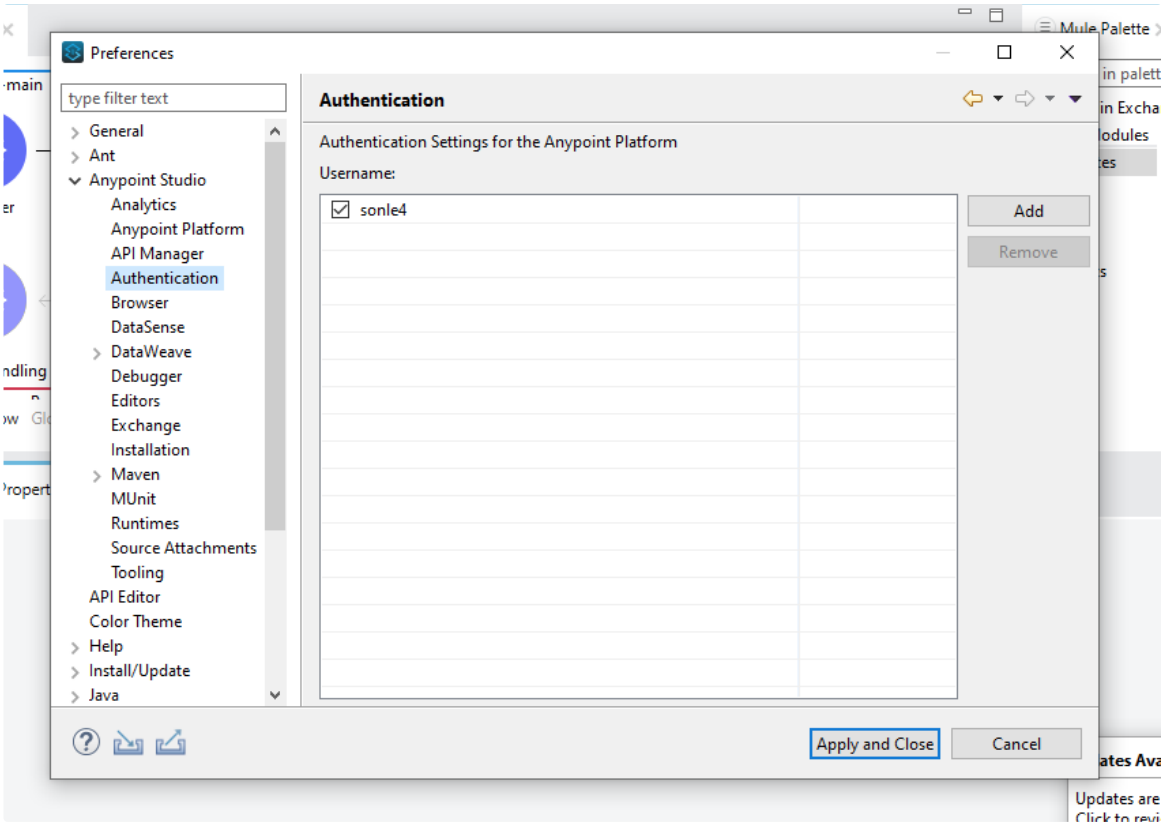
Implement API with Anypoint Studio

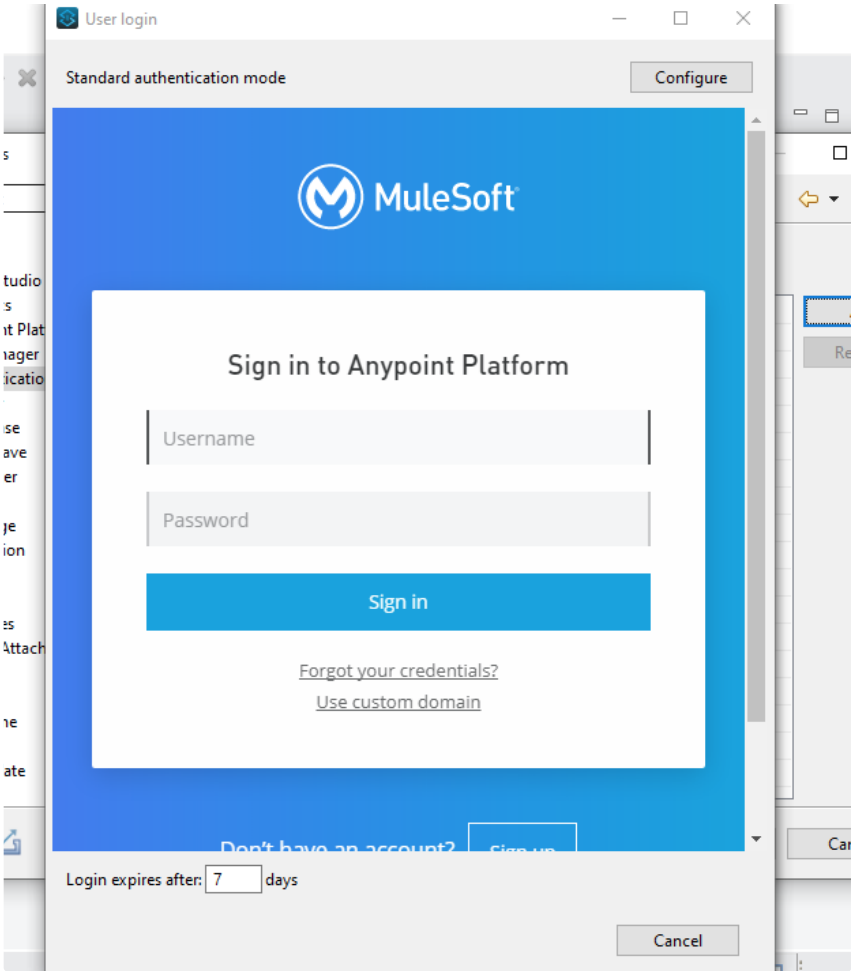
Sync Anypoint Platform account with Anypoint Studio

