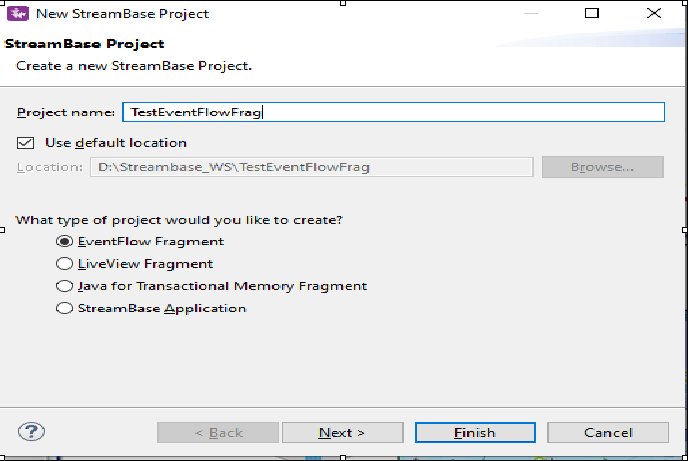
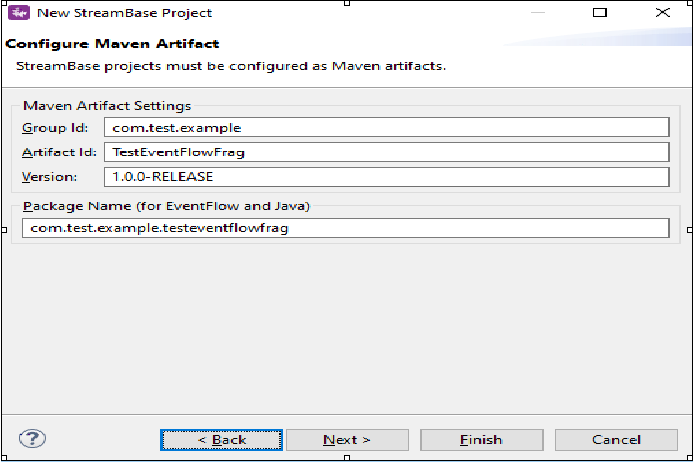
**Create Streambase Studio Project, Create Application Module and Configurations:**

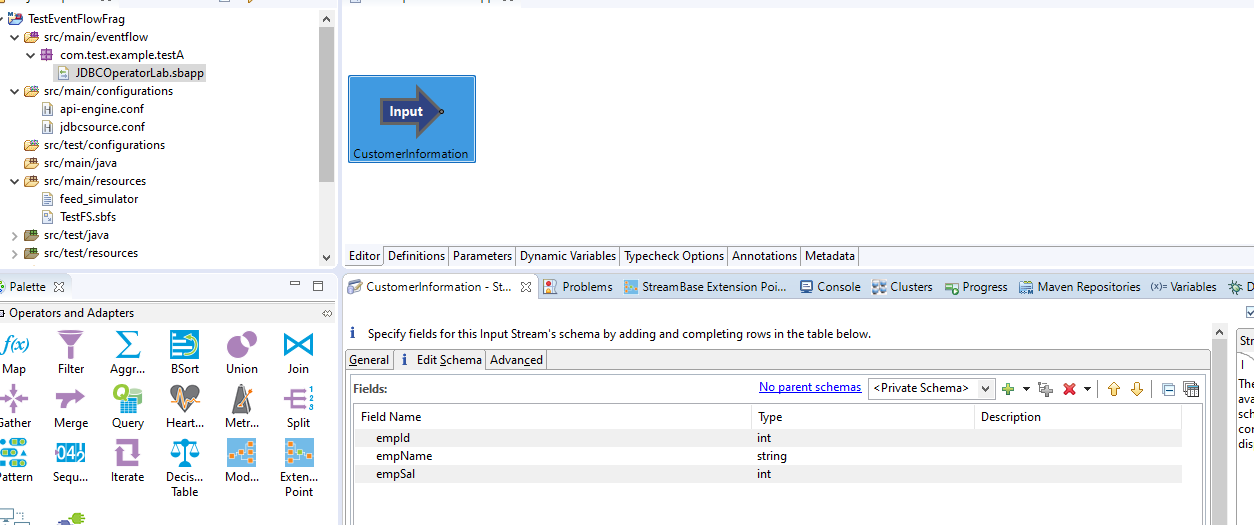
⦁ Create the EventFlow fragment and place all your logic inside this fragment. Select the Radio button as **EventFlow Fragment** as shown in below Screenshot.



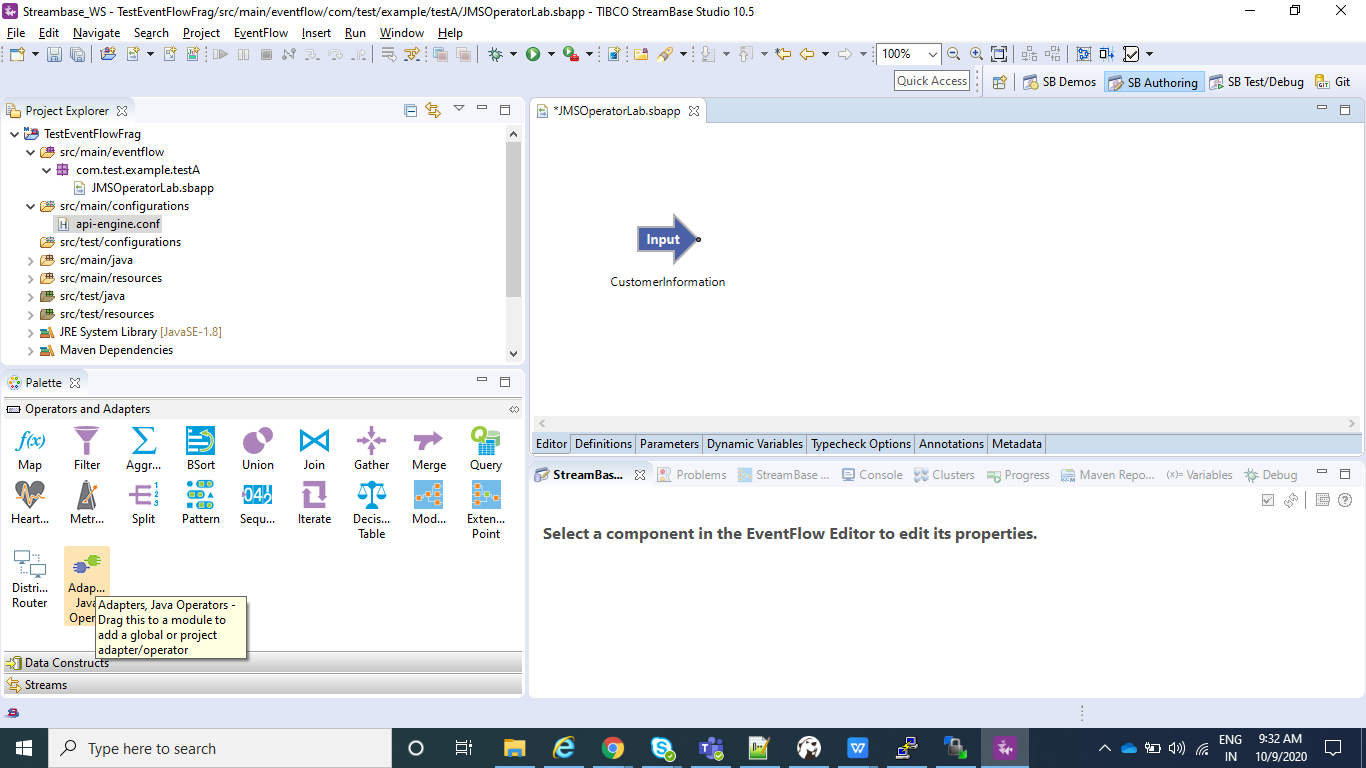
⦁ Click Next and provide the value for **Group Id**, **Artifact Id** and **Version**. Provide the packagename as <<Group Id>>.<<any name>> as shown in below screenshot. Please note down these three parameters and click finish.

