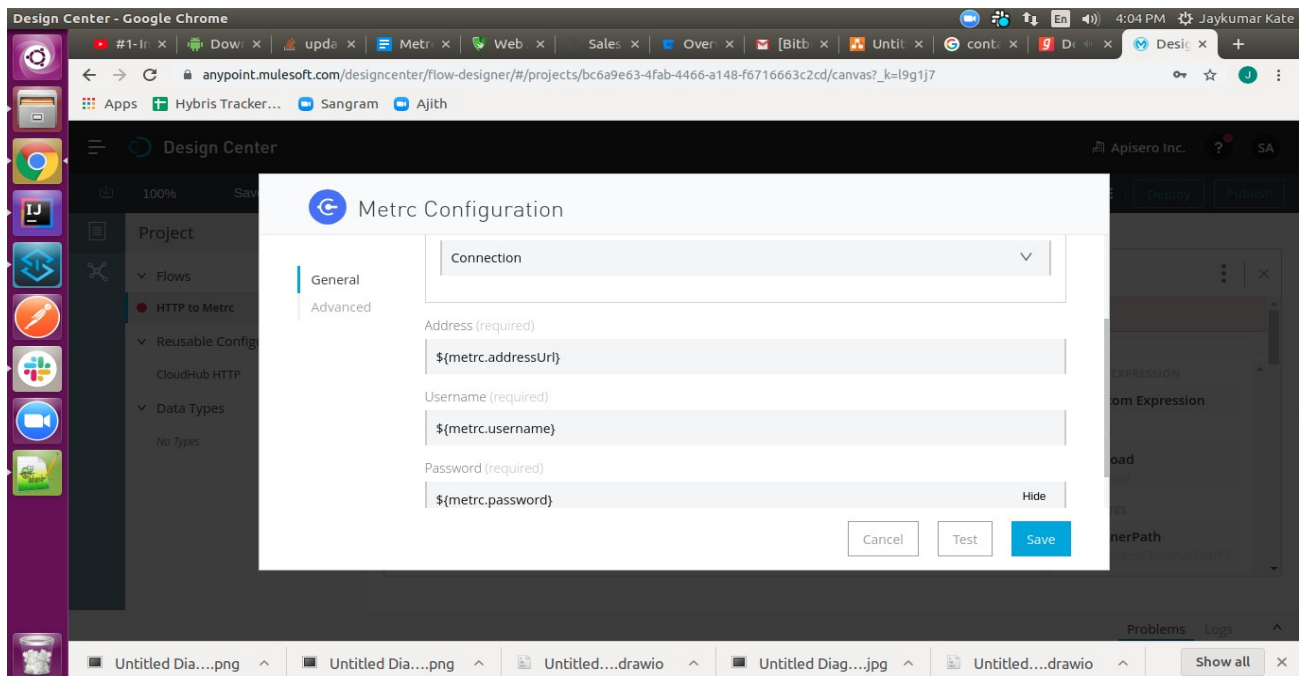
4. Select the plus sign to add a component.
5. Select the Metrc connector as a component.



## 6. Select an operation:



## 7. Configure the Global element for the connector:



Field	Description
<b>Address</b>	URL to access Metrc API
<b>Username</b>	Your vendor-key to access your Metrc API.
<b>Password</b>	Your user-key to access your Metrc API.

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

## E. 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,

**Global Element Properties**

**HTTP Listener config**  
Configuration element for a HttpListener.

General Notes Help

Name: HTTP\_Listener\_config

General TLS Advanced

Connection

Protocol: HTTP (Default)

Host: All Interfaces [0.0.0.0] (default)

Port: \${http.port}

General

Base path:

Listener interceptors: None

Test Connection... OK Cancel

4. Test the connection and click on Okay.
5. 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.
6. Now add configurations for Metrc.
7. Go to global-configurations.xml global elements -> create -> Connector Configuration -> Metrc Configuration
8. Add following properties:  
Address Url :  
Username :  
Password :

**Global Element Properties**

**Metrc Config**

Default configuration


General Advanced Notes Help

Name: Metrc\_Config

General Advanced

Username: *fx* k71aKUBwjtTrp0iNvO2F4n2Ee-VVr3gSwz1ZklAYlFn4OkU

Password:  ☐ Show password

 Test Connection... Cancel OK

The screenshot shows the 'Global Element Properties' dialog box for a configuration named 'Metrc Config'. The dialog has a title bar with standard window controls and a subtitle 'Default configuration'. It features a tabbed interface with 'General', 'Advanced', 'Notes', and 'Help' tabs. The 'General' tab is active, showing the configuration details. The 'Name' field is set to 'Metrc\_Config'. Below this, there are sub-tabs for 'General' and 'Advanced'. The 'Connection' section is expanded, showing various settings: 'TLS Configuration' is set to 'None'; 'Use persistent connections metrc' is checked; 'Max connections metrc' is set to '-1'; 'Connection idle timeout metrc' is set to '60000'; 'Stream response metrc' is unchecked; 'Response buffer size metrc' is set to '-1'; 'Connection timeout metrc' is set to '60000'; 'Reconnection' is unchecked; 'Fails deployment when test connection fails' is unchecked; and 'Reconnection strategy' is set to 'None'. At the bottom, the 'General' section shows the 'Address' field with a value of 'https://sandbox-api-ca.metrc.com'. The dialog concludes with a help icon, a 'Test Connection...' button, and 'Cancel' and 'OK' buttons.