Go to Window → Preferences

Under Anypoint Studio section, click on Authentication item

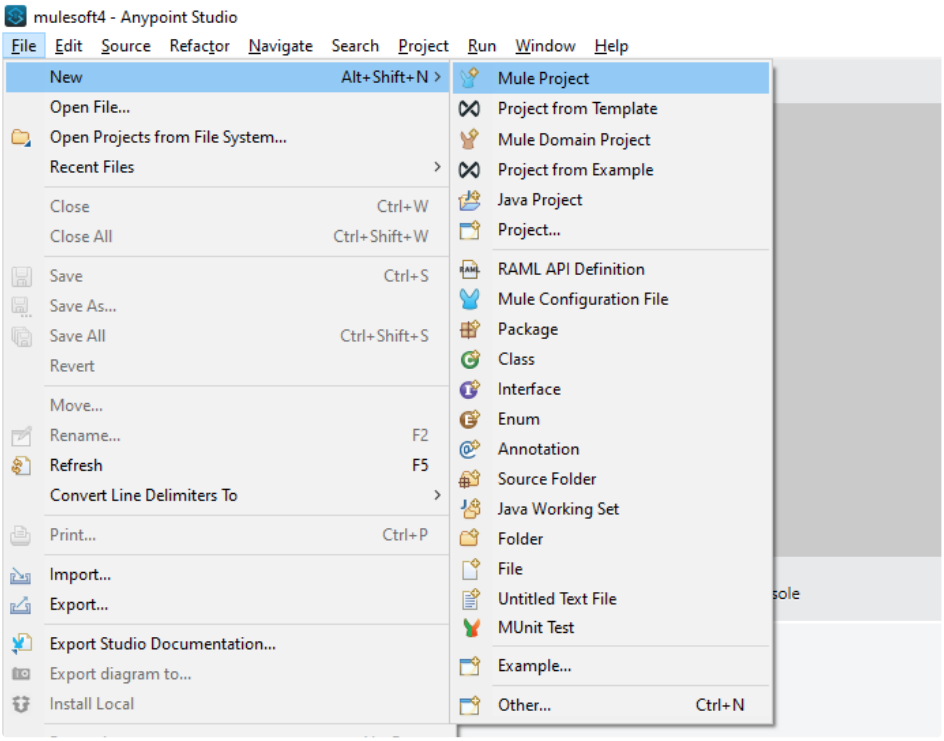


Click Add button, then sign in your Mulesoft account

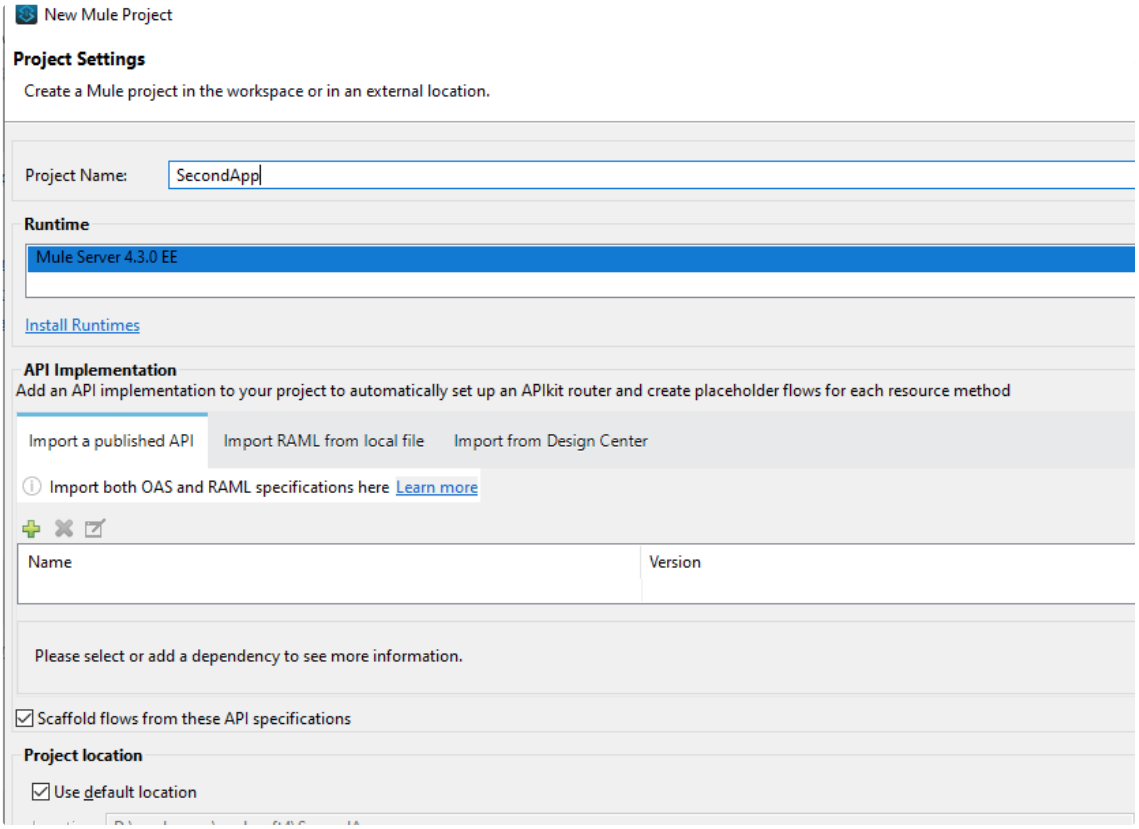


Create new Mulesoft API project in Anypoint Studio

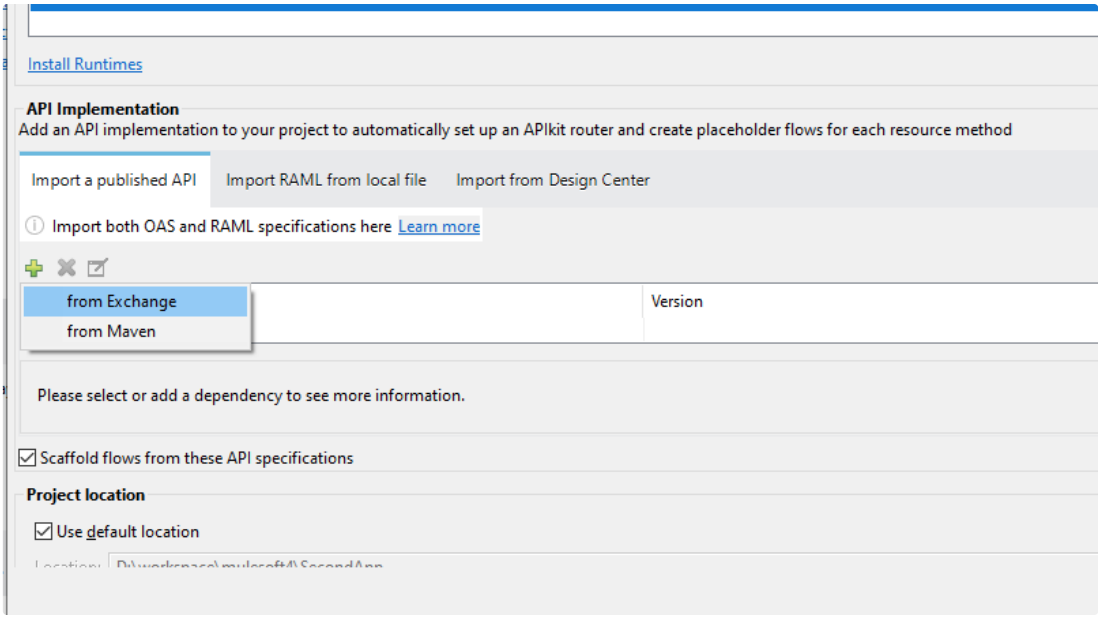
Go to File → New → Mule Project



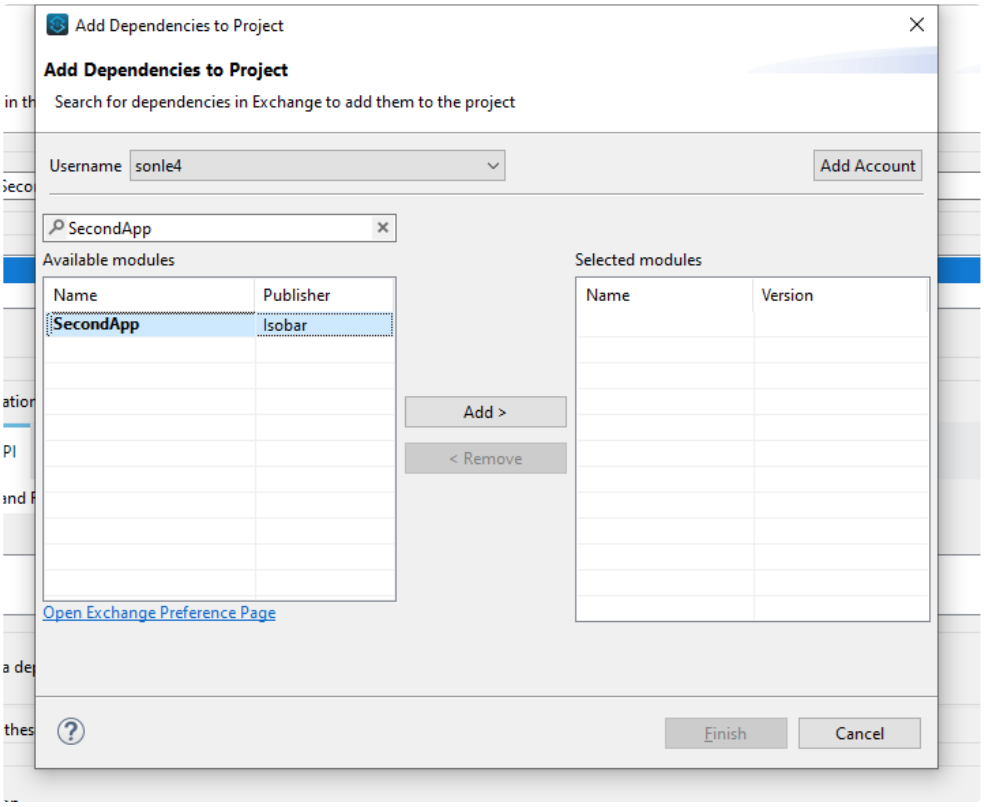
Input the project name



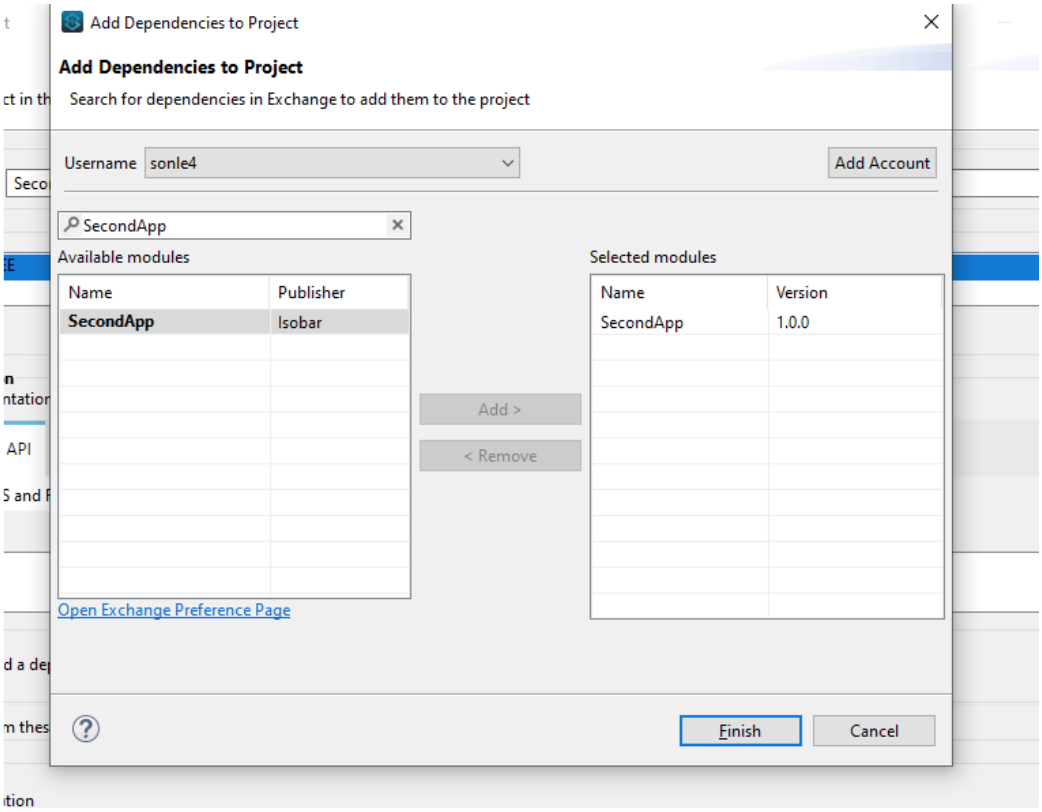
Under Import a published API tab, click green plus button, choose from Exchange