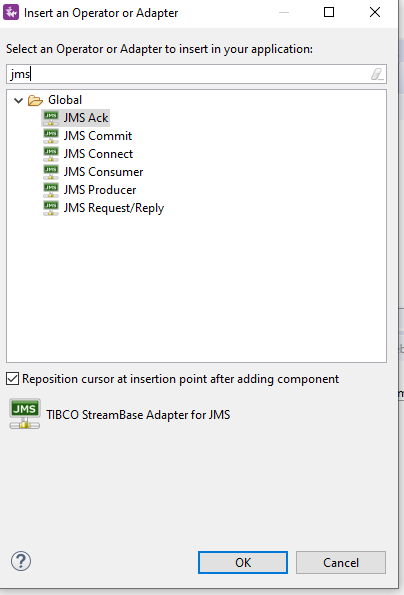


* Create a new event flow fragment and name it as JMSOperatorLab.
* Now go to the folder src/maineventflow and click on JMSOperatorLab.sbapp
* From the palette view click on Input Stream and Drag it into the canvas.
* Name the Input Stream as Customer Information.
* In the Edit Schema tab Give the Field Name as empId,empName,empSal and Select the type as int,string,int.

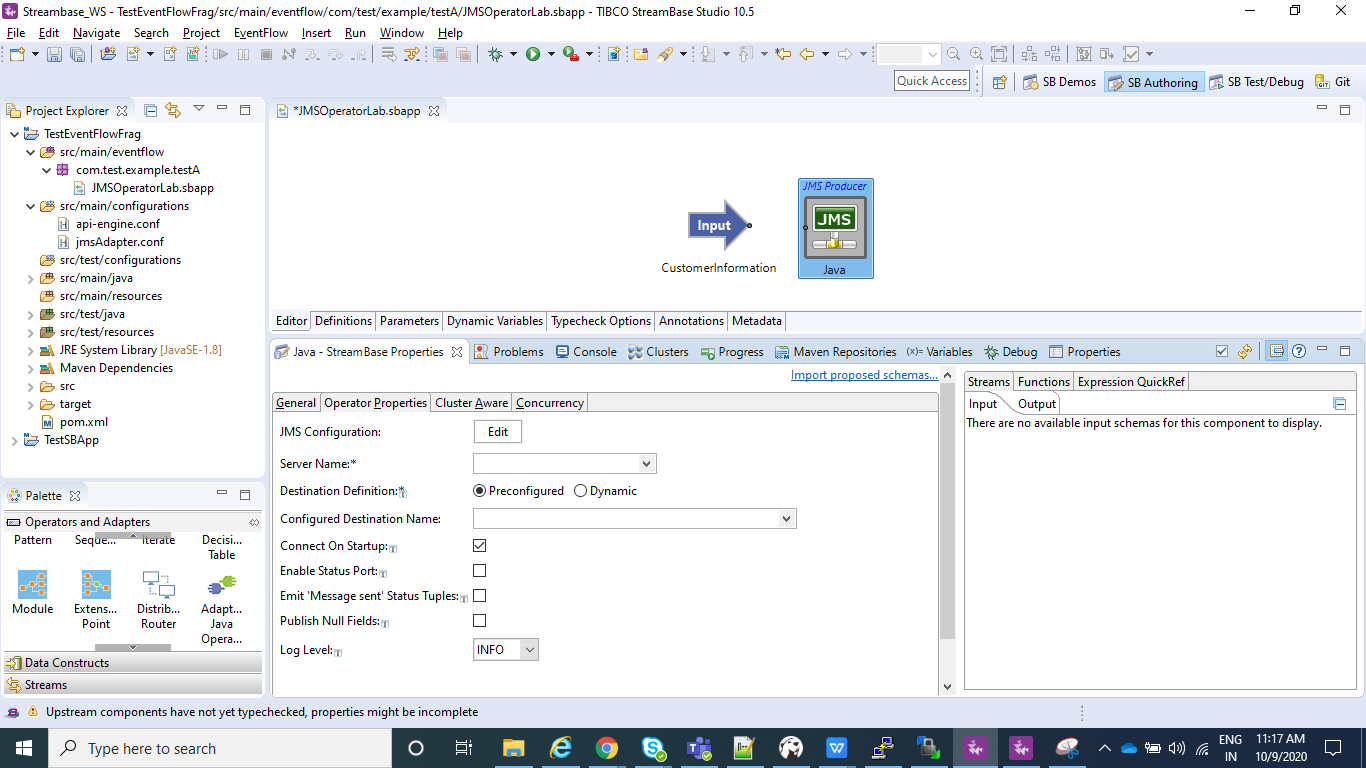


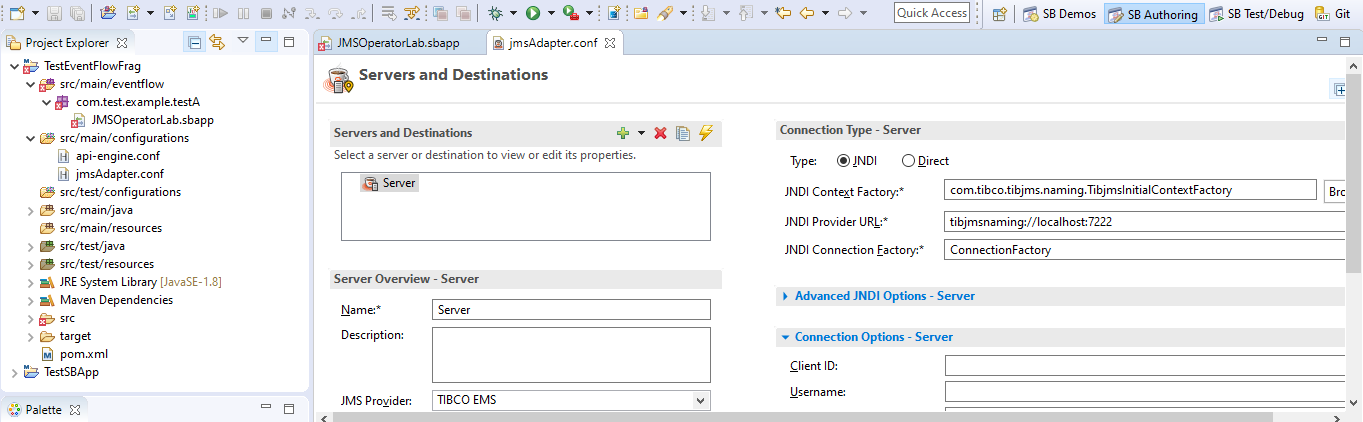
* Now Drag the Adapter,Java Operators to the canvas and search for jms.

