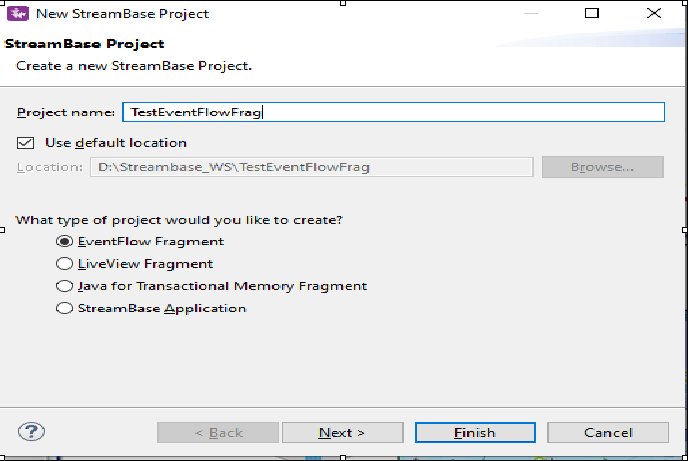
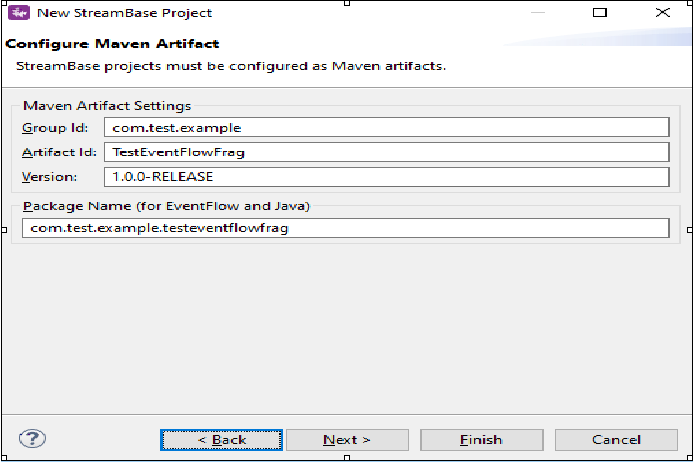
**Create Streambase Studio Project, Create Application Module and Configurations:**

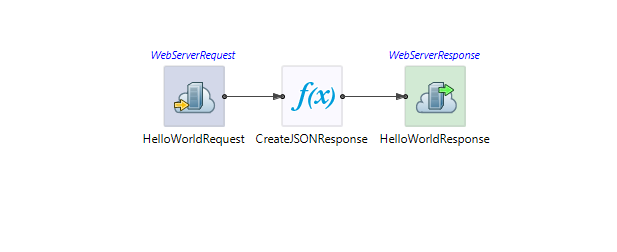
⦁ Create the EventFlow fragment and place all your logic inside this fragment. Select the Radio button as **EventFlow Fragment** as shown in below Screenshot.

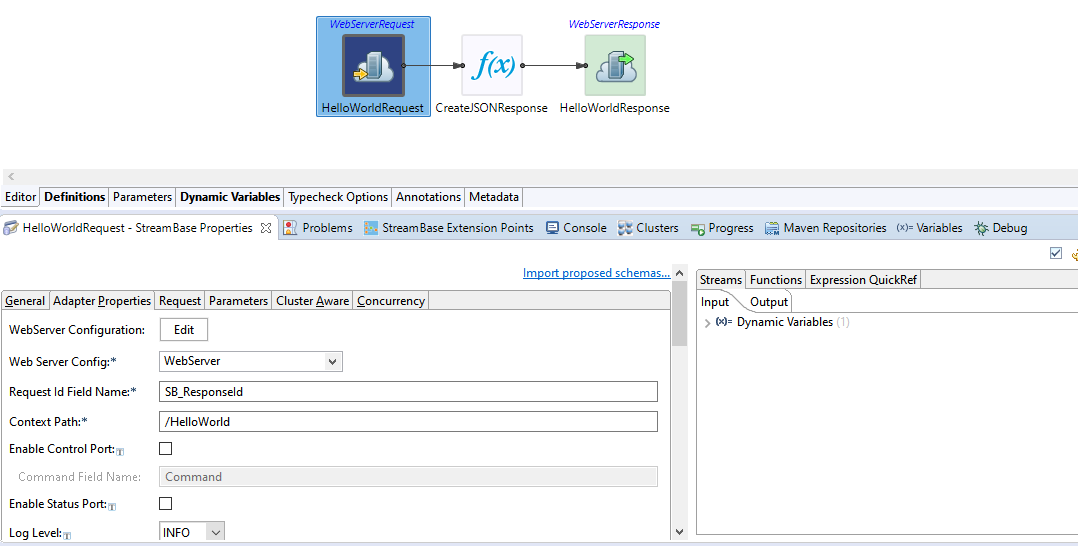


⦁ Click Next and provide the value for **Group Id**, **Artifact Id** and **Version**. Provide the packagename as <<Group Id>>.<<any name>> as shown in below screenshot. Please note down these three parameters and click finish.



⦁ Create a WebServerRequest by Drag-Drop Adapters,Java Operators. Configure the adapter with all the required parameters, Orchestrate incoming data and finally return Response by WebServerResponse adapter. After the above application is created just do a clean install. Sample for same is as shown below:





⦁ For the WebServerRequest we need to create a configuration file(required to select in Adapter Properties-> Web Server Config) and the example for same is given below. Request Id Field Name is something like a global variable that you can define under Dynamic Variables as shown below and Context Path is the URI. Also need to configure the request and the same can be found in below screenshot.This would create a GET Configuration by default, but if some POST configurations need to be created than under Request we need to specify Request Data Schema and also the same need to be defined in Parameters.