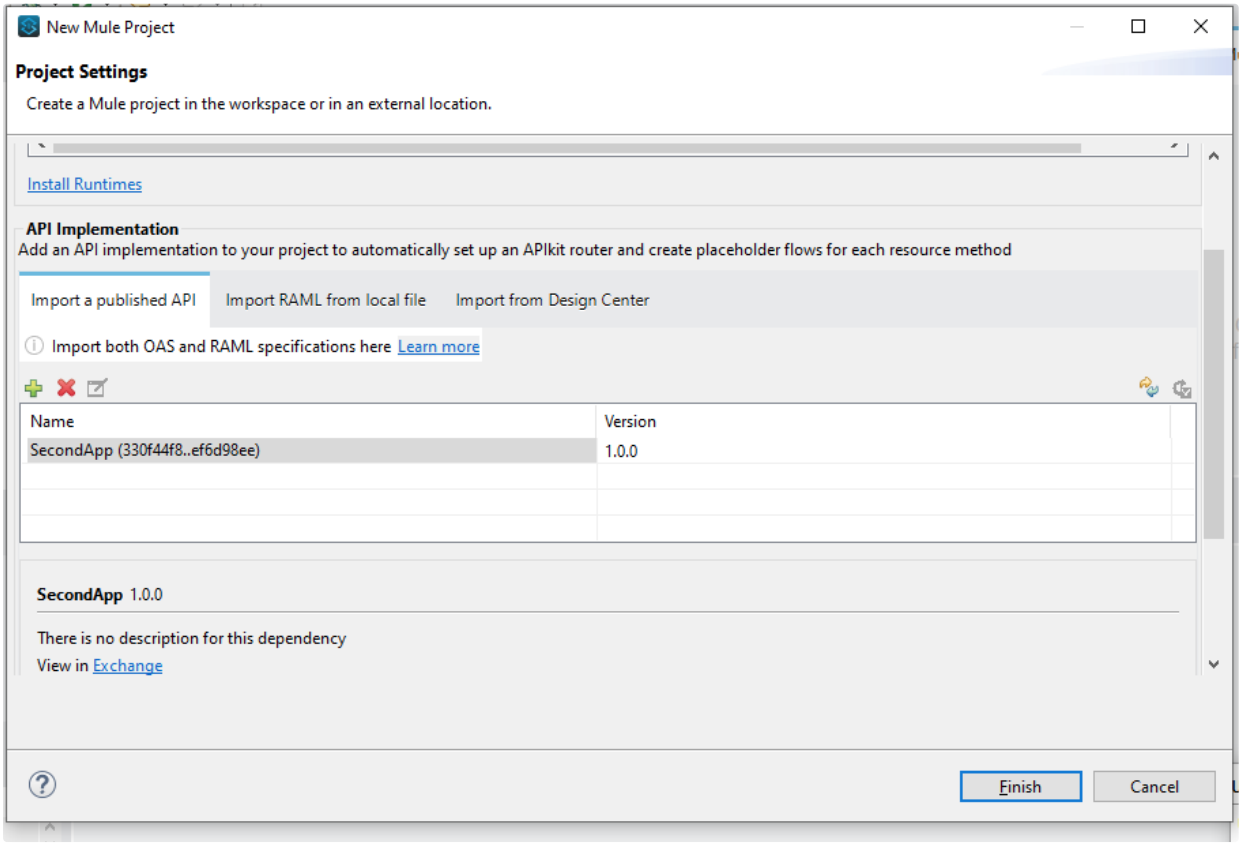
In search box, type the project name that you published to the Exchange in previous article



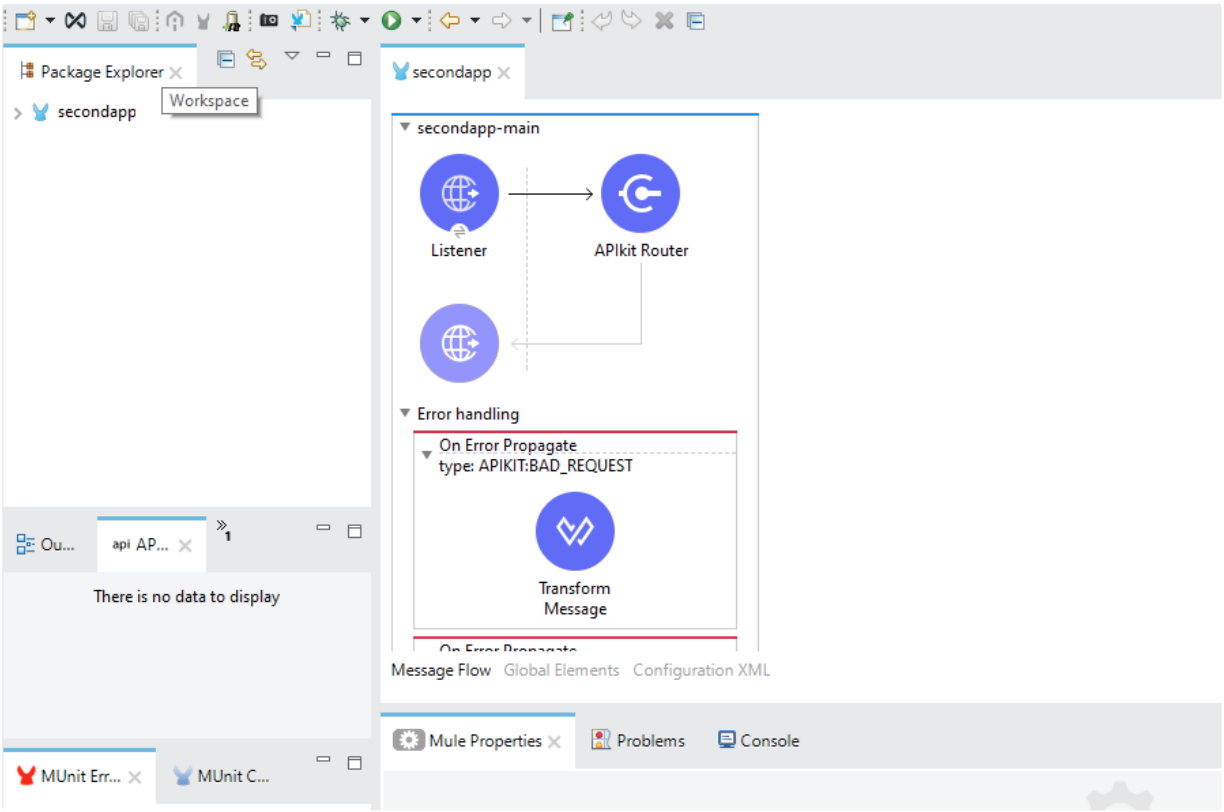
You will see the app under the list, click on the app then click Add button, then click Finish button



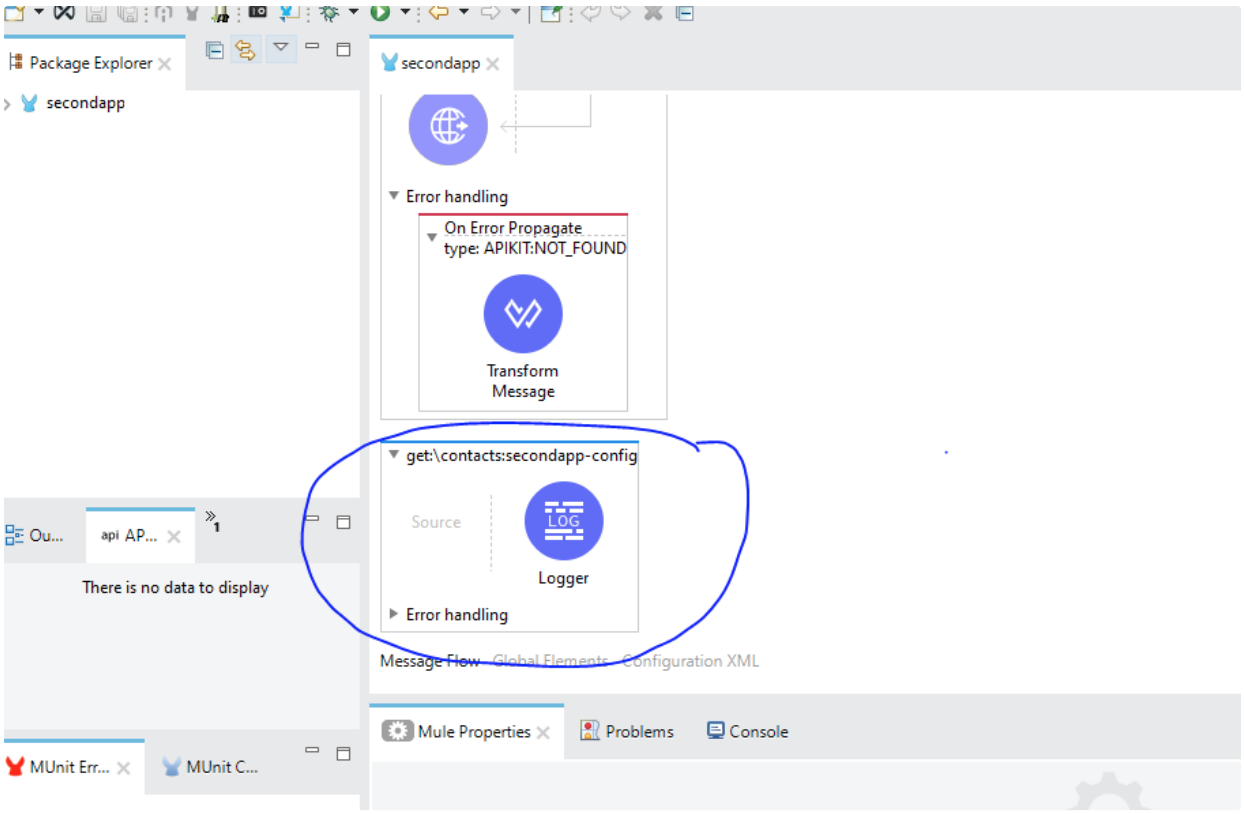
Click Finish button on the Mule Project panel to create the project