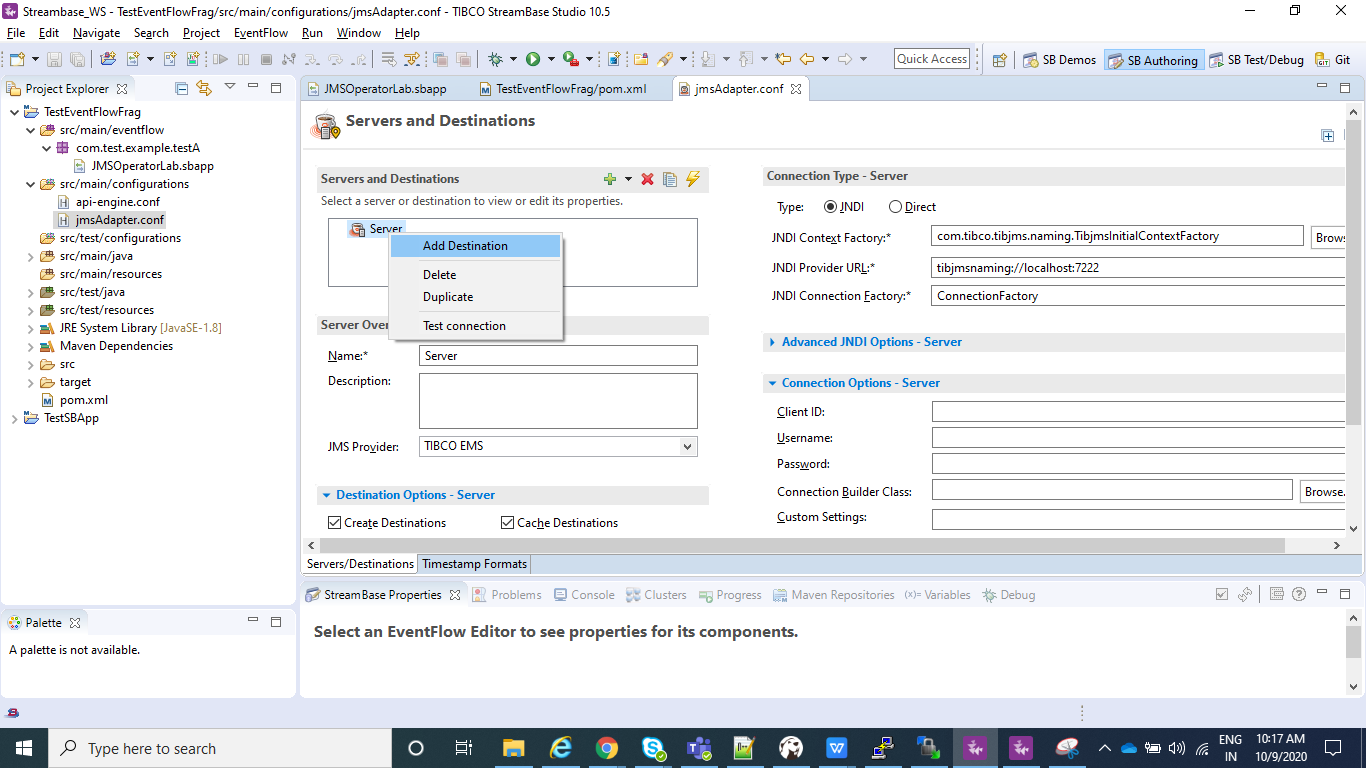




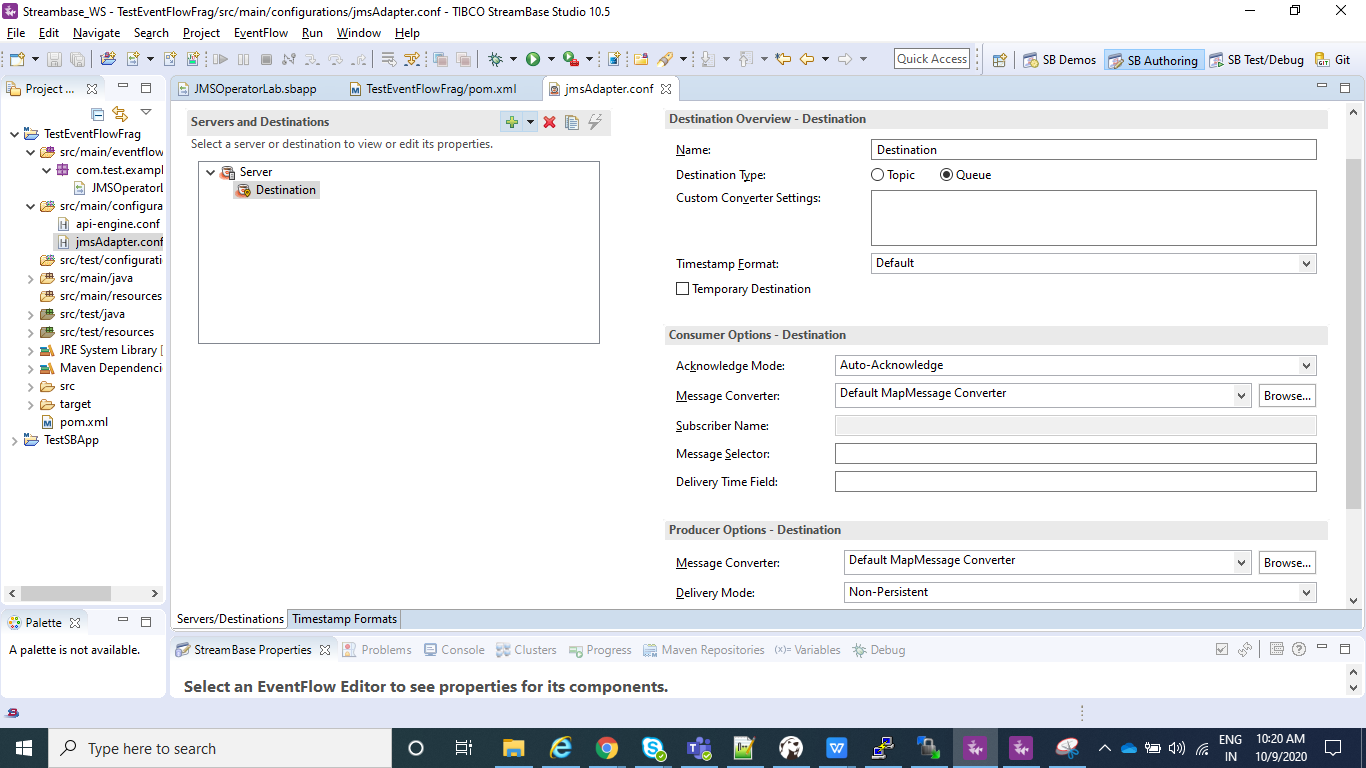
* Now we will first select jms producer to produce the message and than later we will have jms consumer consuming the same messages.
* Now this JMS producer or consumer will need the JMS configuration that we need to do.For this click on Edit under Operator Properties and you will see jmsAdapter.conf is created under src\main\configurations.