**Global Element Properties**

**Metrc Config**  
Default configuration

General Advanced Notes Help

Name: Metrc\_Config

General Advanced

Connection

TLS Configuration None

☒ Use persistent connections metrc

Max connections metrc: -1

Connection idle timeout metrc: 60000

☐ Stream response metrc

Response buffer size metrc: -1

Connection timeout metrc: 60000

☐ Reconnection

☐ Fails deployment when test connection fails

Reconnection strategy None

General

Address: *fx* https://sandbox-api-ca.metrc.com

? Test Connection... Cancel OK

9. Add Salesforce configuration.
10. Go to global-configurations.xml -> global elements -> create -> Connector Configuration -> Salesforce Configuration

## 11. Add following properties

General Apex Advanced Notes Help

Name: Salesforce\_Config

Connection: Username Password (Deprecated)

General Advanced

Proxy

Host: fx

Port: fx

Username: fx

Password: Show pas

Connection

Username: fx \$(sfdc.username)

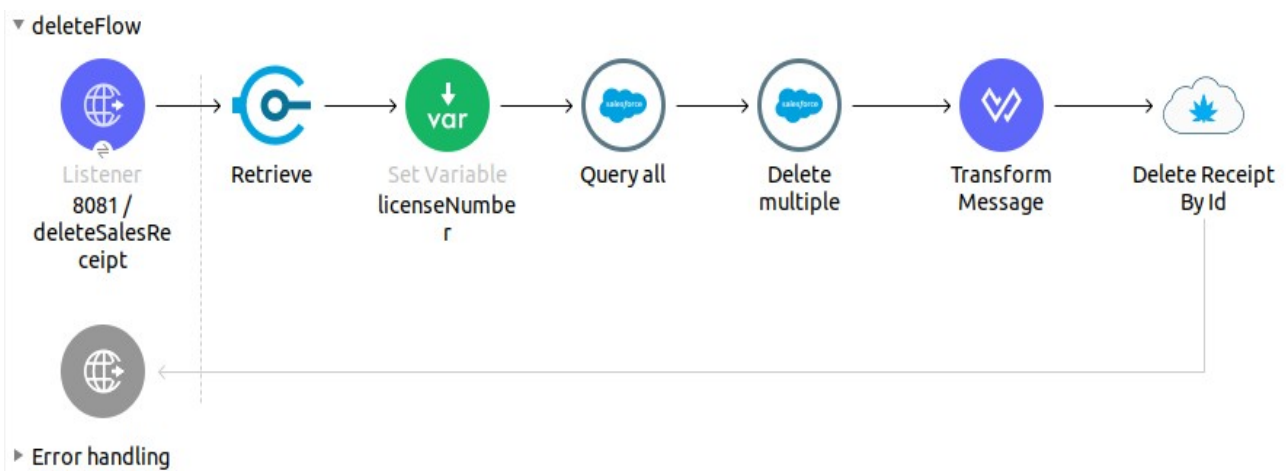
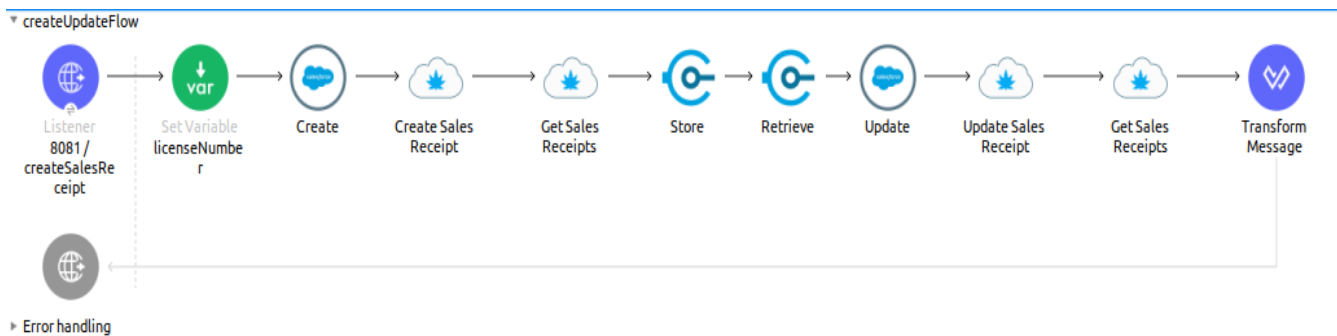
Password: \$(sfdc.password) Show pas

Security token: fx \$(sfdc.securityToken)

Authorization URL: fx \$(sfdc.url)

