# Lab 7: Connect to a System API

[Lab 6: Connect to a REST Endpoint](#_86maahp7gk5r)

[Overview](#_hmlhk16nx5wk)

[Step 1: Configure the RAML Consumer](#_pxqjtxd84ch4)

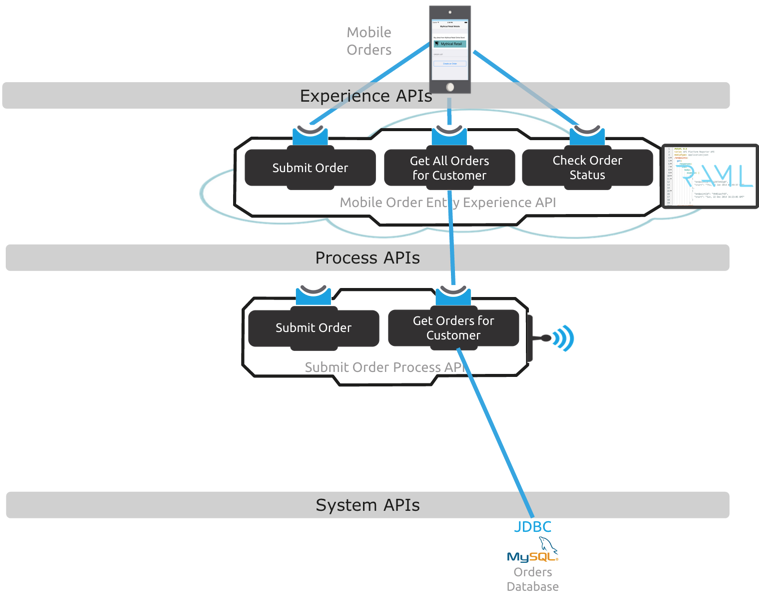
[Step 2: Create DataMapper transformations](#_8avcbmto54bc)

[Step 3: Run the API](#_subkunlz4qxi)

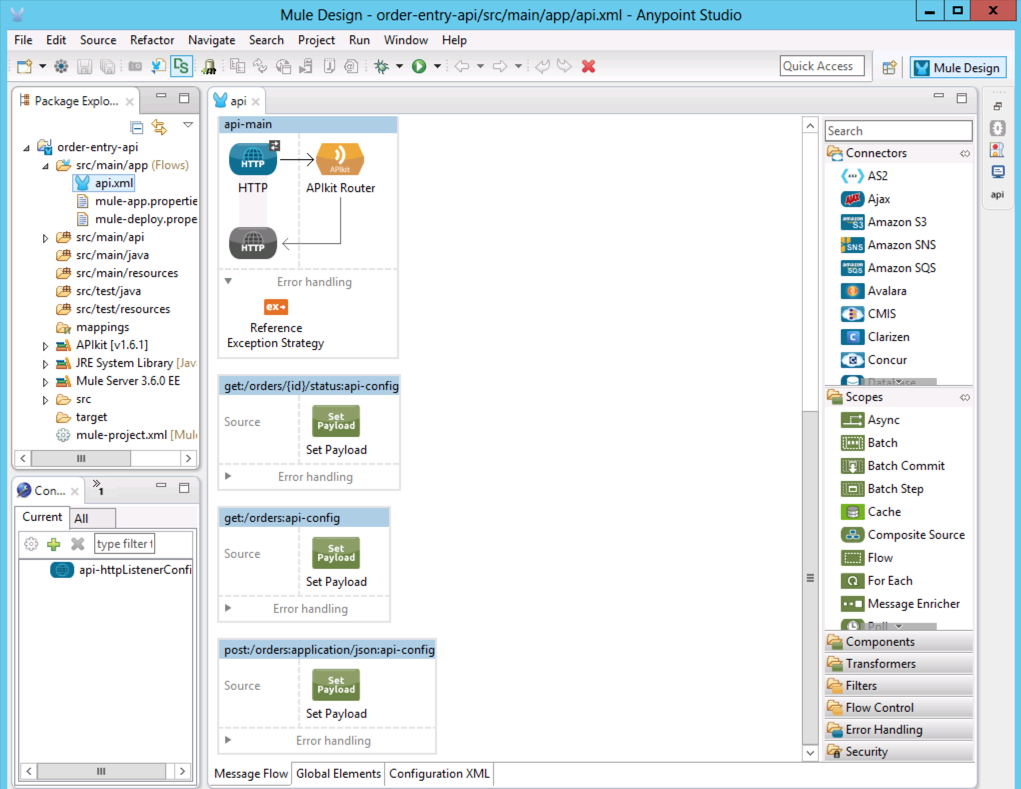
[Summary](#_knuzle5uc8kd)

## Overview

Your mobile developers have been busy testing their apps against the mock service and skeleton API. Now it’s time to do the real implementation. Remember your out-of-the-box API implementation was returning static data, now you need to call the back-end Order Processing systems to get the real data.