The project will be created in while



Now, go to the entry point that we want to implement

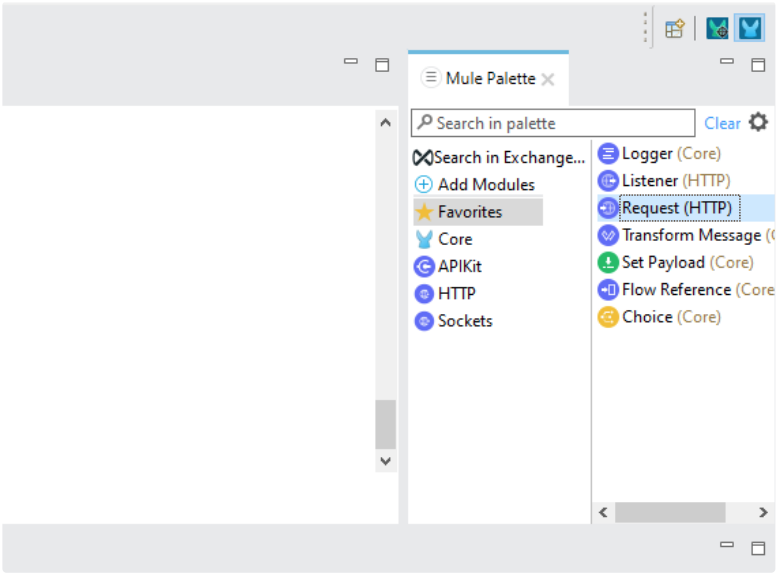


We must prepare an external data, we can use many different kinds of the data from a database, from SOAP webservice, from Restful API, etc. In this article, we just use a simple Restful API

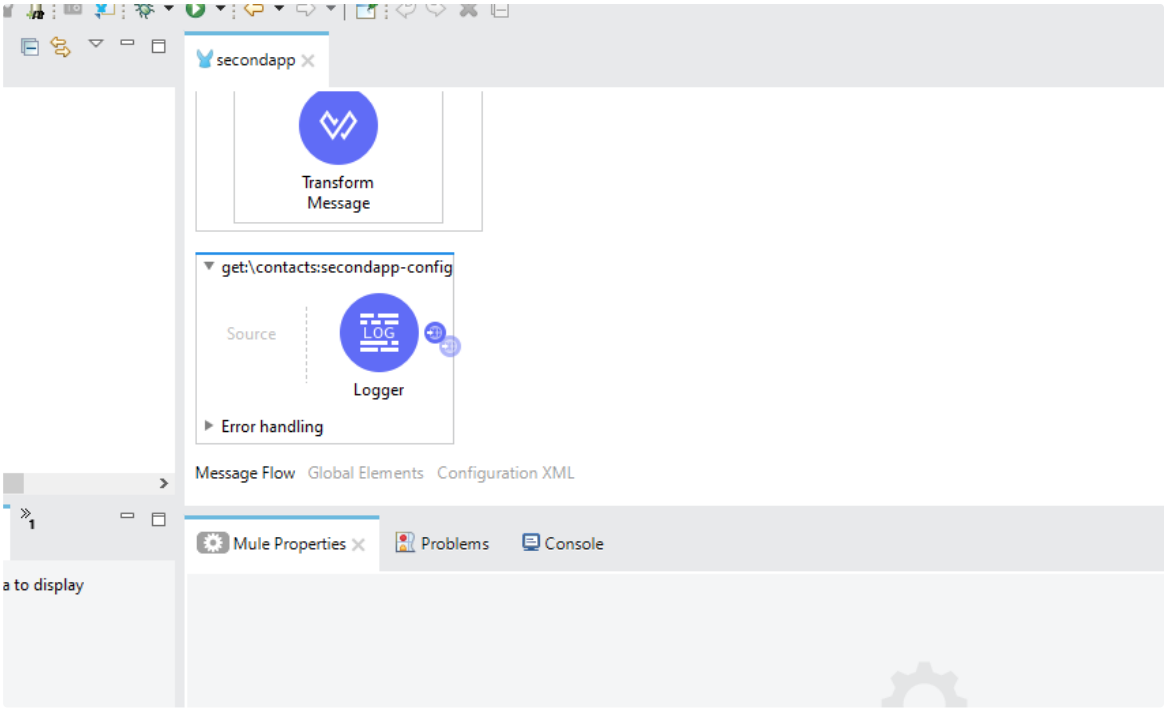
I have prepared an API https://60499478fb5dcc001796a453.mockapi.io/secondapp/contact

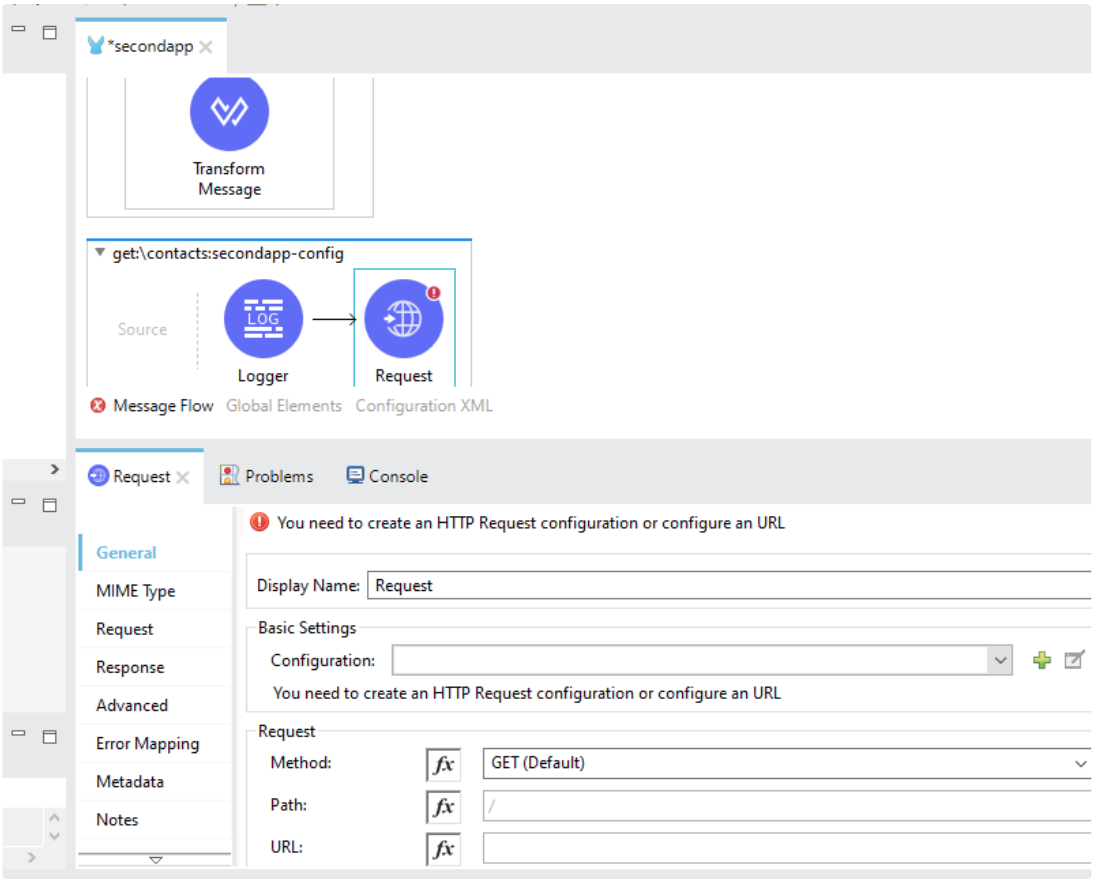
We will use an HTTP Request node to get the data from the API and use a Transform Message node to transform the data to expected data

Under the Mule Palette perspective, search Request, then choose Request (HTTP) node



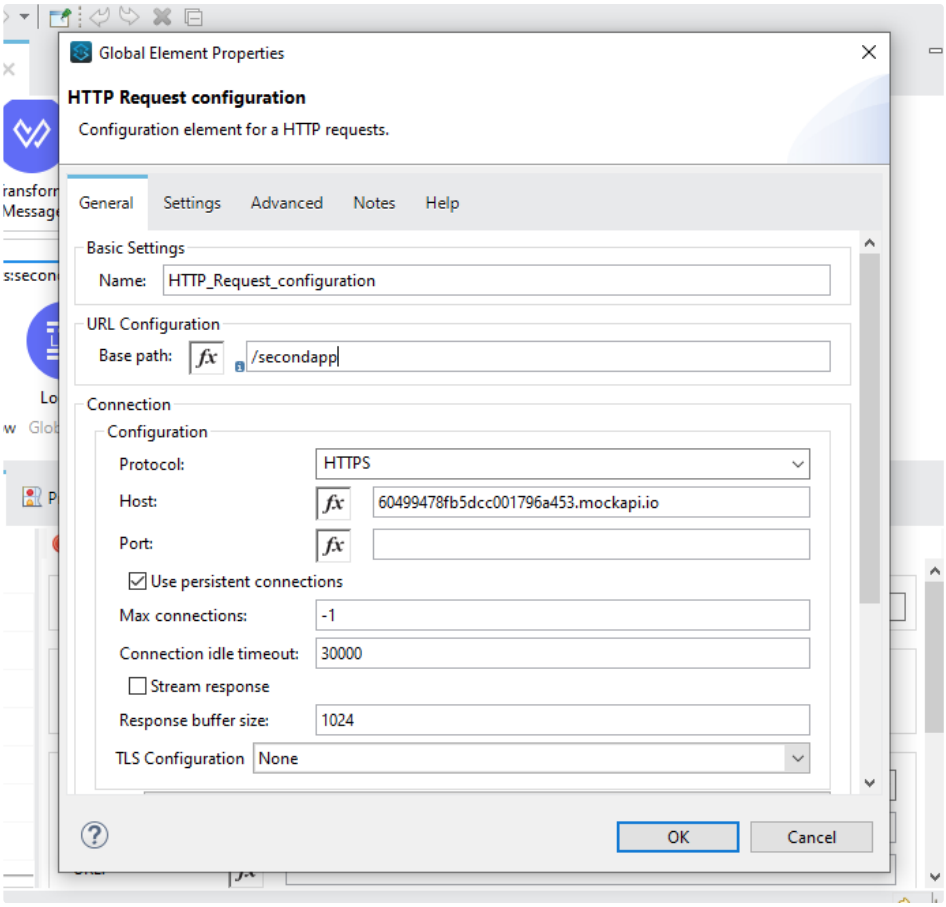
Drag the node to the main panel and connect with the Logger node