<adapter-configurations>

<adapter-configuration name="webservers">

<section name="webserver">

<setting name="id" val="WebServer"/><!-- **This is the name you will see in dropdown for WebServerRequest->Web Server Config** -->

<setting name="Port" val="9191" /><!-- **This is the port you want to use** -->

<setting name="StartOnStartup" val="true" />

<setting name="IdleTimeoutMS" val="30000" />

<setting name="OutputBufferSize" val="32768" />

<setting name="RequestHeaderSize" val="8192" />

<setting name="ResponseHeaderSize" val="8192" />

<setting name="SendServerVersion" val="true" />

<setting name="SendDateHeader" val="false" />

<setting name="GZipMimeTypes" val="text/html,text/plain,text/xml,application/xhtml+xml,text/css,application/javascript,text/javascript,image/svg+xml" />

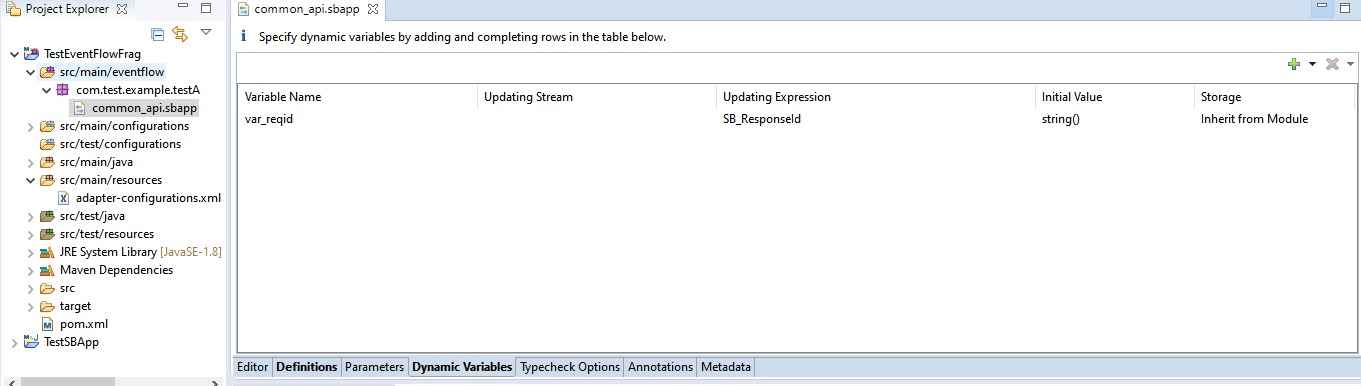
<setting name="WebSocketAsyncWriteTimeoutMS" val="5000" />

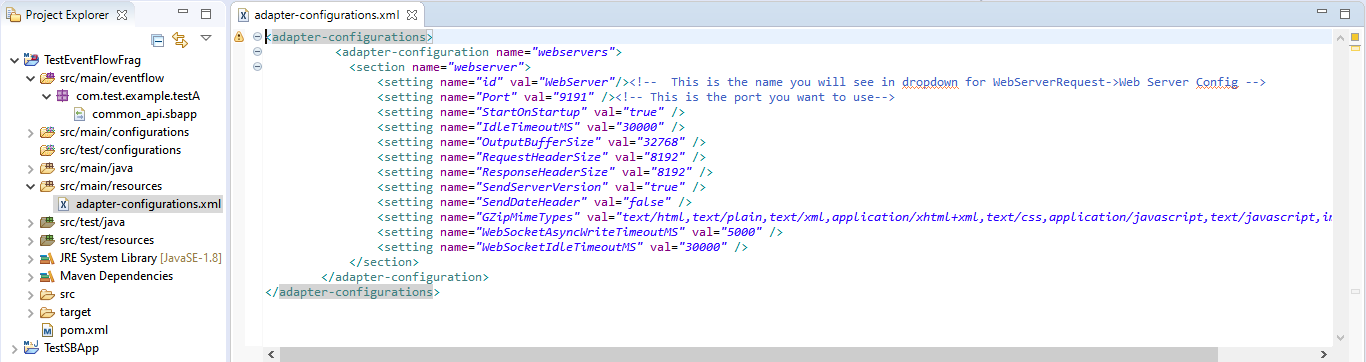
<setting name="WebSocketIdleTimeoutMS" val="30000" />