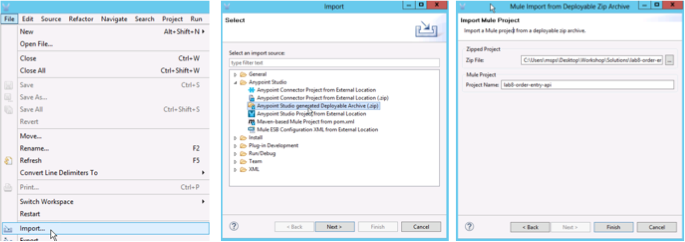


For the purpose of this workshop, we have a pre-built **Internal Sales Order API** that retrieves and processes orders. This application is already deployed on the Mule run-time within your workshop image. At this point, we should already have a generated version of our API project created in Anypoint Studio like shown in the following screen.



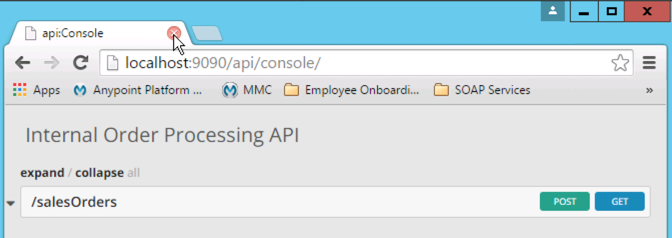
In this lab, we will start implementing the Order Entry API’s **get:/orders** method we have defined in the previous lab, particularly we will retrieve the list of orders from the the **Internal Sales Order API**. Since this is a **RAML** based API we will use the **RAML** consumer component.

|  |
| --- |
| **NOTE*:*** If you wish to skip this lab or need to catch up, you can open the completed solution by importing the following project:  **C:\Users\msps\Desktop\Workshop\Solutions\lab7-order-entry-api.zip** |



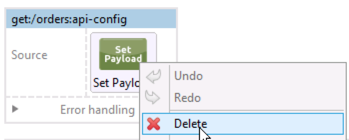
## Step 1: Configure the RAML Consumer

Let’s start implementing the **get:/orders** back-end flow by calling the **Internal Sales Order API,** which is using the REST API Modelling Language (RAML) for its definition. Because this API is using RAML, we will configure the RAML Consumer in our HTTP endpoint to interact with it. The **Internal Sales Order API** is accessible from its API Console: [**http://localhost:9090/api/console**](http://localhost:9090/api/console)**.**

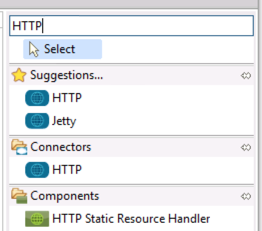


1. Preview the **Internal Sales Order API**  by taking a look at its API console from a browser using the URL: [**http://localhost:9090/api/console**](http://localhost:9090/api/console)

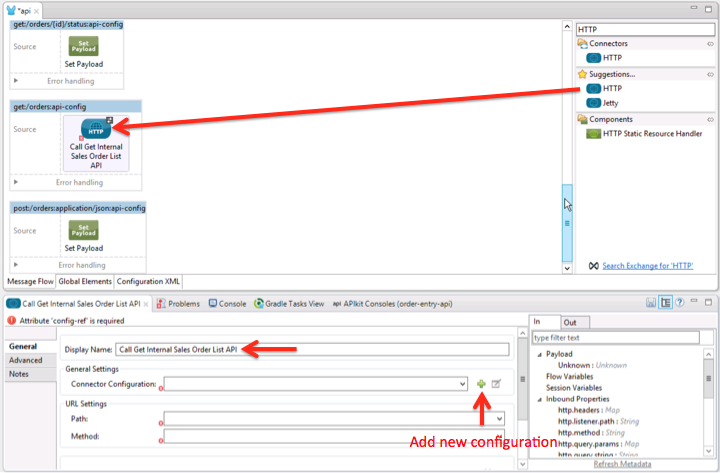
|  |
| --- |
| **NOTE**: As you progress through the lab make sure you are editing the correct flow in each step! |