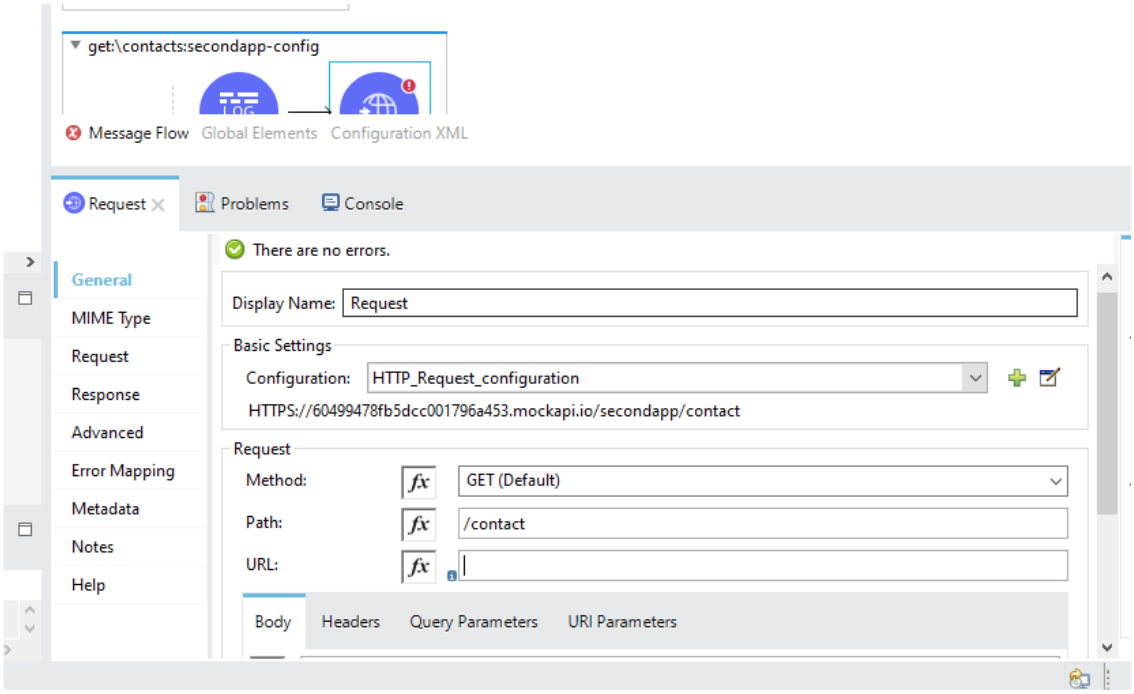


Click on the Request node, the Anypoint Studio will open new Tab allow us to fill the necessary information for this node.

Under the Basic Settings, click on green plus button beside the Configuration input



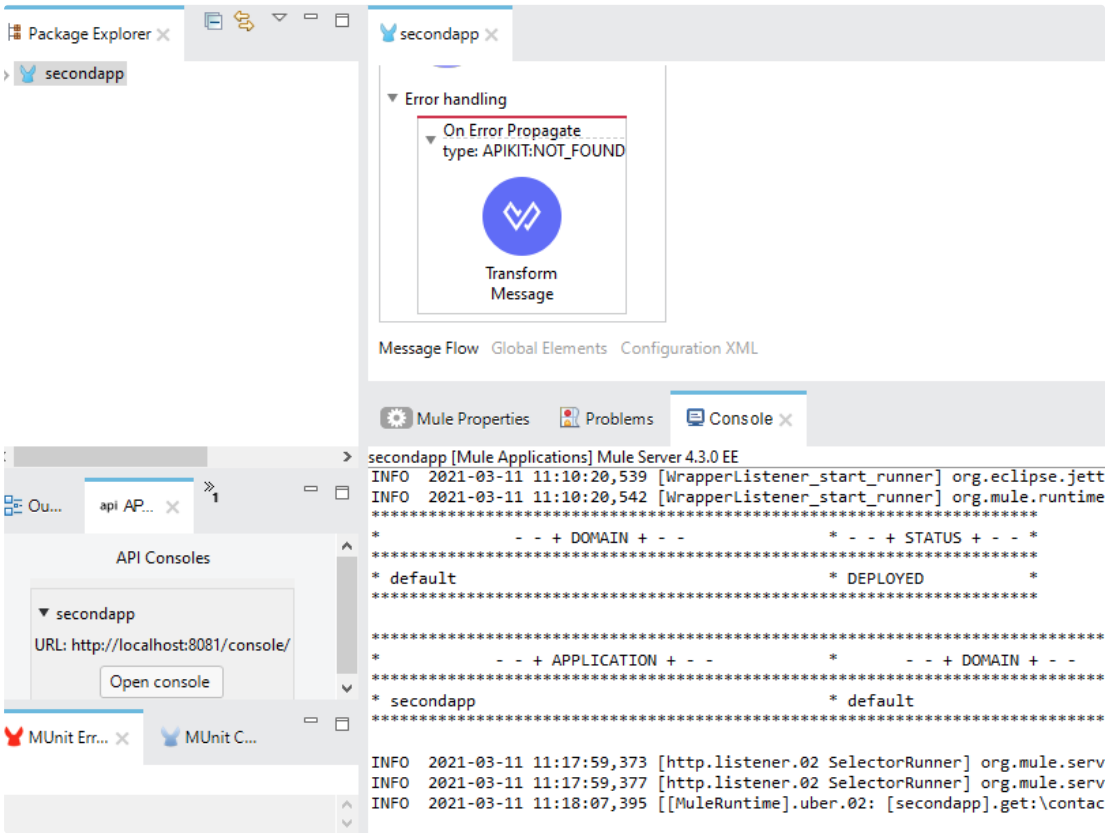
Input the Host, select HTTPS as protocol, input the base path then click OK button



We are using the GET method for this request and input “contact” as the path for the request

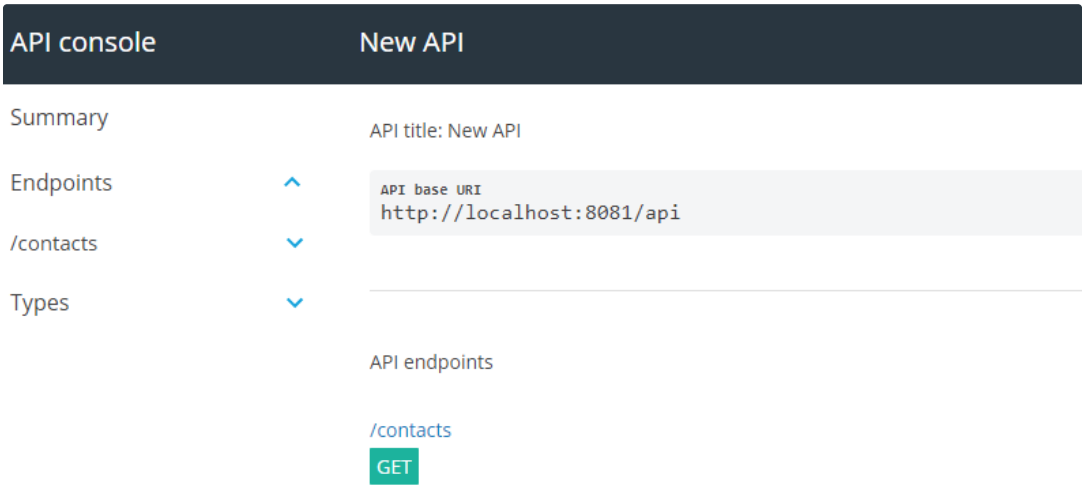
Save the project, right now, you can try to run the app on your local by go to Run → Run as → Mule Application

The Anypoint Studio will build the Application and run the API on http://localhost:8081/console