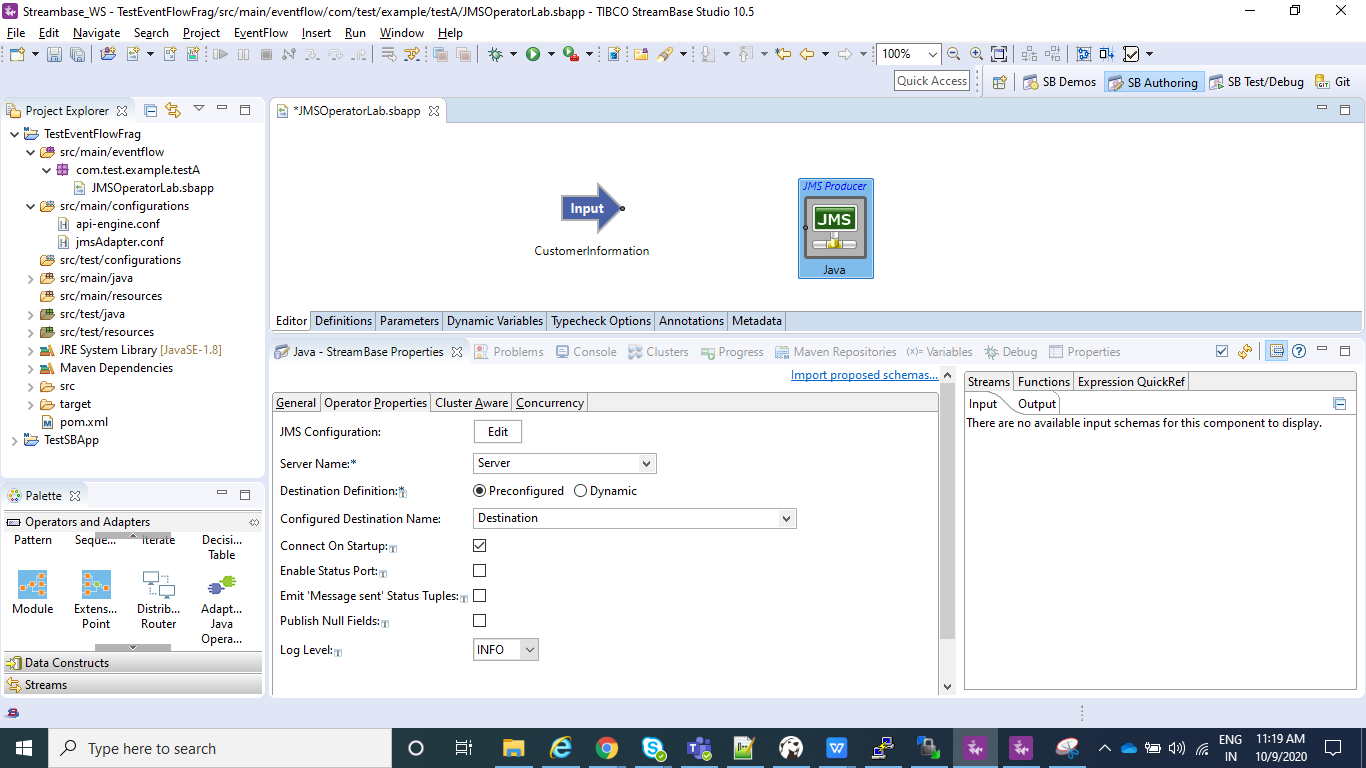




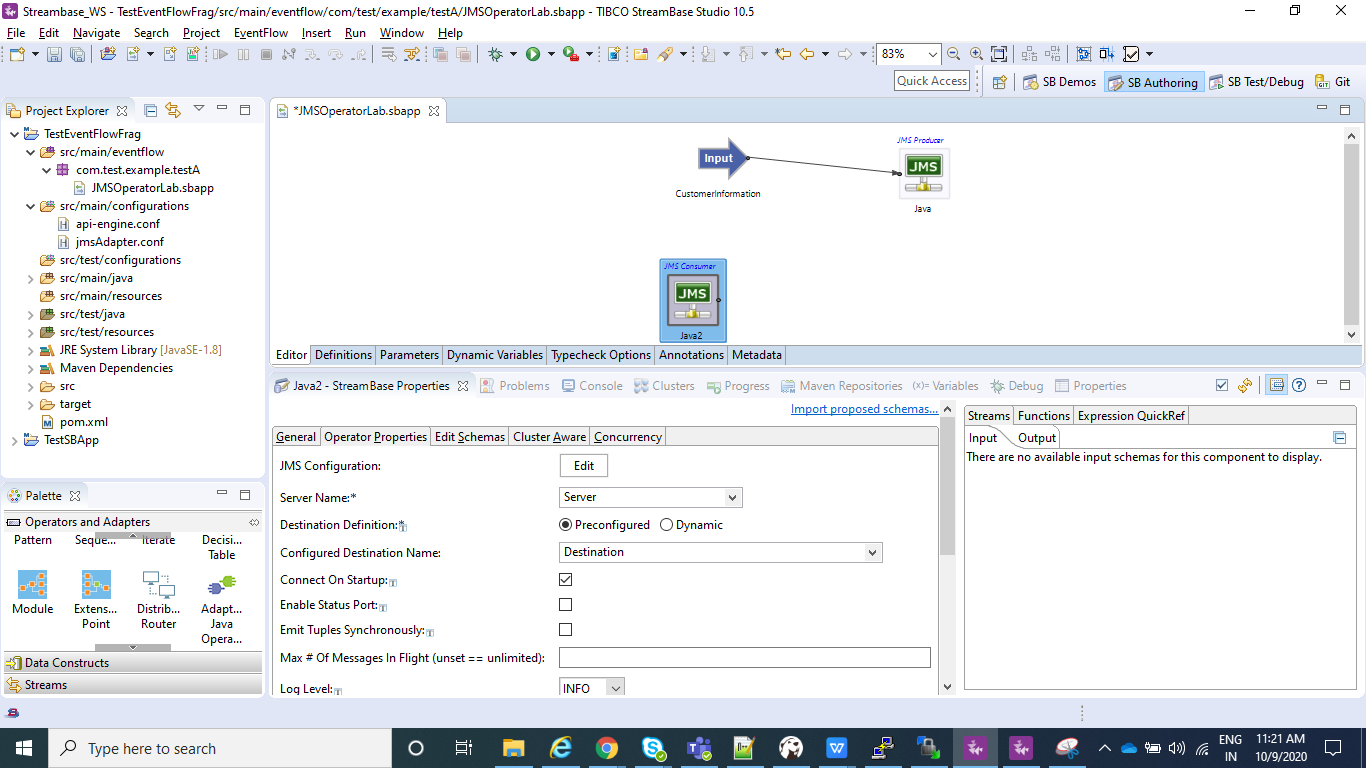
* As we can see in the above screenshot the connection type- server is pointing to localhost. You can change the server configuration according to your need. Also we will be adding destination which will be used for sending and receiving messages.
* We would select our destination as Queue and Message Converter as Default MapMessage Converter for both Producer and Consumer.