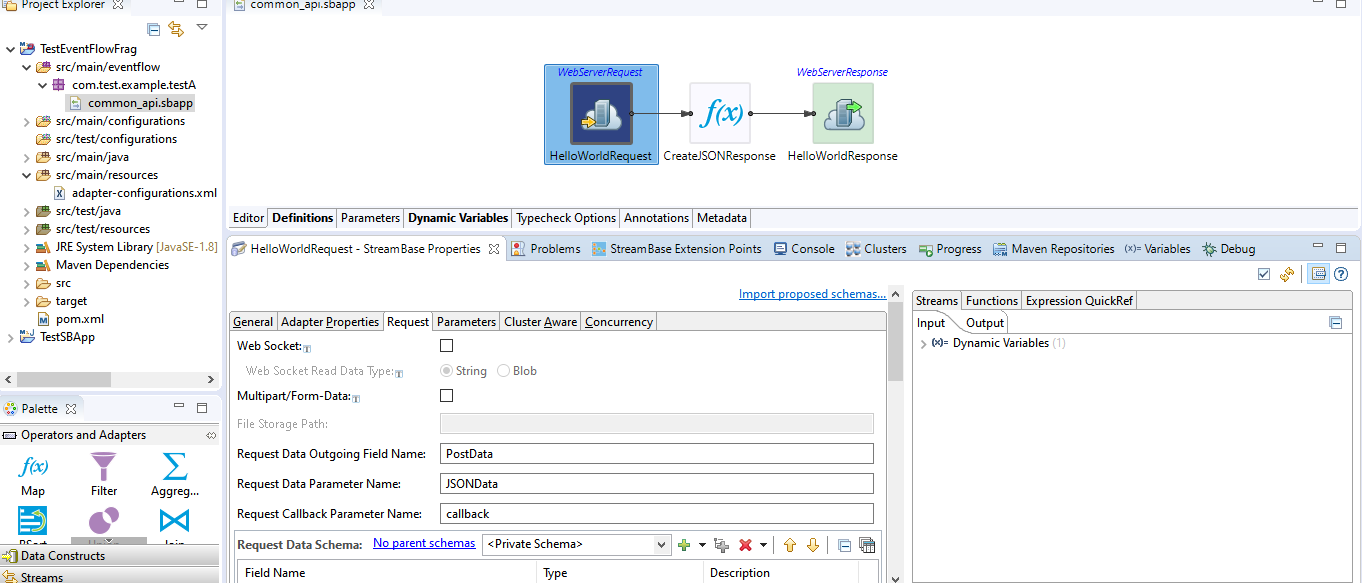
</section>

</adapter-configuration>

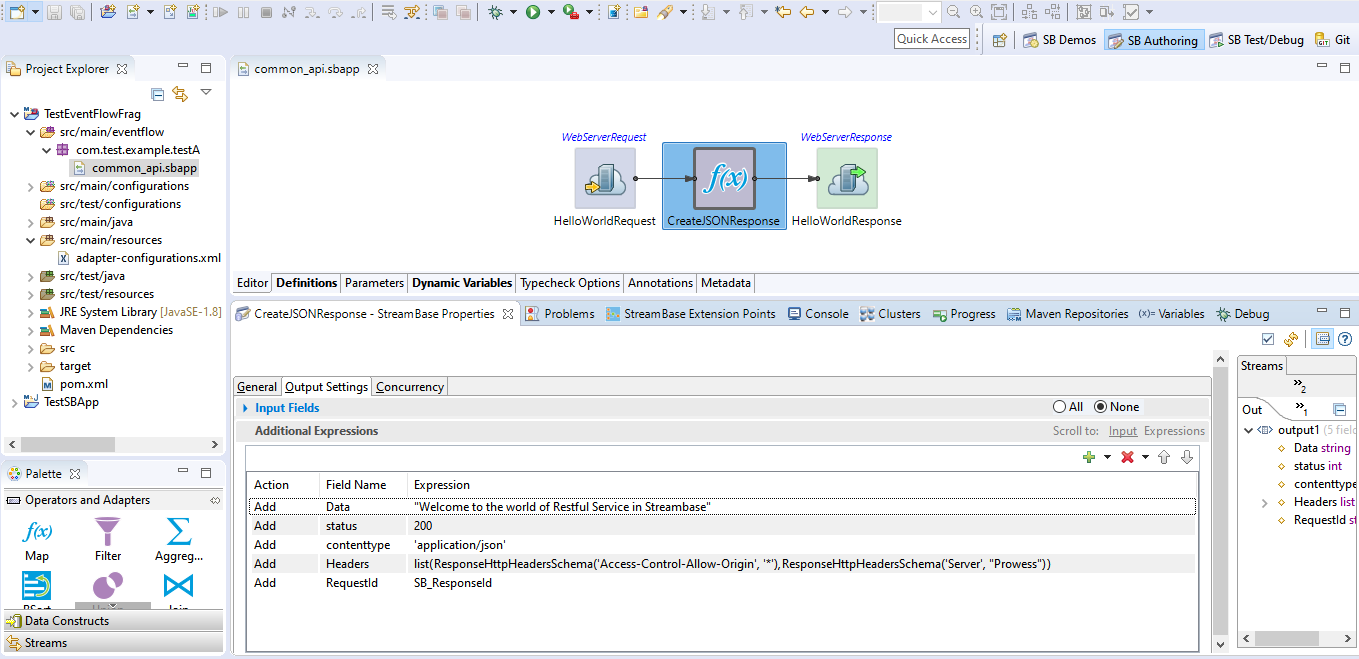
</adapter-configurations>



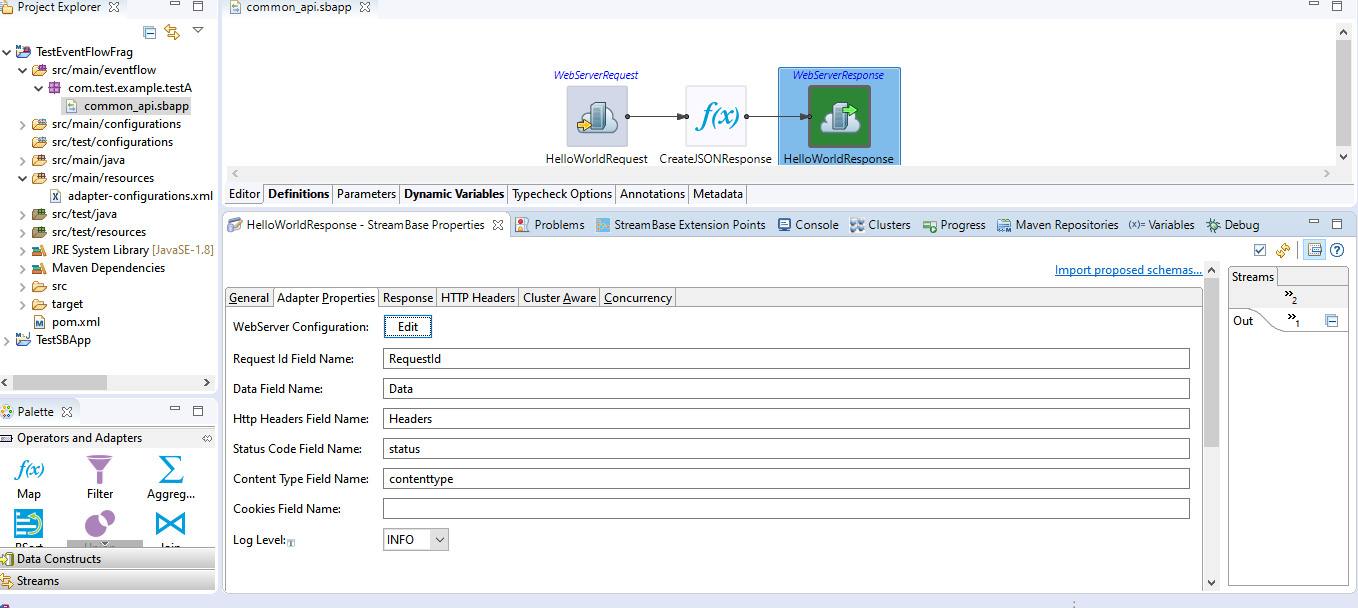




⦁ Now I have created a mapper with some values so that same can be mapped to the WebServerResponse and the same you can see in below screenshot.



⦁ As you can see in above screenshot, Data is something which is you can say it as a Response for the Rest call, Headers if any can be defined and RequestId is something to which the SB\_ResponseId is mapped which is the one I created in Dynamic Variables.Now this fields from mapper would be mapped to WebServerResponse and same can be seen in below screenshot.