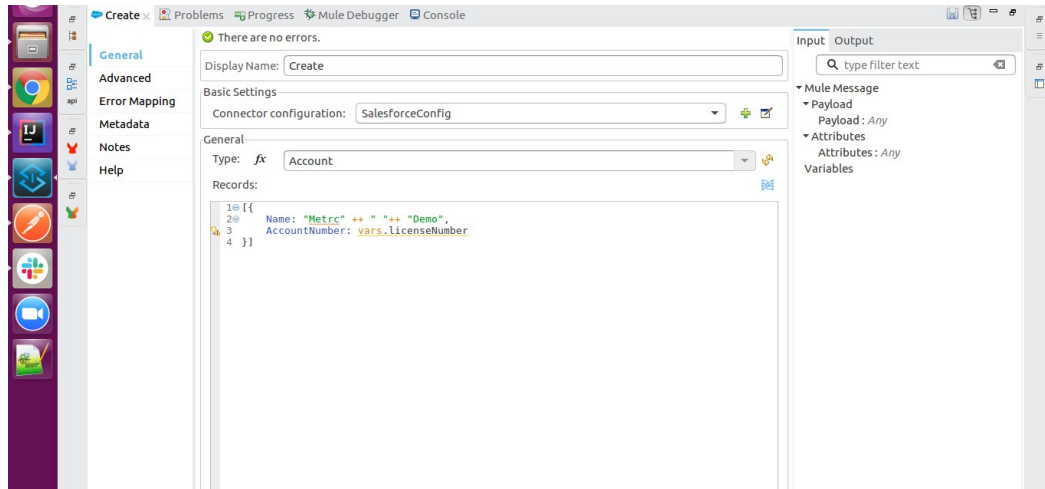
Test Connection... OK Cancel

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

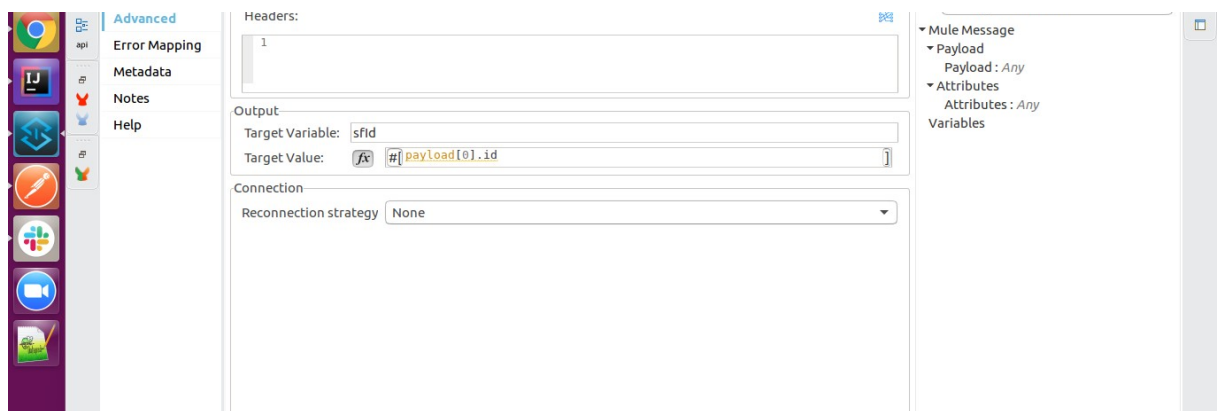


13. Individual mappings for each component are illustrated in below screenshots:

Create Salesforce account request body:

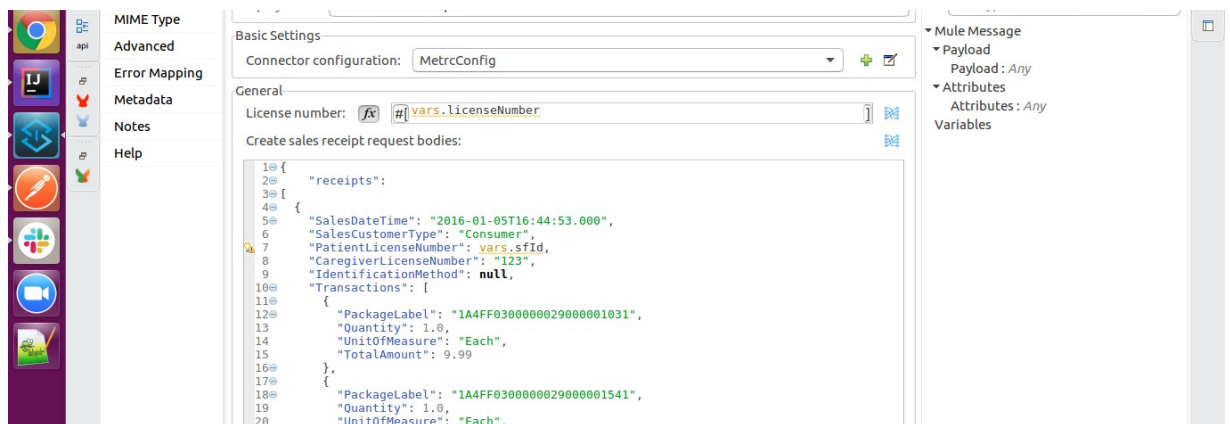


Store Salesforce account id:

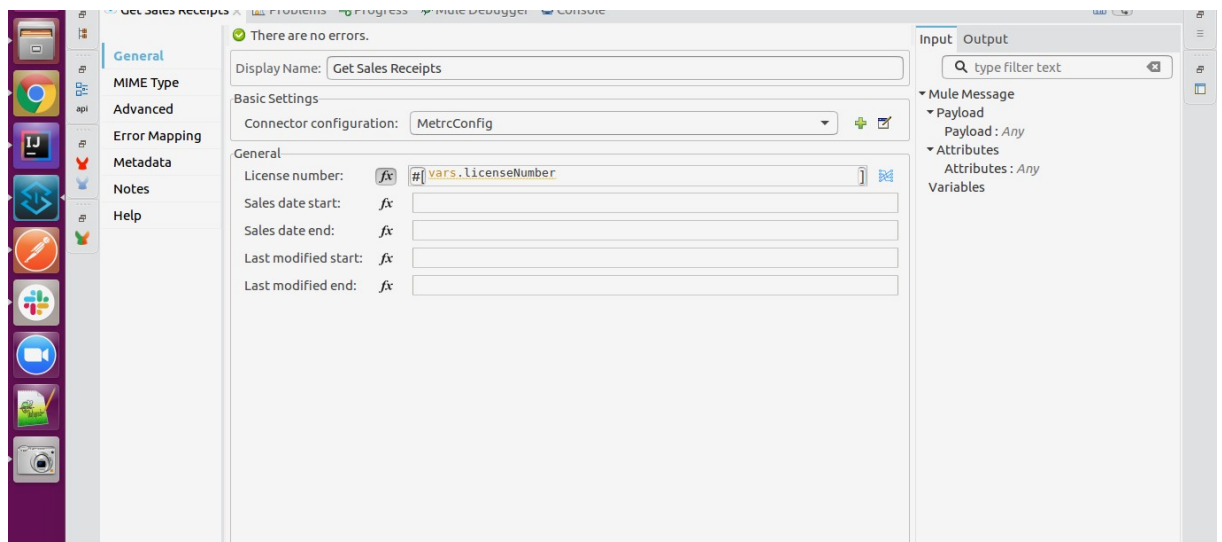




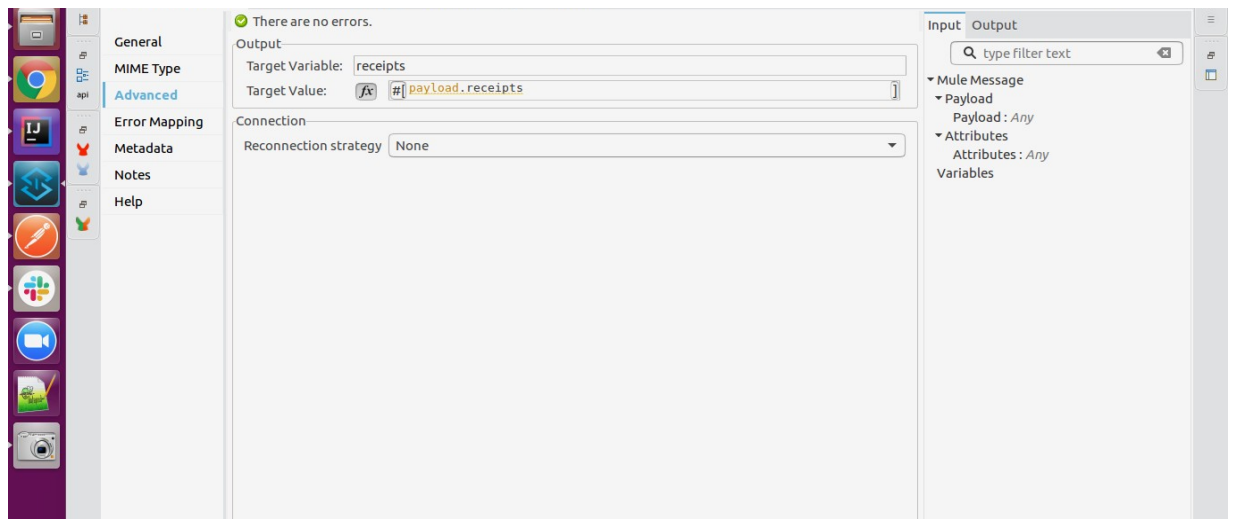
### Creating sales receipt:



### Get all sales receipts :



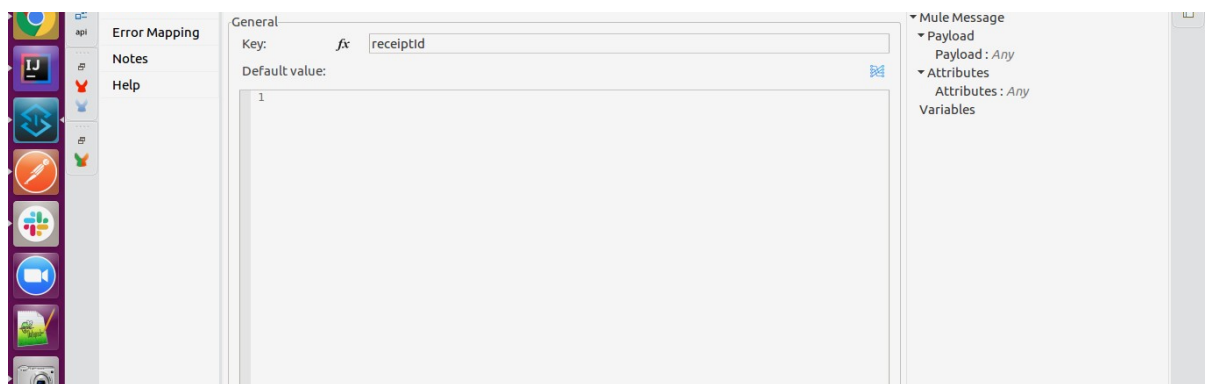
store sales receipts in variable:



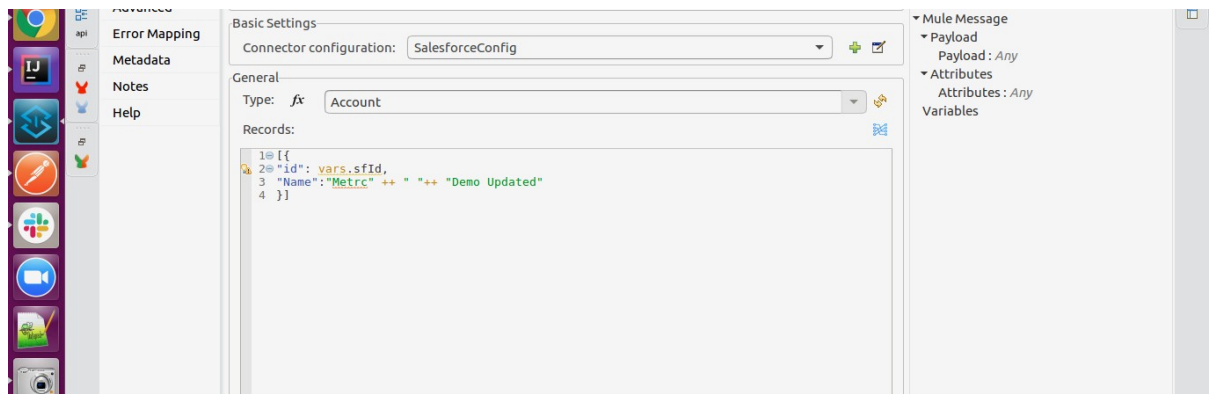
Storing receipt id in Object store:



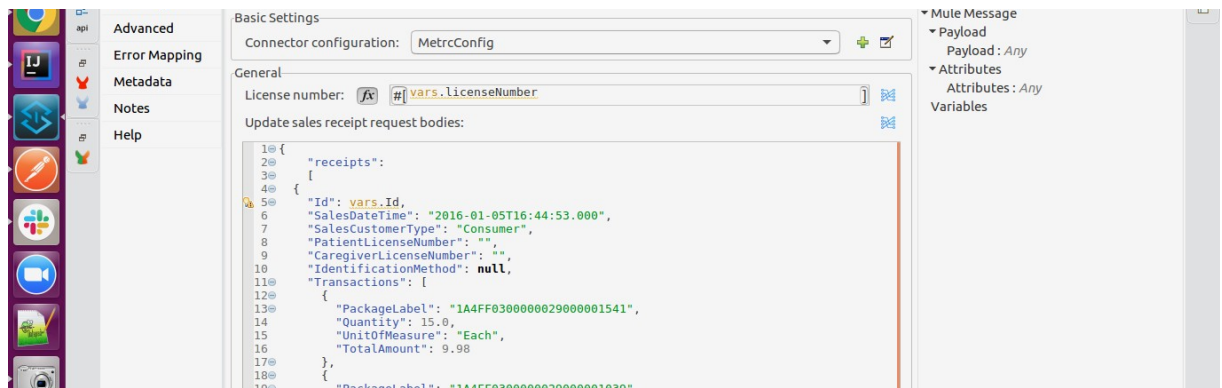
Retrieve receipt id from object store:



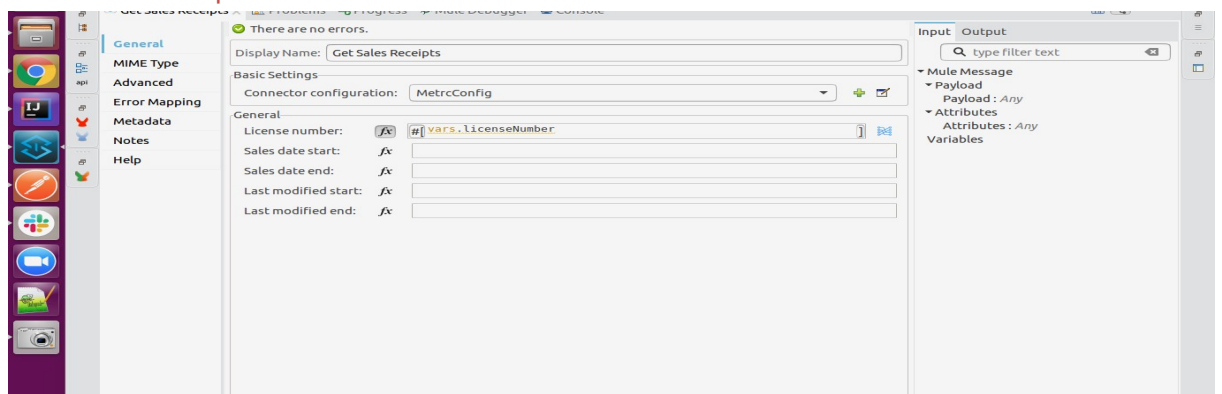
## Update salesForce account:



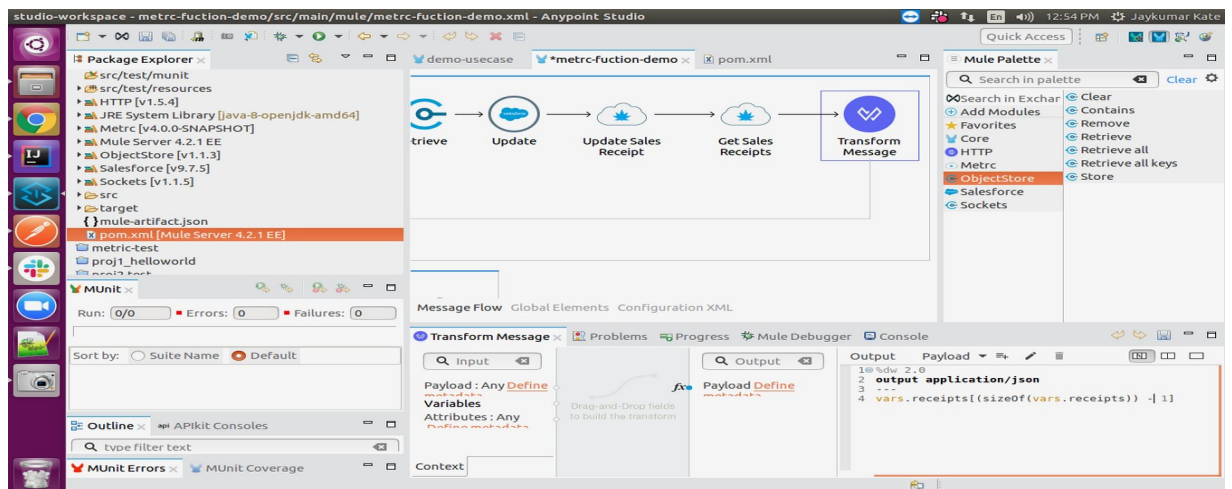
## Updating sales receipt:



## Get all sales receipts:



## Retrieve updated sales receipt:

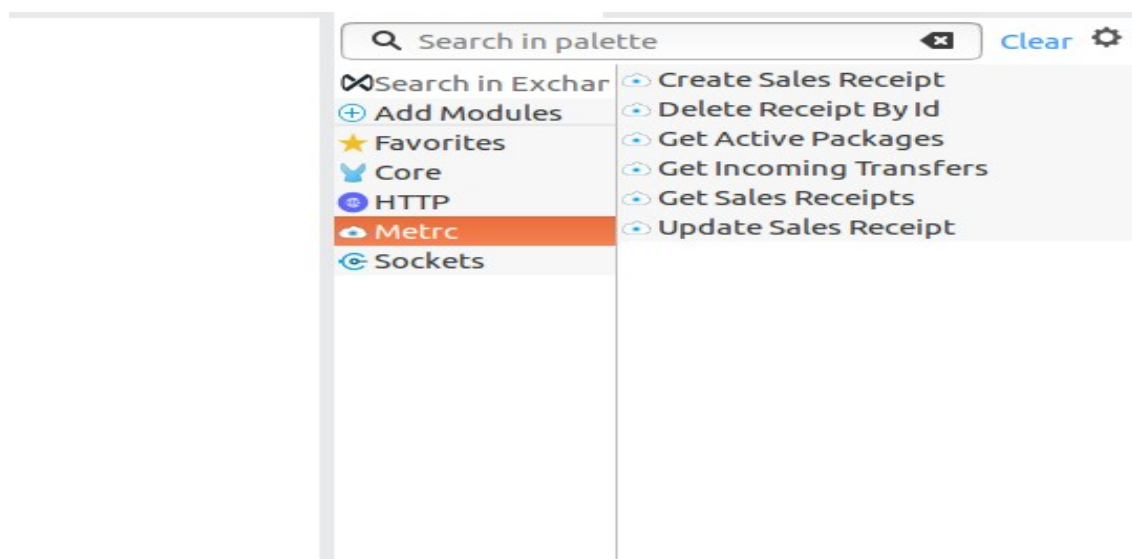


## APPENDIX A

### INSTALL METRC CONNECTOR IN ANYPPOINT 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.

### 1. AUTHENTICATION

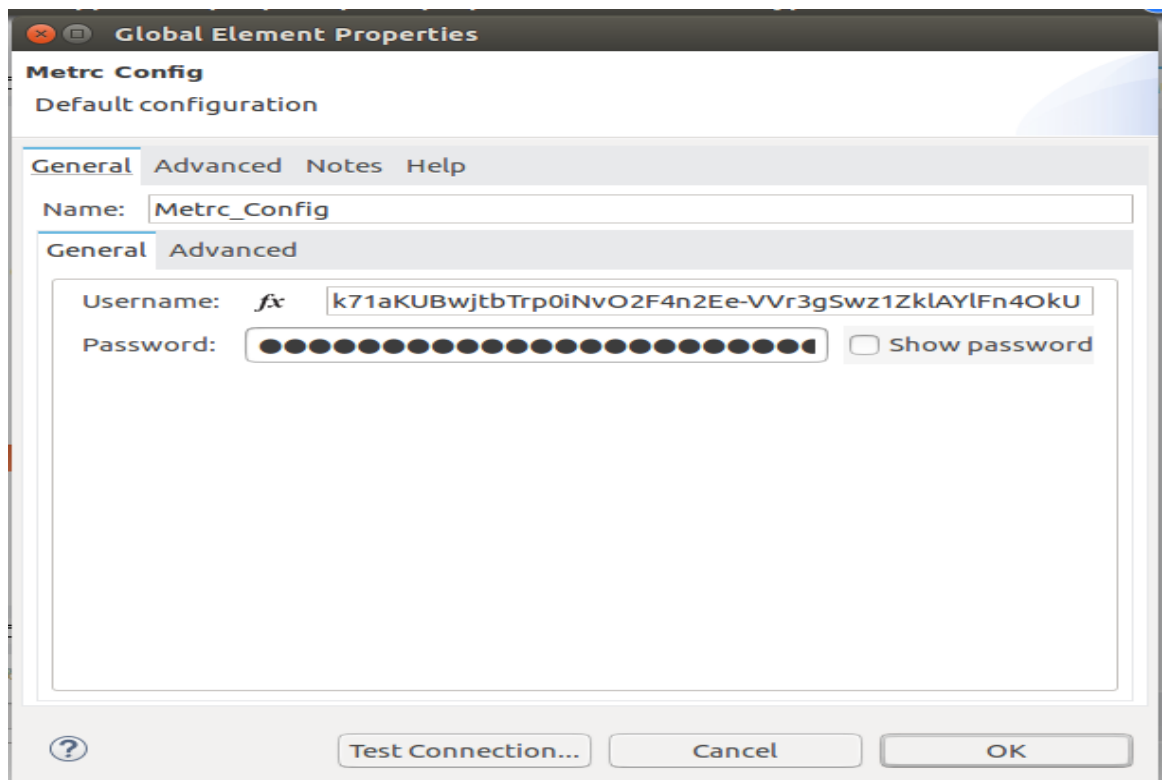
Provide following credentials in configuration file

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.

### 2. CONFIGURE IN ANYPOINT STUDIO

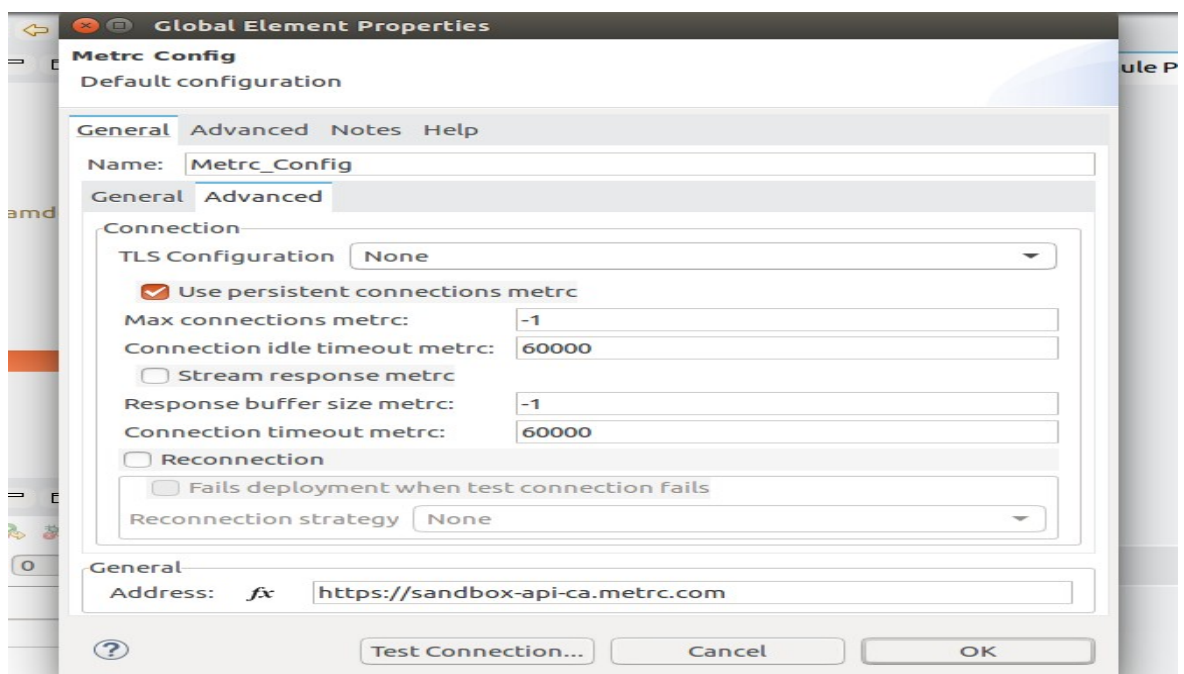
- a. First thing we need to do to configure the connector, provide above credentials required to access the APIs, as mentioned in this document previously
- b. Once you have your credentials, drag and drop your connector in the mule pallet and make sure you have a listener in the canvas.
- c. In the Global elements, create a new Metrc configuration.