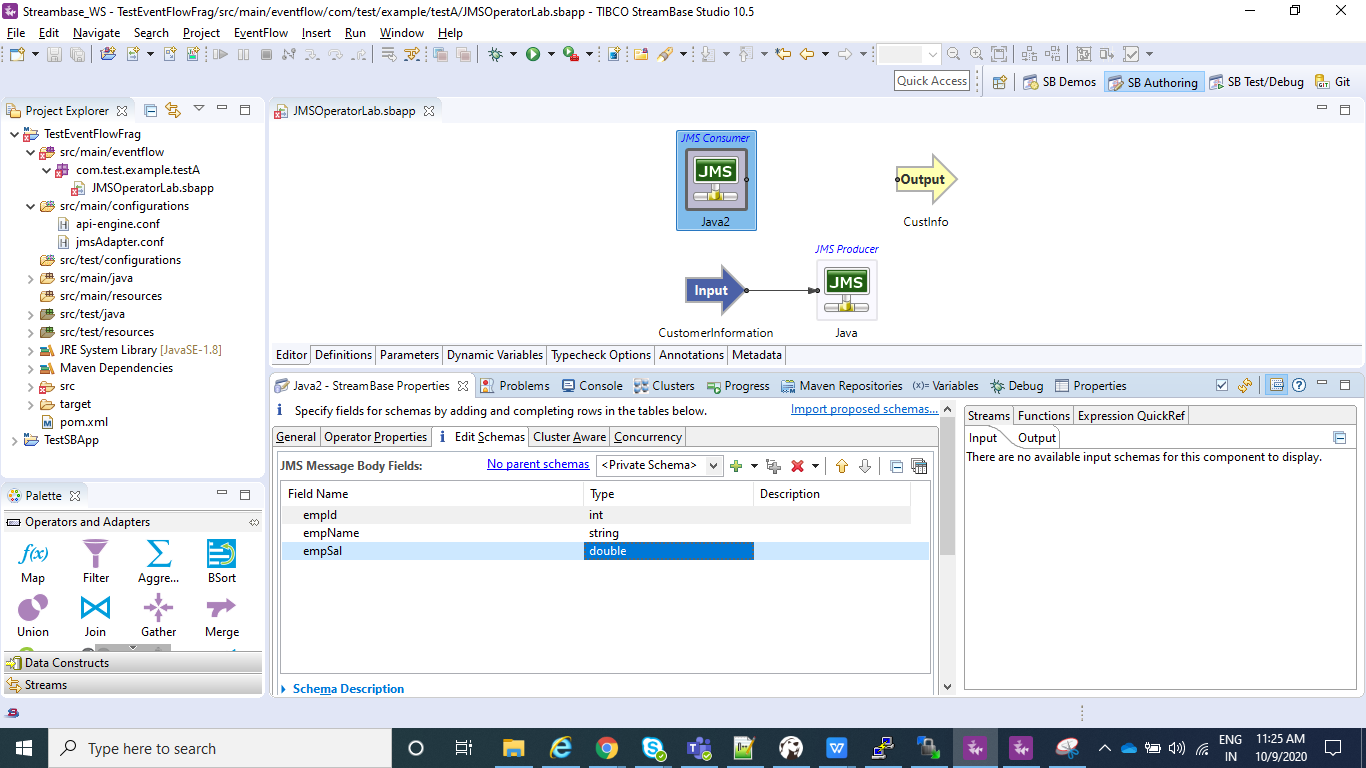
* Now after the above step you will be able to see Server under Server Name and destination under Configured Destination Name.



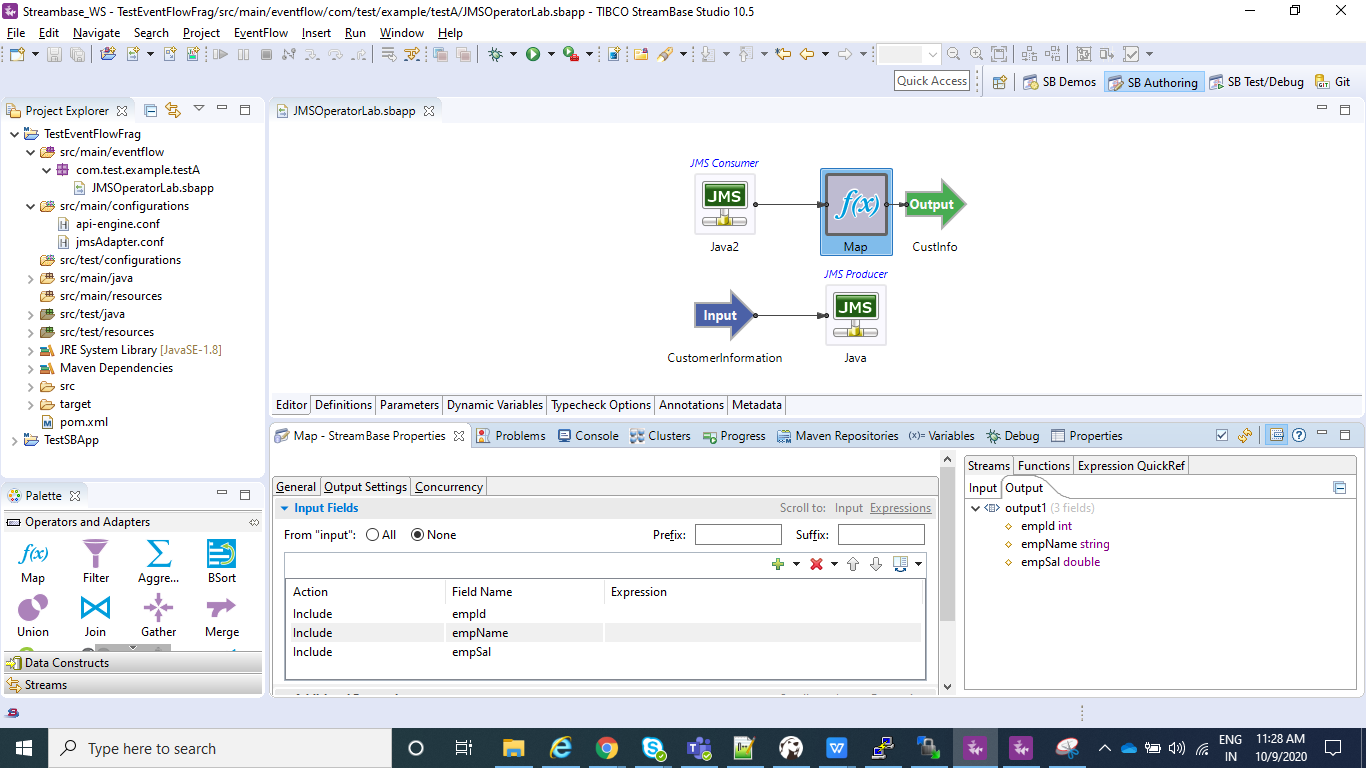
* Now whatever message you send via your Input Stream(CustomerInformation), the same is send across the destination configured.So now we need consumer to consume those messages, so now we will select JMS Consumer.
* Now for this we don’t need to click on edit as the jmsconfig.conf is already created. So we just need to select the Server name and Configured Destination name in JMS consumer.