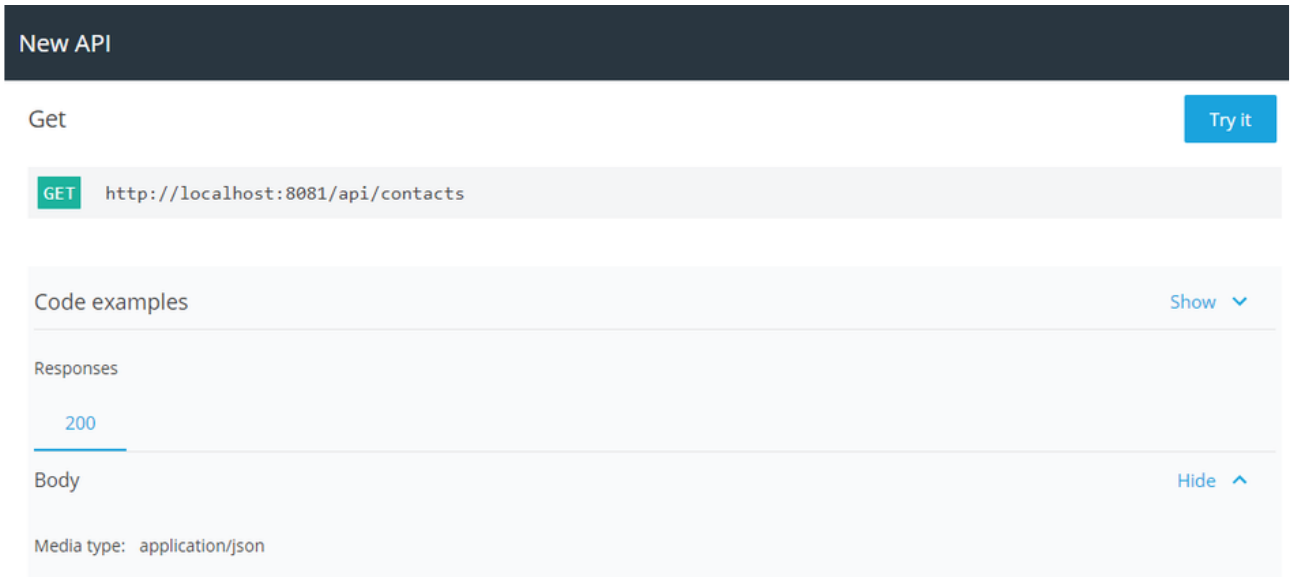


Open your browser and go to http://localhost:8081/console, you can see all the information about the API

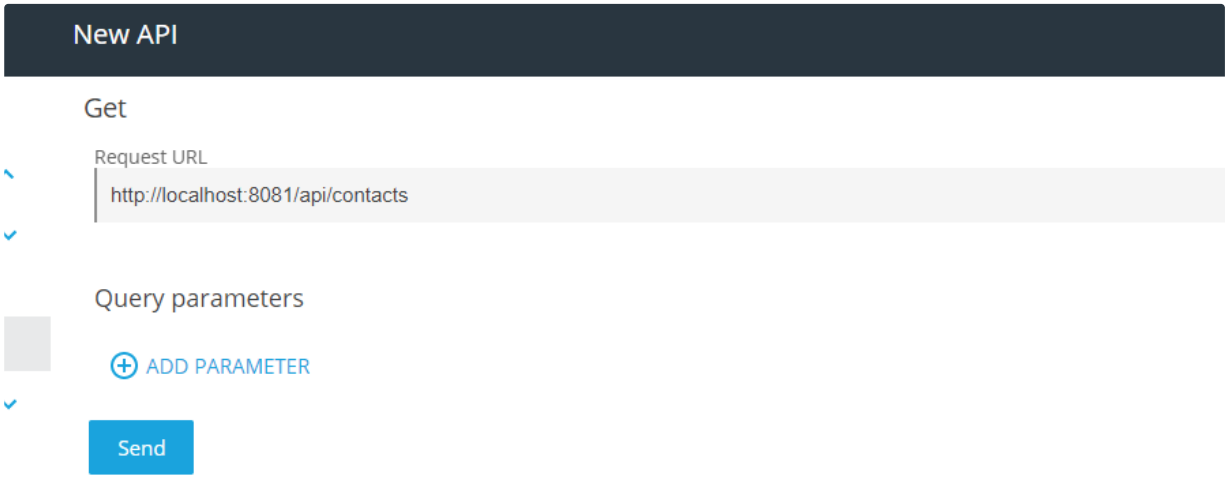
Let try to run the API by click on the GET method of the contacts endpoint



Then click on the Try it button



Then click on the Send button



You will get the data from the external API with 200 status code

200 OK 1586.53 ms

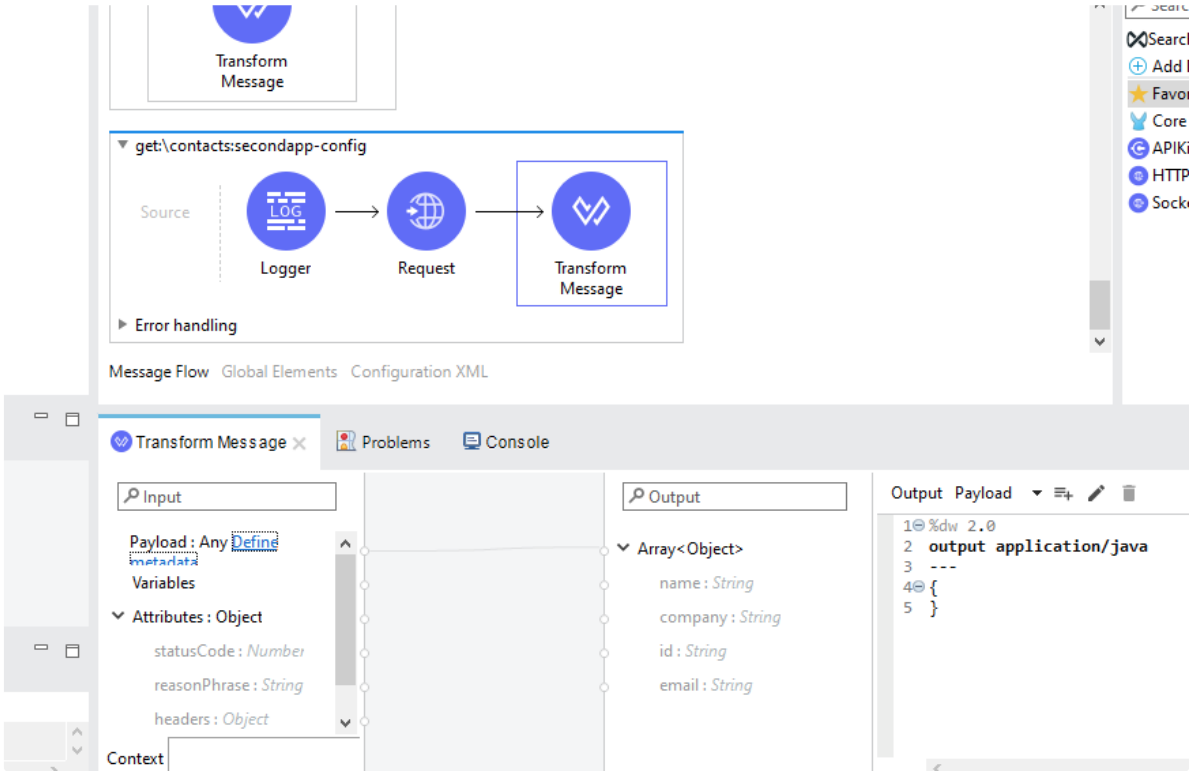
Copy Save Source view Data table


```
[Array[50]
  -0:  {
      "Id": "1",
      "Name": "Name 1",
      "Email": "Email 1",
      "Company": "Company 1"
   },
  -1:  {
      "Id": "2",
      "Name": "Name 2",
      "Email": "Email 2",
      "Company": "Company 2"
   },
  -2:  {
      "Id": "3",
      "Name": "Name 3"
```

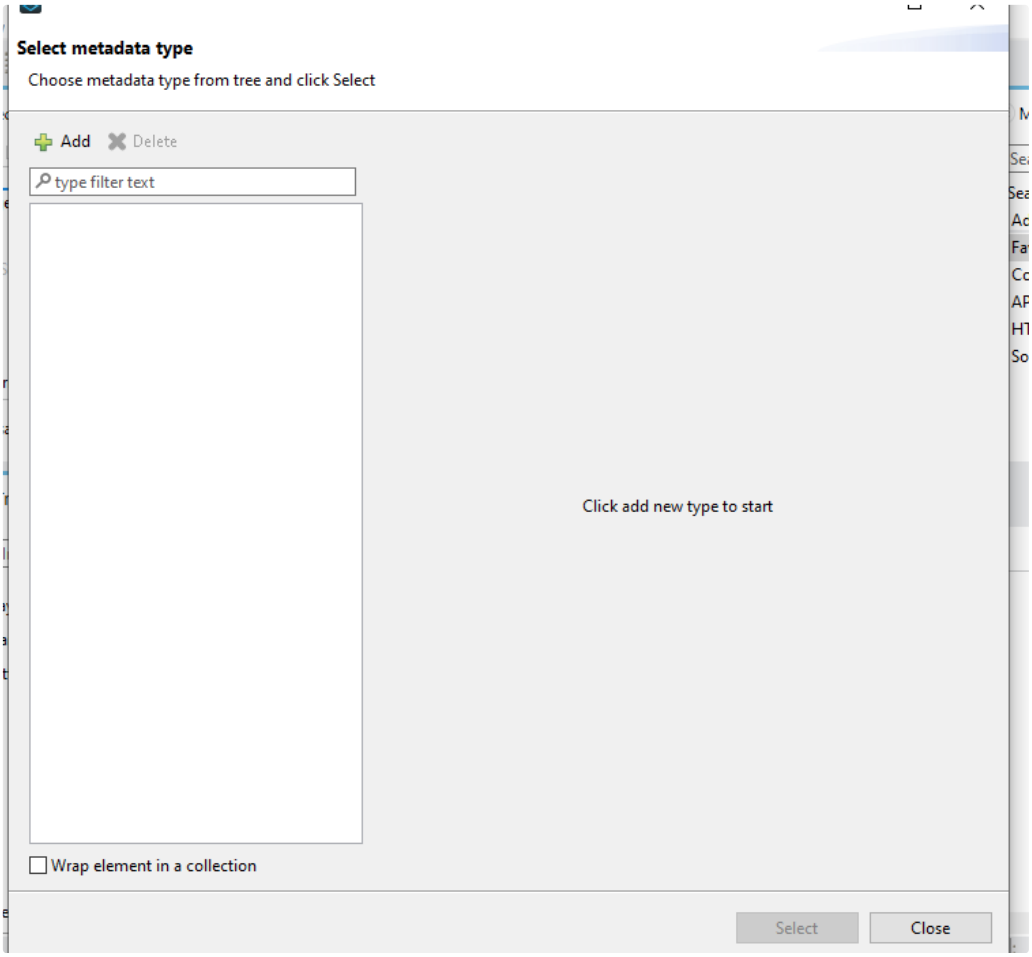

As you can see, the data is same the external API data but this isn't our expected data. Our data should has the lowercase for the object property as we defined in previous article.