- d. Provide username(vendor-key) & password(user-key).



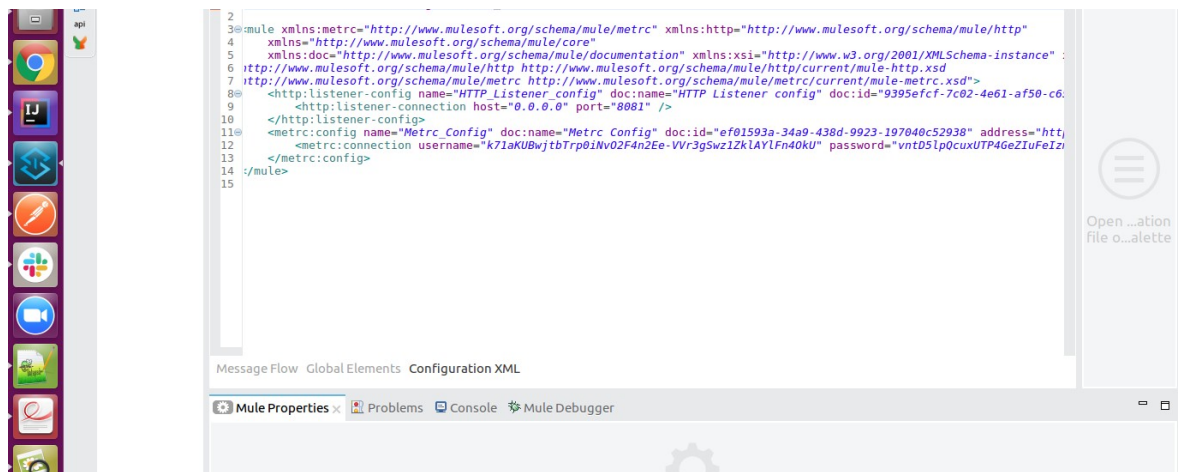
The screenshot shows the 'Global Element Properties' dialog box for 'Metrc Config'. The 'General' tab is selected. The 'Name' field is 'Metrc\_Config'. The 'General' sub-tab is also selected. The 'Username' field contains 'k71aKUBwjtTrp0iNvO2F4n2Ee-VVr3gSwz1ZklAYlFn4OkU'. The 'Password' field is masked with dots. A 'Show password' checkbox is present. At the bottom, there are buttons for 'Test Connection...', 'Cancel', and 'OK'.

- e. Provide Metrc API address by clicking on Advanced tab inside General tab to access Metrc API.



The screenshot shows the 'Global Element Properties' dialog box for 'Metrc Config'. The 'General' tab is selected, and the 'Advanced' sub-tab is also selected. The 'Name' field is 'Metrc\_Config'. The 'Connection' section is expanded, showing 'TLS Configuration' set to 'None'. The 'Use persistent connections metrc' checkbox is checked. The 'Max connections metrc' field is '-1'. The 'Connection idle timeout metrc' field is '60000'. The 'Stream response metrc' checkbox is unchecked. The 'Response buffer size metrc' field is '-1'. The 'Connection timeout metrc' field is '60000'. The 'Reconnection' section is expanded, showing 'Fails deployment when test connection fails' checkbox is unchecked. The 'Reconnection strategy' dropdown is set to 'None'. At the bottom, there are buttons for 'Test Connection...', 'Cancel', and 'OK'.

- f. Once these fields are added, you can test the connection
- g. 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:



### 3. 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/metric>

Schema Location: <http://www.mulesoft.org/schema/mule/metric/current/mule-metric.xsd>