* Now we would just Drag and drop output stream so that data we just received can be seen in output. We will name the output stream as CustInfo. Also in the edit schema of JMS Consumer we will define all the fields that we are expecting.So we will define empId, empName and empSal.



* Now we will add only those parameters that we wish to see in output with the help of mapper and connecting that to the output stream(CustInfo).For this in output setting of mapper we will select From input as None and adding only the required fields as shown below.



* In the pom.xml of the project we need to make sure that following dependency is present. If not than we need to add this dependencies in our pom.xml as shown below and than need to do clean install.

<dependency>

<groupId>com.tibco.ep.sb.adapter</groupId>

<artifactId>jms</artifactId>

<version>10.4.3</version>

</dependency>

<dependency>

<groupId>javax.jms</groupId>

<artifactId>jms</artifactId>

<version>2.0</version>

</dependency>

<dependency>

<groupId>com.tibco</groupId>

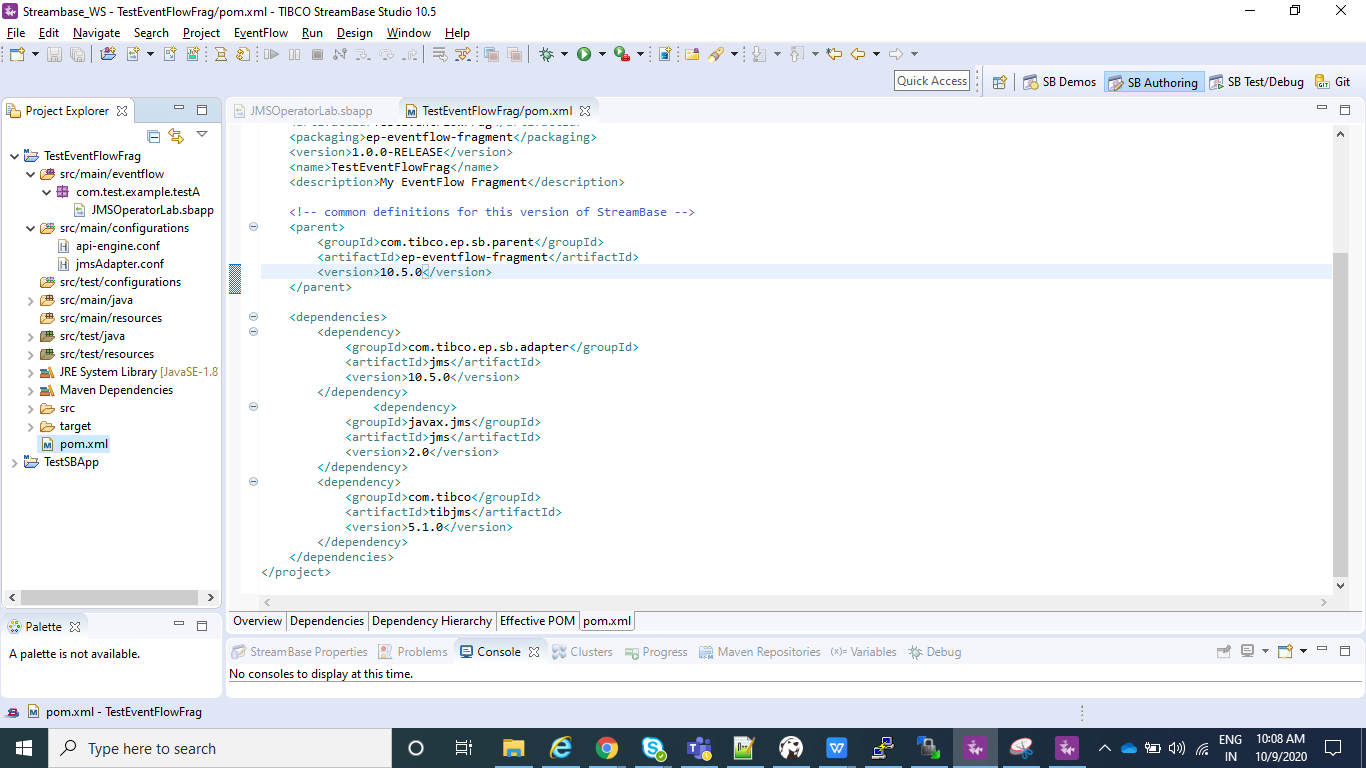