We need the Transform Message node to transform the data to our expected data

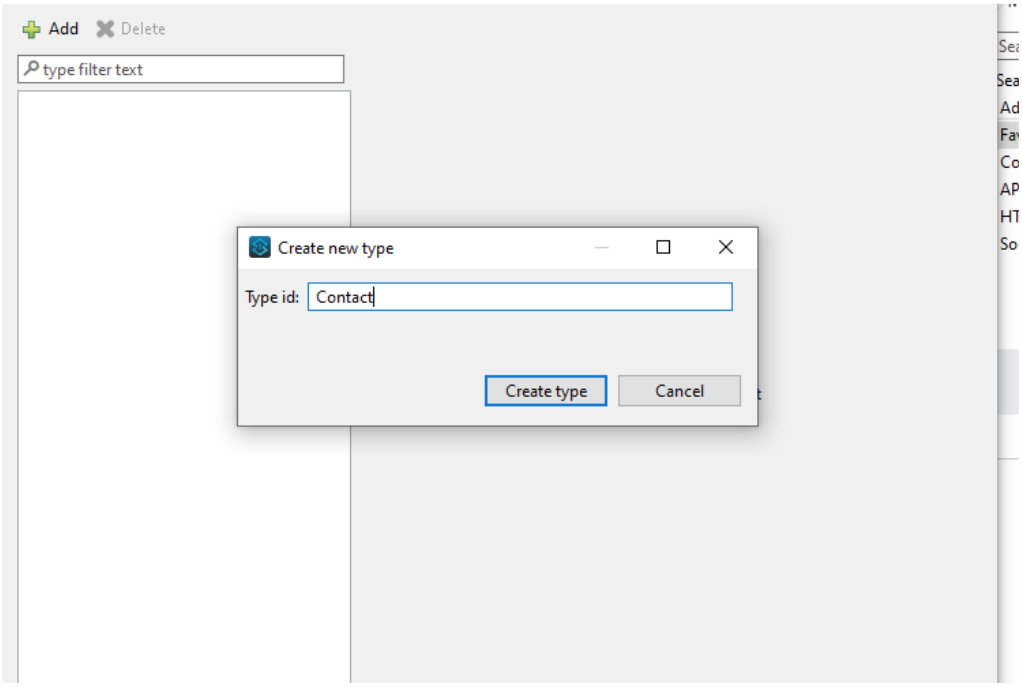
Under the Mule Palette perspective, search Transform, then select the Transform Message node, drag it to main panel and connect with the Request node



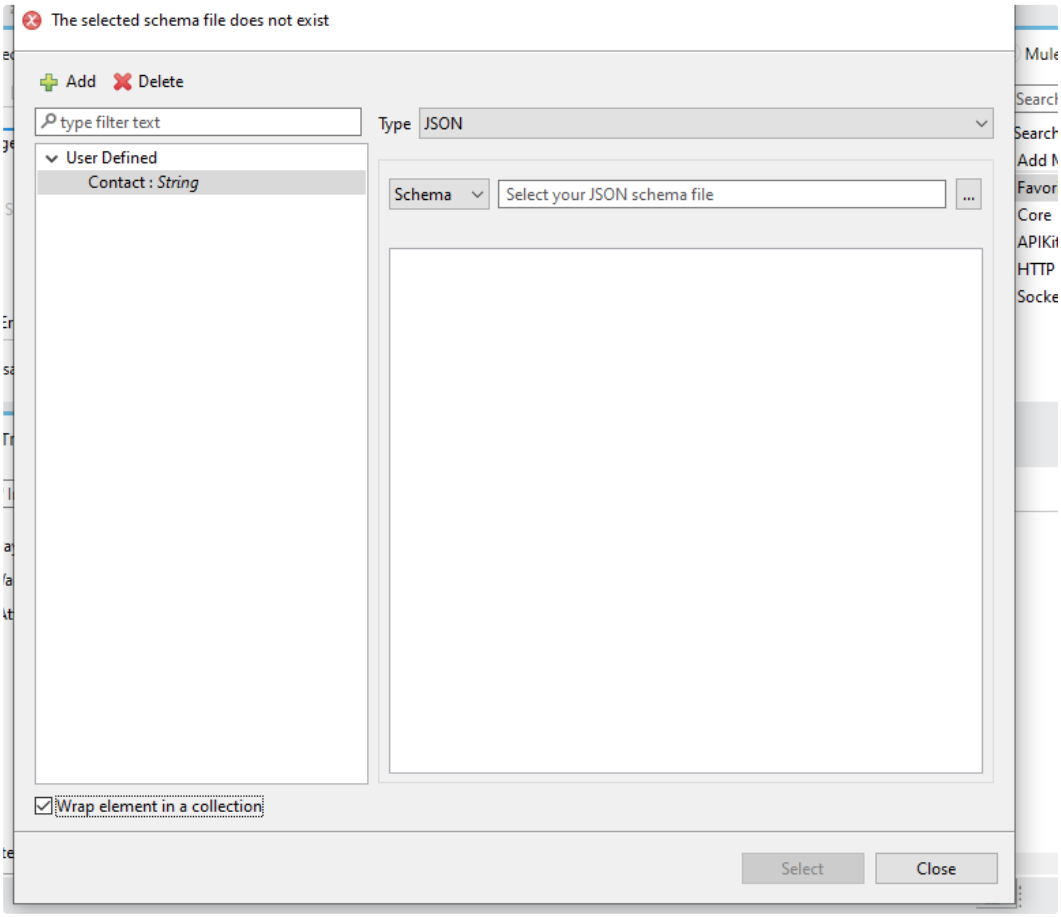
Under the Transform Message tab, beside the Payload, click on the Define metadata text link