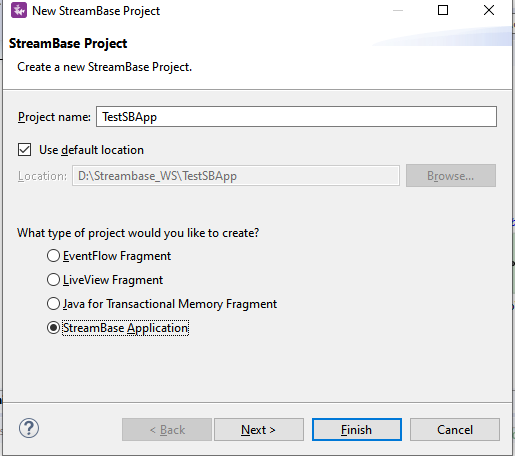


⦁ As seen from above screenshot, the only thing that need to be configured is Adapter Properties.All the values are been defined in mapper already and the same thing is mapped over her.

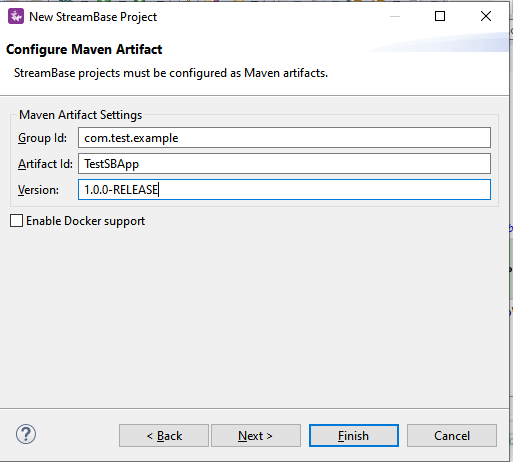
⦁ Now In the Project Explorer view, select and right-click the Streambase Application Project **(TestEventFlowFrag in my case)--> Run As>Maven Build...**

⦁ After that In the **Goals** field, enter **clean install** and Select the **Skip Tests** check box. Click Apply and Run.

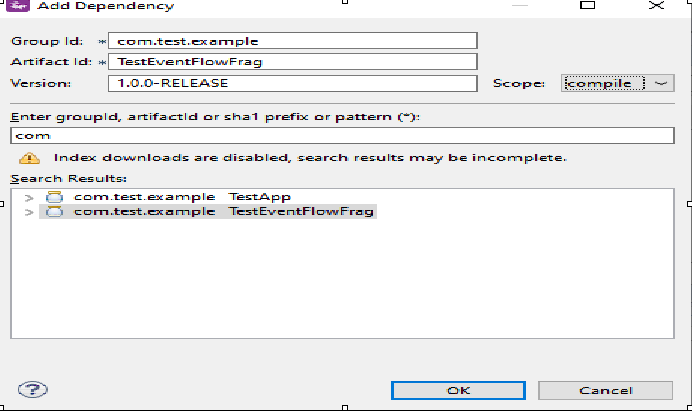
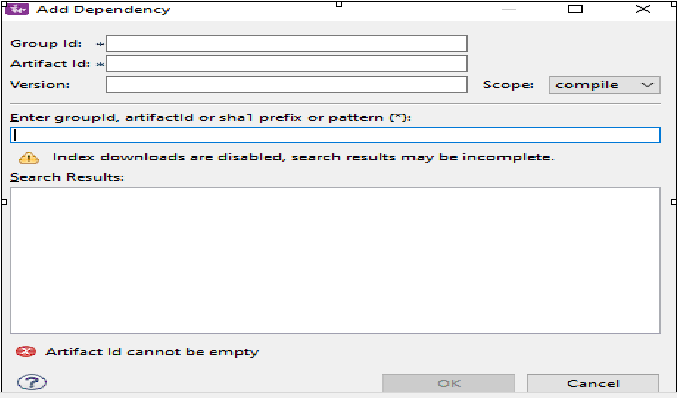
⦁ Now create a new Streambase Project. Select Radio button as **StreamBase Application**.