<artifactId>tibjms</artifactId>

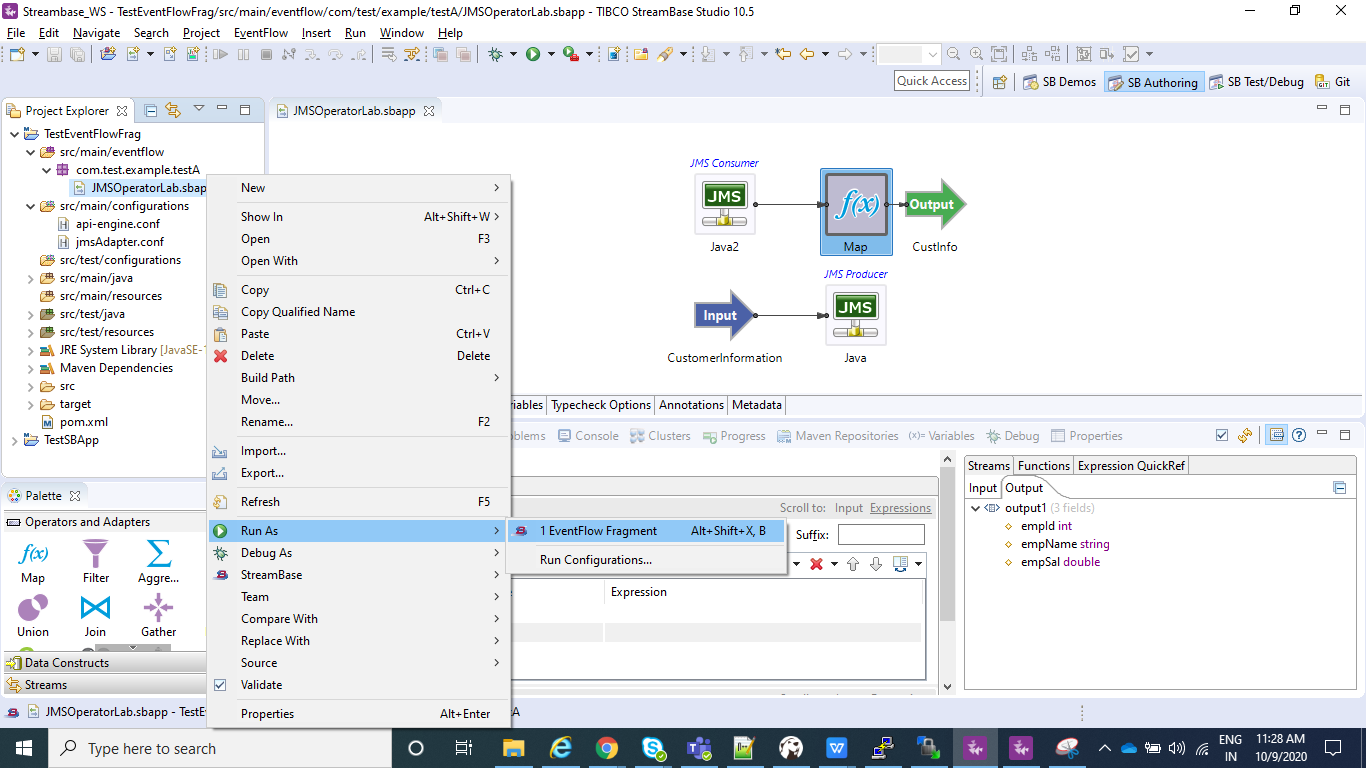
<version>5.1.0</version>

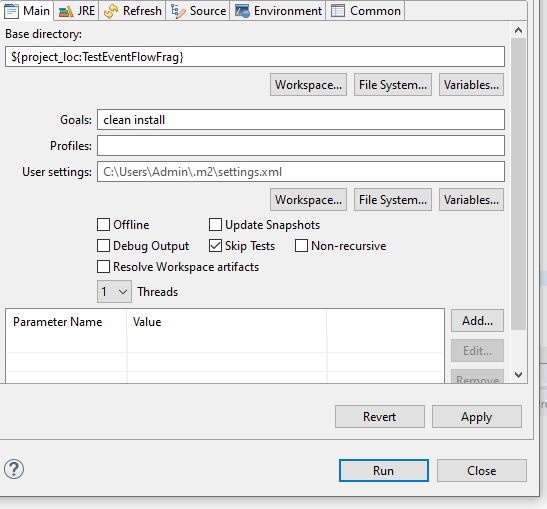
</dependency>

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

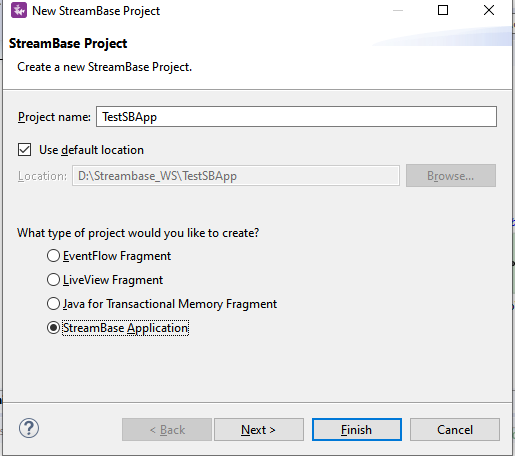
⦁ After that In the **Goals** field, enter **clean install** and Select the **Skip Tests** check box. Click Apply and Run.



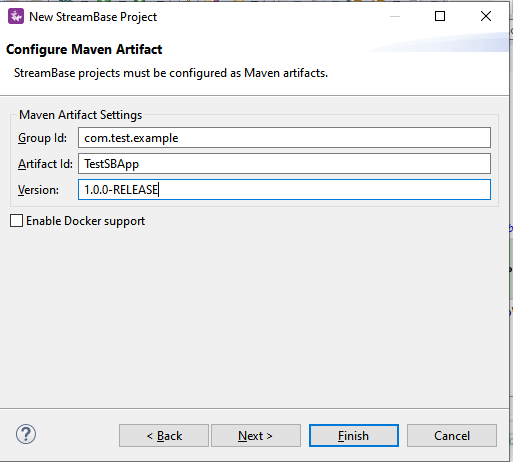




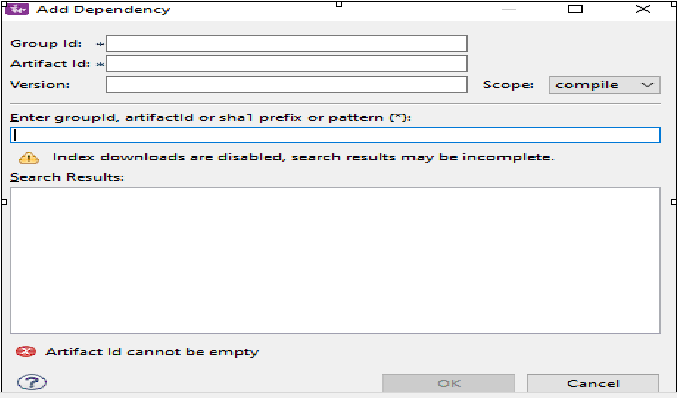
⦁ Now create a new Streambase Project. Select Radio button as **StreamBase Application**.

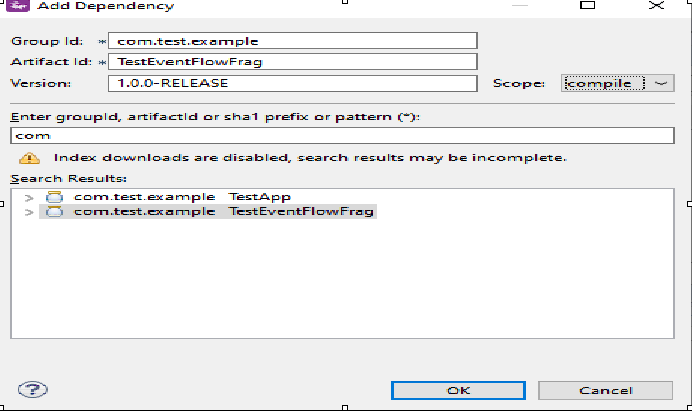


⦁ Now click next and make sure you provide the same **Group Id** and **Version** that you specified for the EventFlow Fragment created Earlier and click **Finish**.



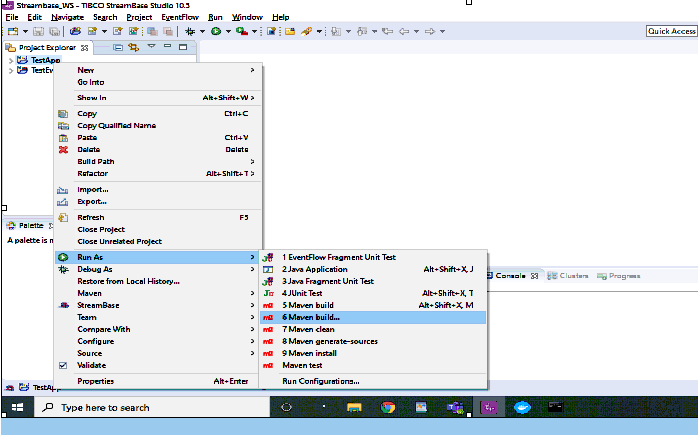
⦁ Add the EventFlow Fragment Project depedency to our main Streambase Application. Right click on Streambase Application Project **(TestSBApp in my case) --> StreamBase>Manage Project Dependencies**. Add the depedency for the EventFlow Fragment and click ok.