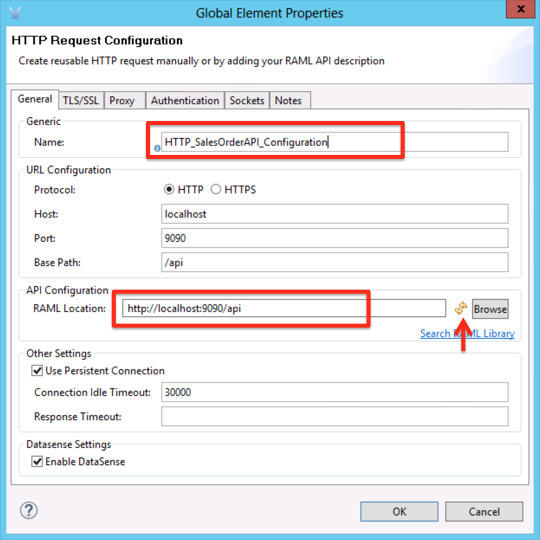
1. Back in the flow diagram in Anypoint Studio, delete the Set Payload processor from the flow named **get:/orders:api-config**.



1. Search for HTTP in the component palette.



1. Click on the **HTTP Connector**
2. Drag it to the flow `
3. Double-click on it to display the properties. You should see the component properties for you to configure in the **Mule Properties View**.
4. Add a meaningful name for the operation like Call Get Internal Sales Order List API in the **Display Name** field for the HTTP properties.

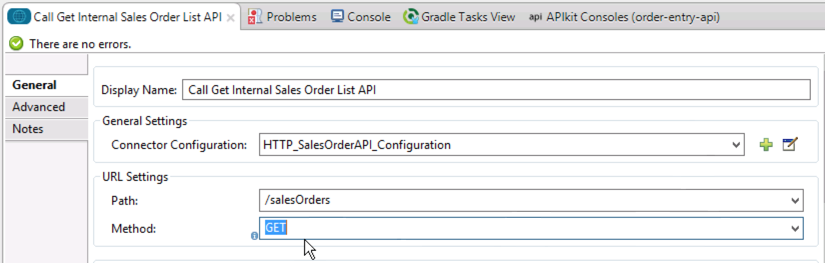


You need to define the HTTP Configuration you will use with this API.

1. Click on the + icon to create a new configuration as shown in the screen.
2. Change the name to: **HTTP\_SalesOrderAPI\_Configuration.**
3. On the **RAML location** text field, enter **the Internal Sales Order API** endpoint: [http://localhost:9090/api](http://internal-salesorder-api-orderentry.cloudhub.io/api)
4. Click on the refresh button.



1. Once you click on the refresh button, the RAML consumer component will read the RAML metadata from the API and will populate the **Host**, **Port** and **Base Path**. The RAML Consumer does this by sending an HTTP request to the endpoint with Header "Accept" set to "application/raml+yaml". This fetches the RAML definition.
2. Click on **OK** to close the window.

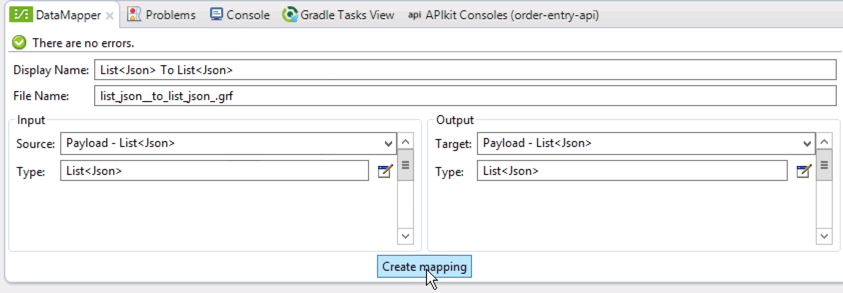


1. Now the **RAML** consumer populate the Path combo box with the resources defined in the API, in the this case: /**salesOrders**.
2. Click on the Method combo box and select the **GET** method as shown.

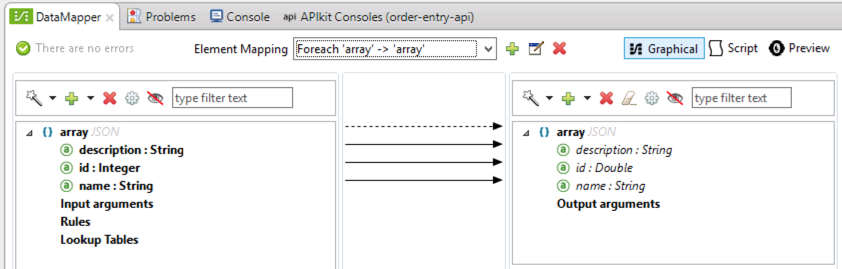
## Step 2: Create DataMapper transformations



1. Search for the DataMapper component in the “Transformers” section of the component palette and drag it to the end of the **get:/orders** flow.