⦁ Now click next and make sure you provide the same **Group Id** and **Version** that you specified for the EventFlow Fragment created Earlier and click **Finish**.

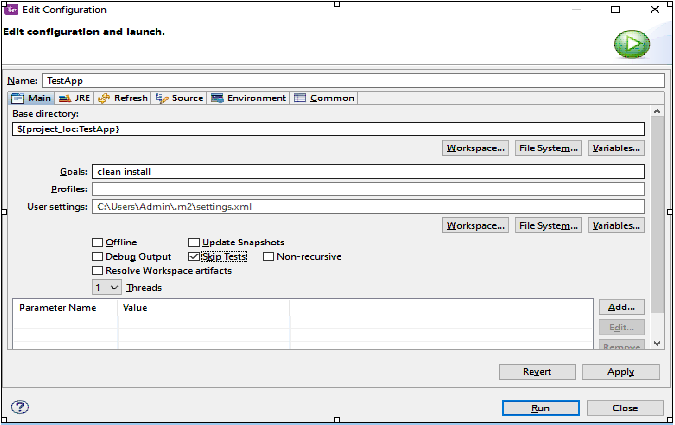
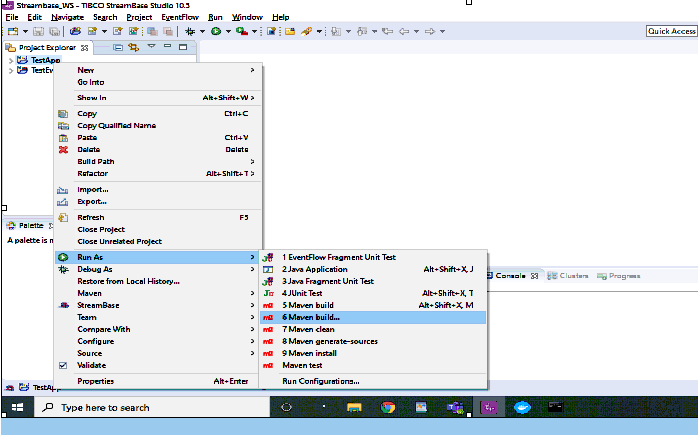


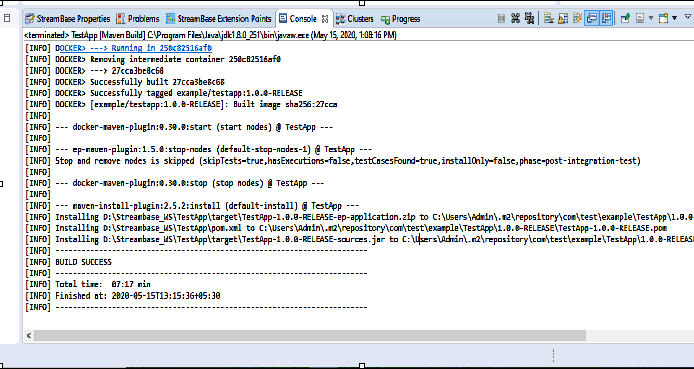
⦁ Add the EventFlow Fragment Project depedency to our main Streambase Application. Right click on Streambase Application Project **(TestSBApp in my case) --> StreamBase>Manage Project Dependencies**. Add the depedency for the EventFlow Fragment and click ok.