Click Add button, input Type ID then click Create type button



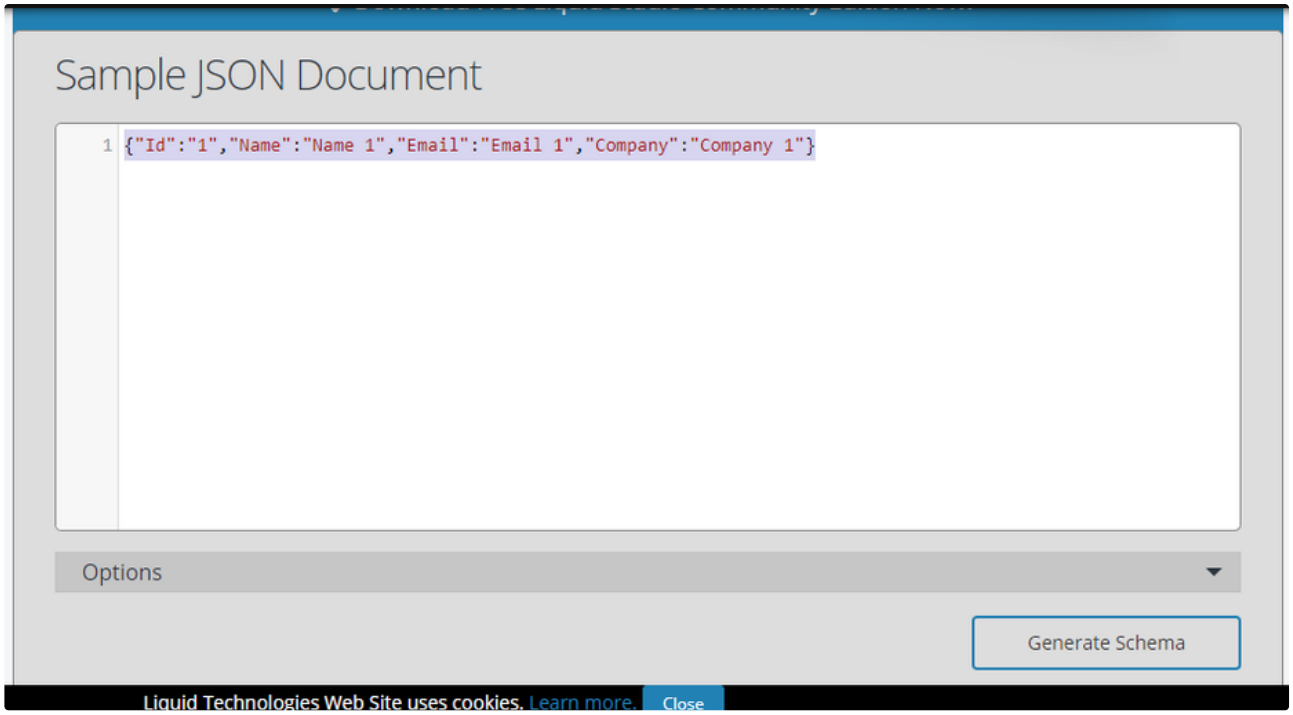
In the type dropdown box, select JSON

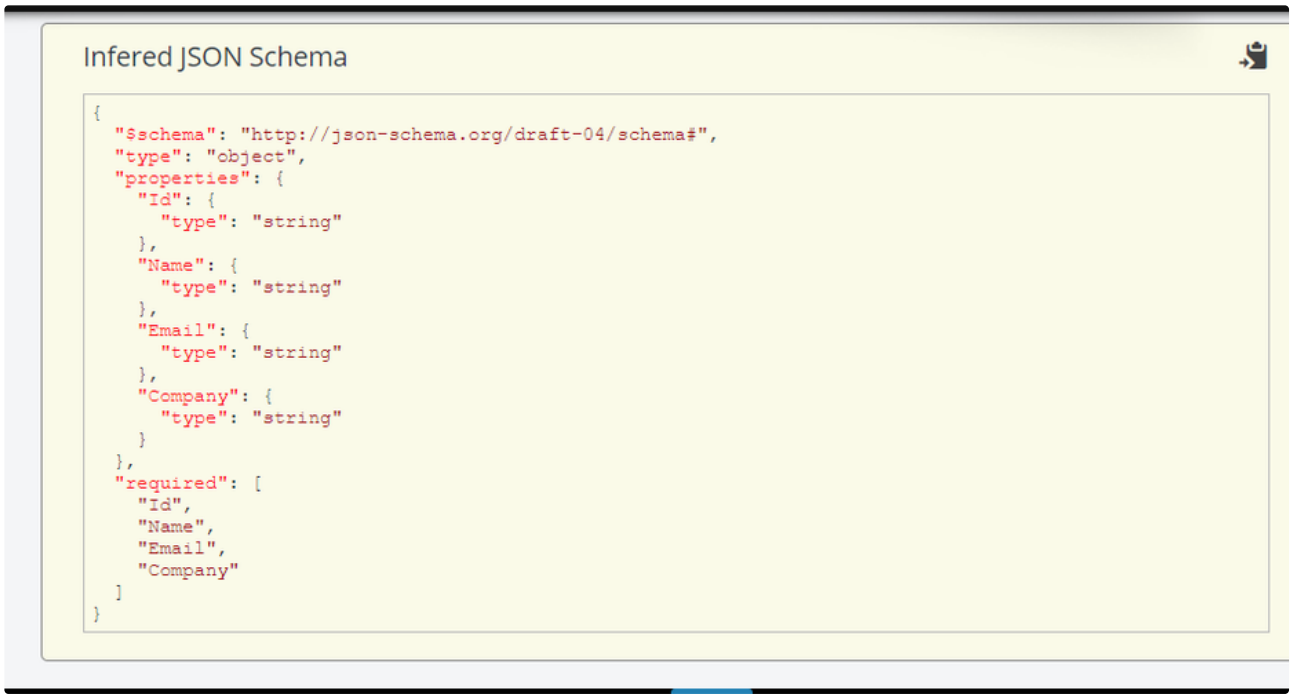


We need to prepare a JSON schema for the Contact object

Go to Free Online JSON to JSON Schema Converter

Input {"Id":"1","Name":"Name 1","Email":"Email 1","Company":"Company 1"} as sample JSON data, then click Generate Schema button to get the JSON schema

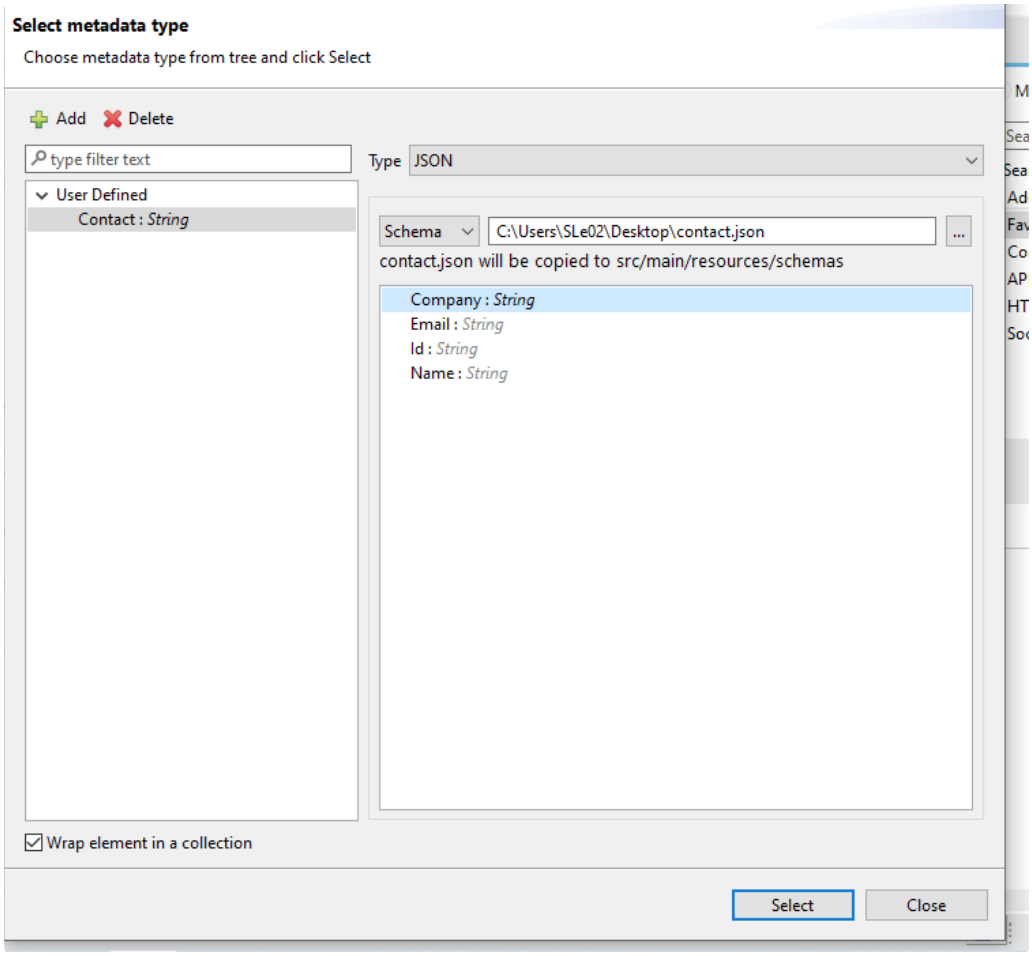




Save this string to a text file with extension .json

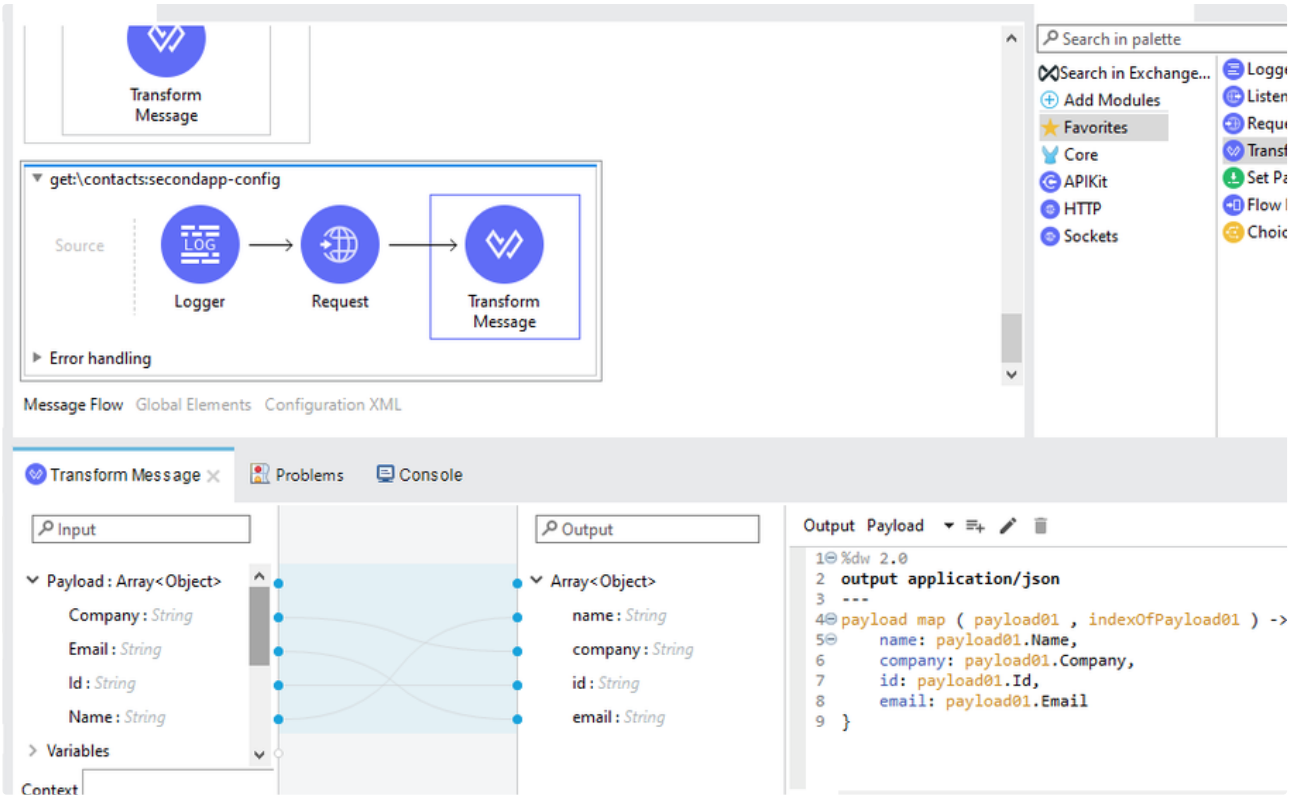
Back to Anypoint Studio, In the Schema, select the file that you saved before

Select **Wrap element in a collection** checkbox



Click Select button

Drag the properties on the left column to the right column with the corresponding property to connect them



Now, run the Project again, try to run the API again and you can get the expected data