****

1. Click on the Data Mapper and you should see the component properties for you to configure in the **Mule Properties View.** Notice that it detects the JSON input from the Internal Sales Order API, and the JSON Output from your RAML. Click on **Create mapping.**

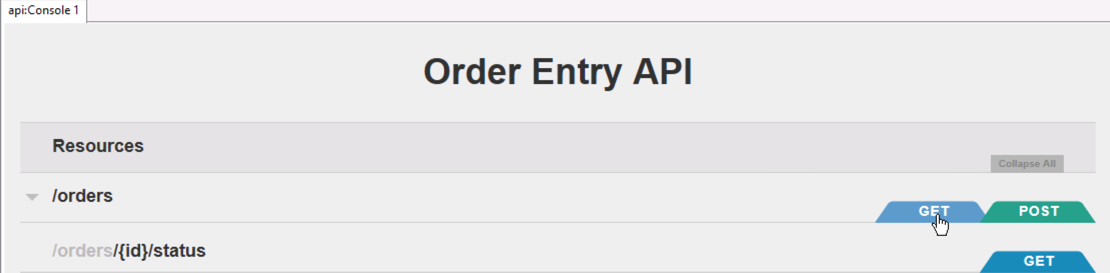


1. Because the JSON structure is the same for the input and the output, the fields will be automatically mapped. (Note that we could have opted not to do a transformation, but here we use Data Mapper to see what the input and output data structures look like and in most cases, they will not be the same).
2. Click **Save All**. 

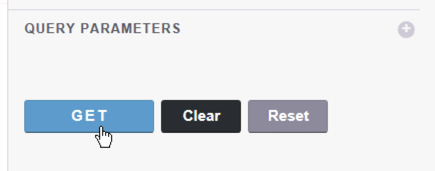
## Step 3: Run the API

Now it's time to run the API in **Anypoint Studio** to see how it all works together.

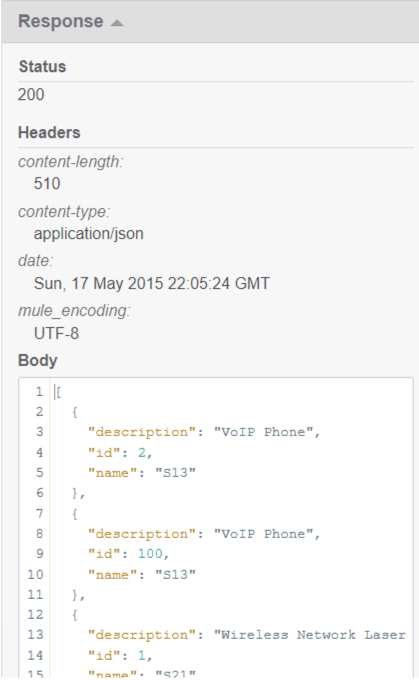
1. Start the application using Debug this time. Right-click the application
2. **Debug as**
3. **Mule Application**



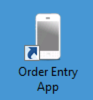
1. Once the **API Console View** opens in Anypoint Studio, click on the **GET** button on the /**orders** resource.



1. Under **Try it**, view the request body then click on the **GET** button below.



1. Once the request is served, you should see a **200** response code and the JSON response of our API, returning the actual orders coming from the Internal Sales Order system.



1. Test the app from the Mobile Order Entry Application as done is the previous lab. You should see the orders retrieved from the **Internal Sales Order API**.

## 

## Summary

In this lab, you completed the following steps:

[Step 1: Configure the RAML Consumer](#_pxqjtxd84ch4)

[Step 2: Create DataMapper transformations](#_8avcbmto54bc)

[Step 3: Run the API](#_subkunlz4qxi)

These steps allows you to easily consume internal and external API's, allowing us to quickly create aconnectivity layer on top of System and Process APIs. In this case, we used the HTTP Connector to consume a REST RAML based API and transformed the data using Data Mapper.

Check out the doc for more information on using:

* the new [HTTP Request Connector](http://www.mulesoft.org/documentation/display/current/HTTP+Request+Connector) with the RAML Consumer
* [Data Mapper](http://www.mulesoft.org/documentation/display/current/Datamapper+User+Guide+and+Reference)

Congratulations! You have finished Lab 7.

Please update the spreadsheet indicating you have completed Lab 7.