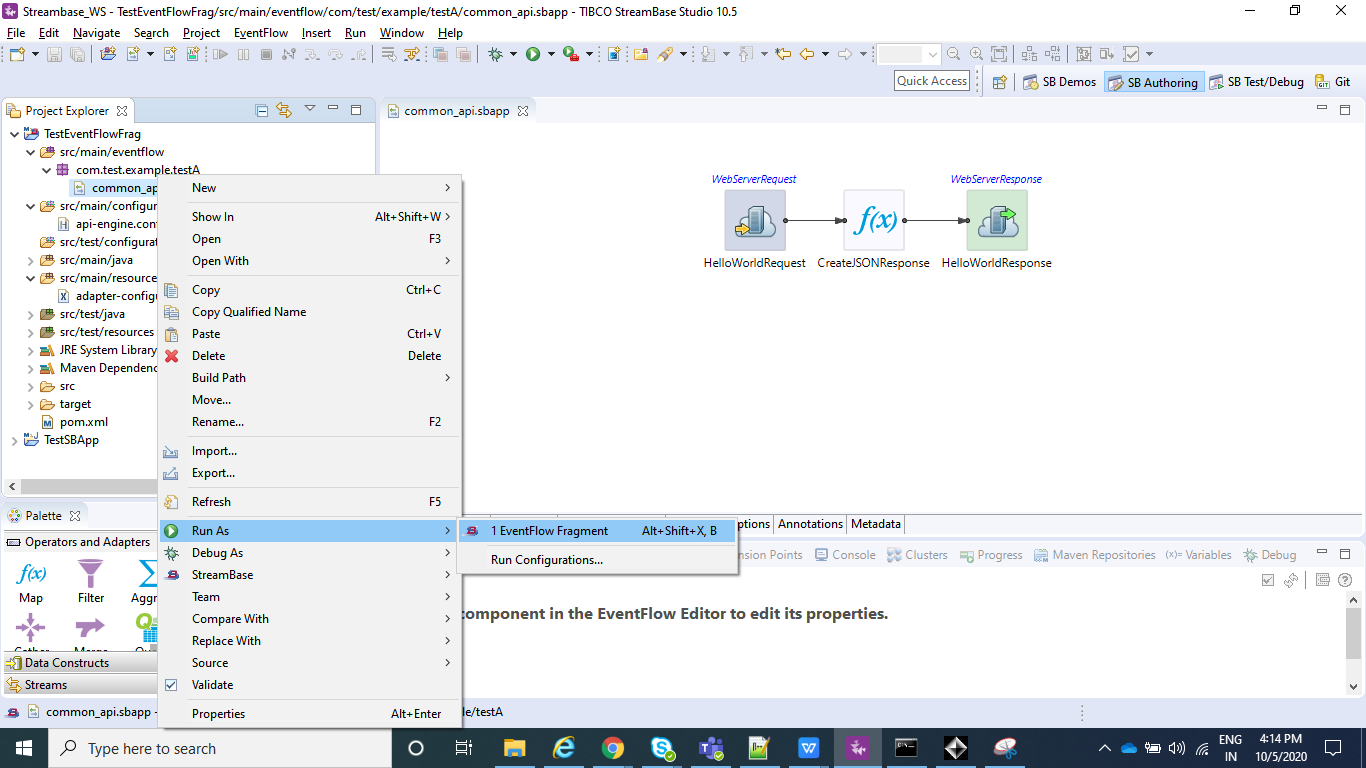
⦁ Now In the Project Explorer view, select and right-click the Streambase Application Project **(TestSBApp in my case)--> Run As>Maven Build...**

⦁ After that In the **Goals** field, enter **clean install** and Select the **Skip Tests** check box. Click Apply and Run.

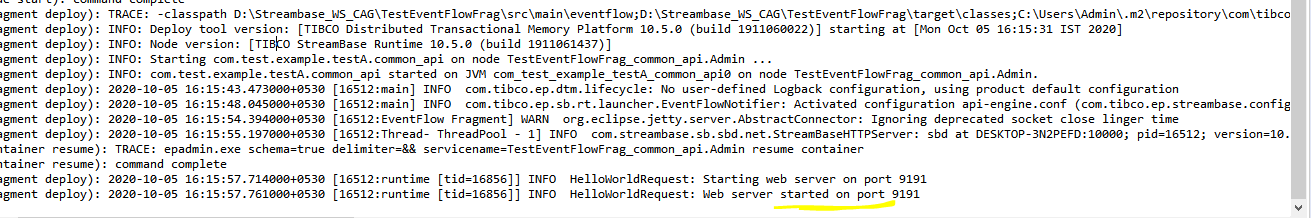


⦁ After the above step check the console and it should be showing BUILD SUCCESS message.

**Run the application in Studio:**

⦁ Right click on your .sbapp and select Run as EventFlowFragment as shown in below screenshot:

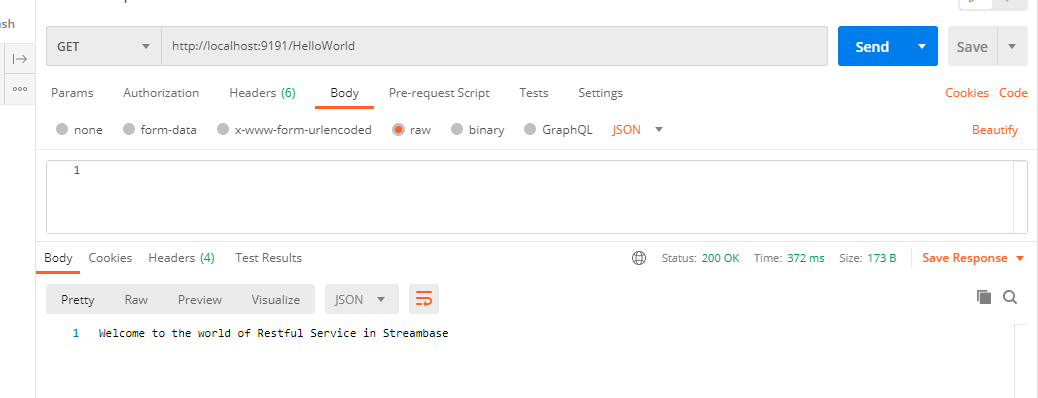
⦁ After you application is running you can see to the console logs as **Web server started on port 9191.**



⦁ Now in order to test the application we need some REST client, like postman.We can download the same from : **<https://www.postman.com/downloads/>**

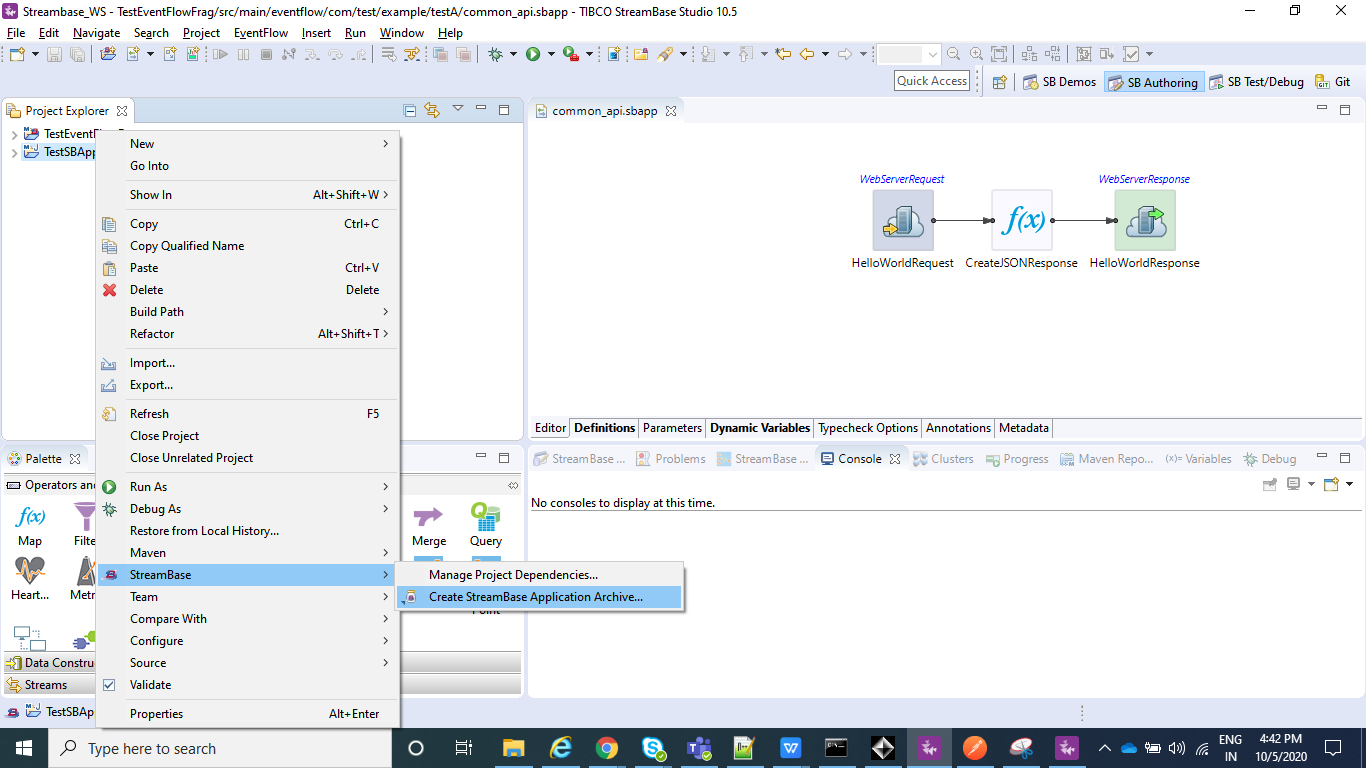
⦁ After the postman is downloaded you can just browse the app and skip the login part and there will be an option which says take me to the app. So can just go with that option and your Postman will be launched.

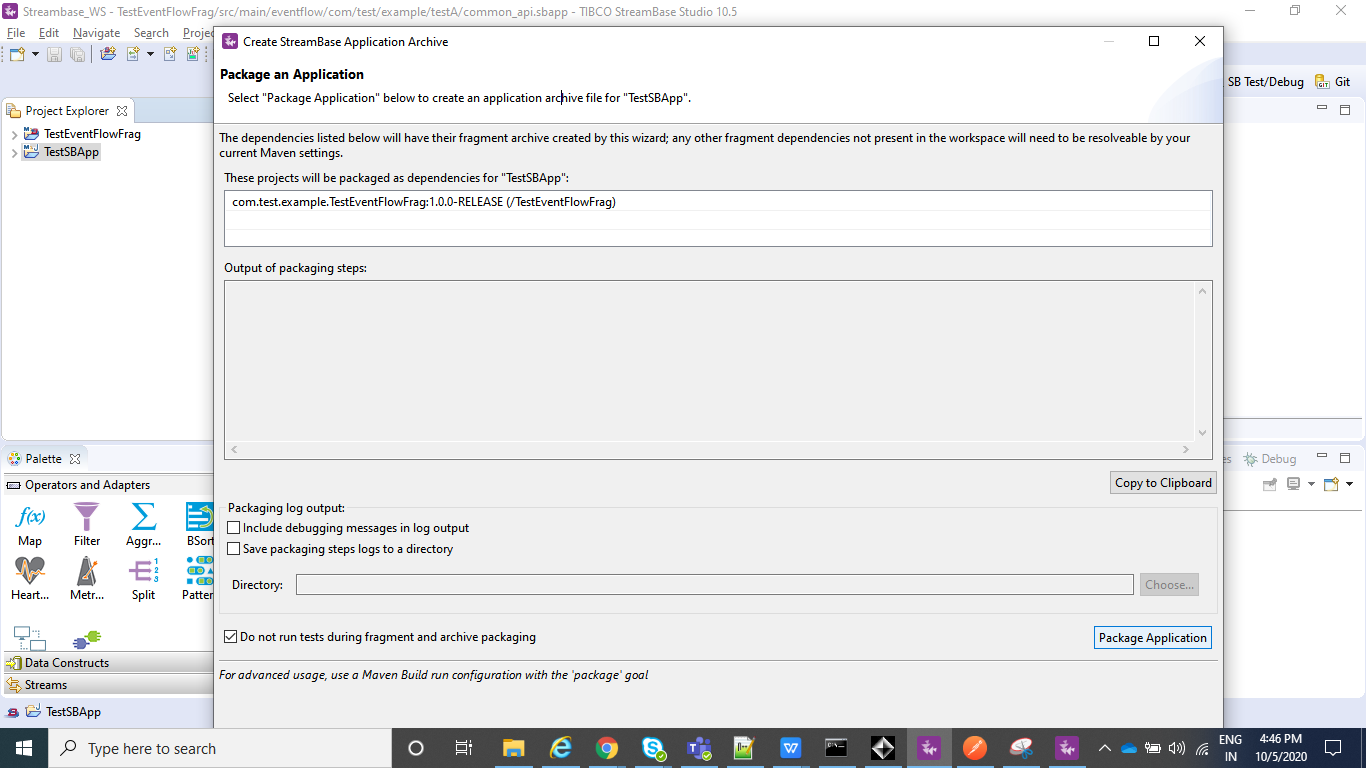
⦁ Now your application is running and same can be tested using postman as: <http://localhost:9191/HelloWorld>



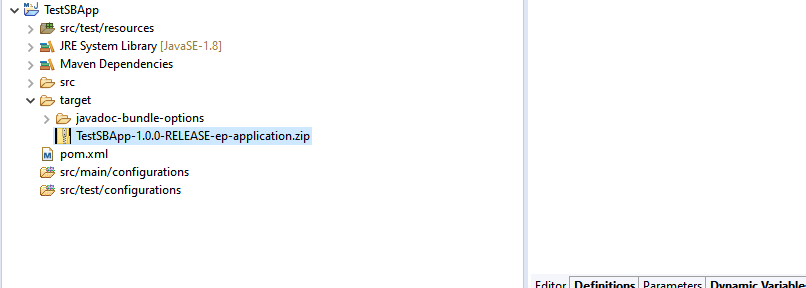
**Create the Application Archive:**

⦁ Right click on the Streambase Application(**TestSBApp** in my case) and than select Create Streambase Application Archive and than select Package Application**.**





⦁ After the above step is completed we can see a .zip file under our Streambase Application under target folder.



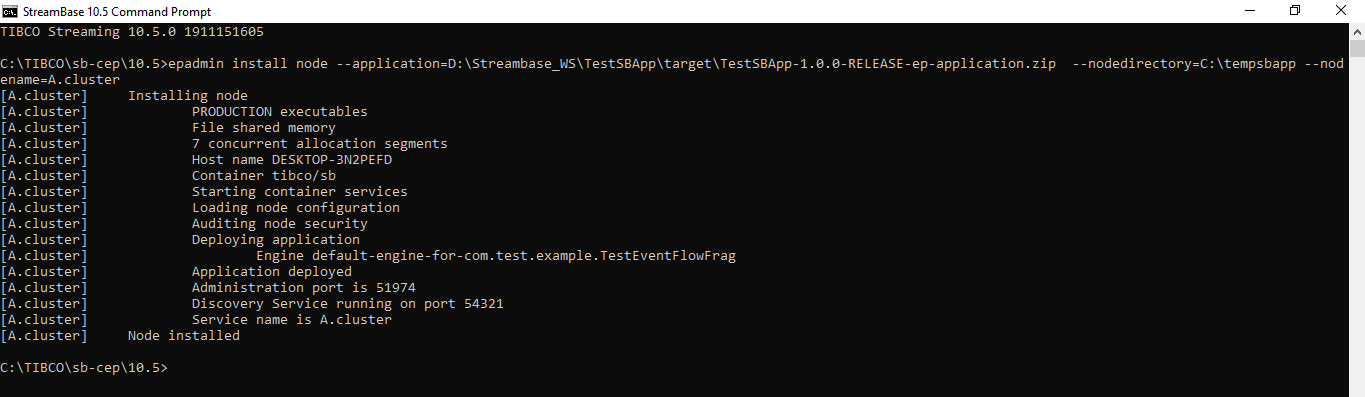
**Deploy and Run the Application:**

⦁ Open the Streambase CommandPrompt to Deploy and Run the application locally as shown in below screenshot.



⦁ Open the Streambase CommandPrompt to Deploy and Run the application locally as shown in below screenshot and run the below command where --**application=D:\Streambase\_WS\TestSBApp\target\TestSBApp-1.0.0-RELEASE-ep-application.zip** is the path of the zip file we just created above,  **--nodedirectory=C:\tempsbapp** is any temp directory created in C drive and **nodename=A.cluster** is the any random name assigned to the node which will be running.

**epadmin install node --application=D:\Streambase\_WS\TestSBApp\target\TestSBApp-1.0.0-RELEASE-ep-application.zip --nodedirectory=C:\tempsbapp --nodename=A.cluster**



⦁ Now after your Node is installed, we can **start**, **stop** or **remove** node using below commands and once you have started the node you can test your application using postman in the same way we did before:

**epadmin servicename=A.cluster start node**

**epadmin servicename=A.cluster stop node**

**epadmin servicename=A.cluster remove node**