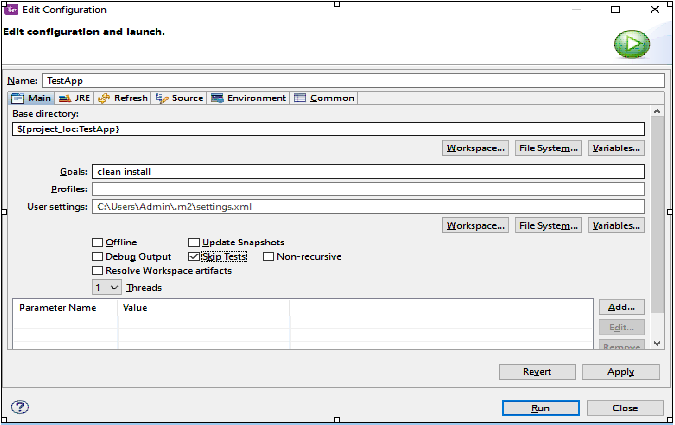


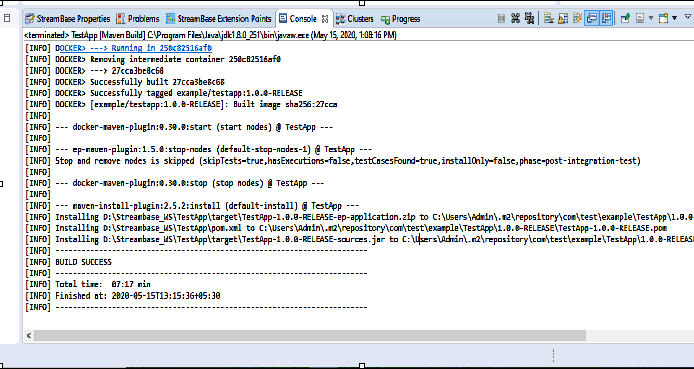


⦁ Now In the Project Explorer view, select and right-click the Streambase Application Project **(TestSBApp in my case)--> Run As>Maven Build...**

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

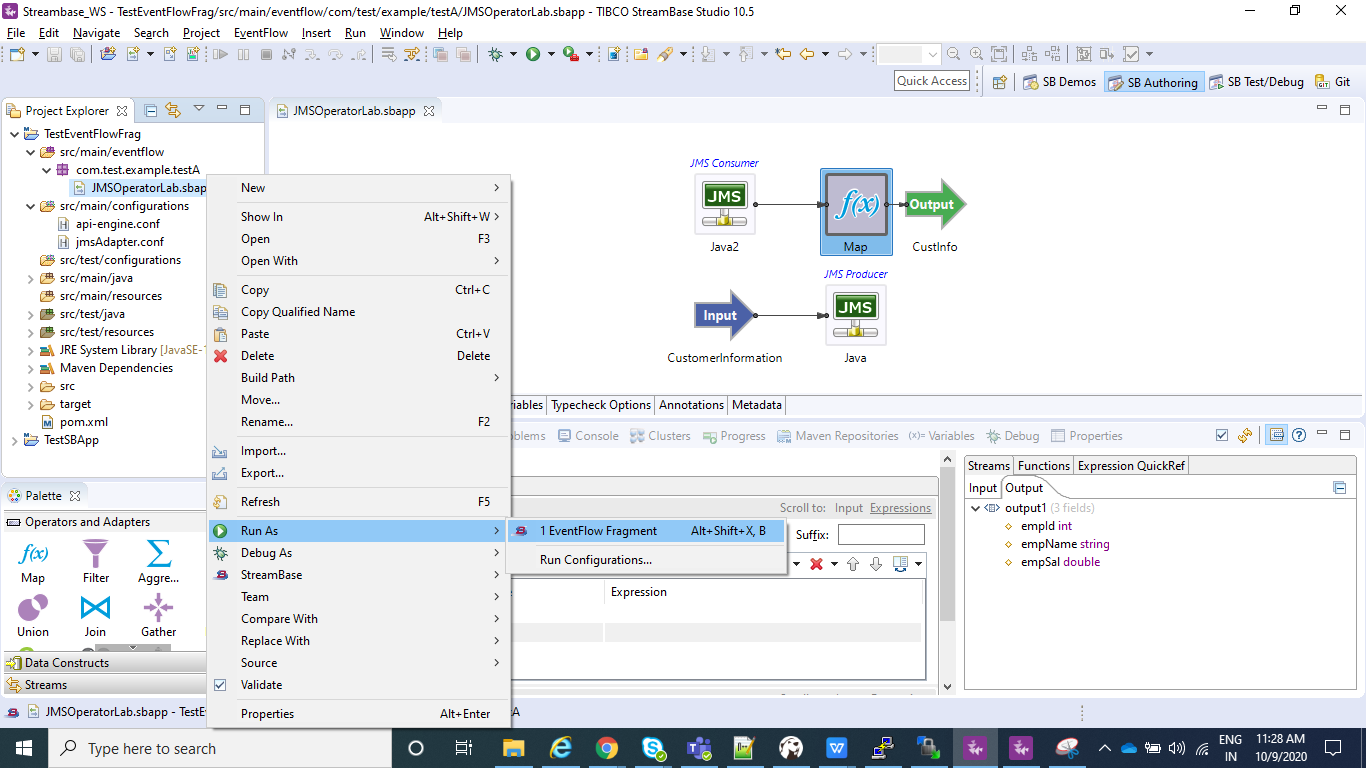




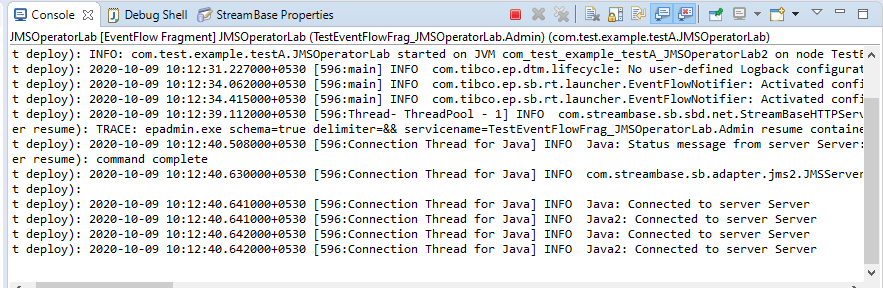
⦁ After the above step check the console and it should be showing BUILD SUCCESS message.

**Run the application in Studio:**

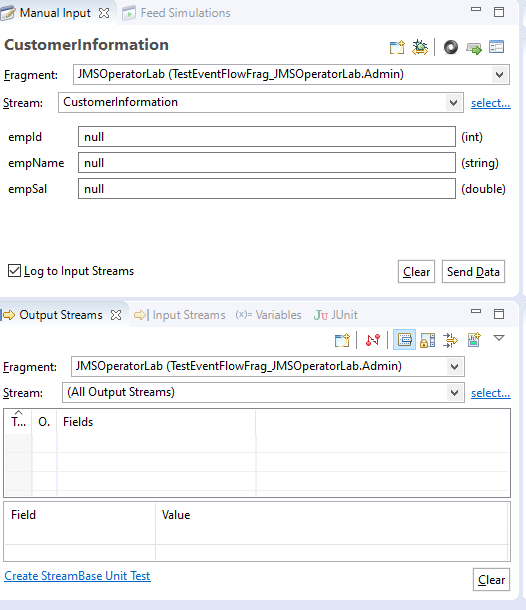
⦁ Right click on your .sbapp and select Run as EventFlowFragment as shown in below screenshot:

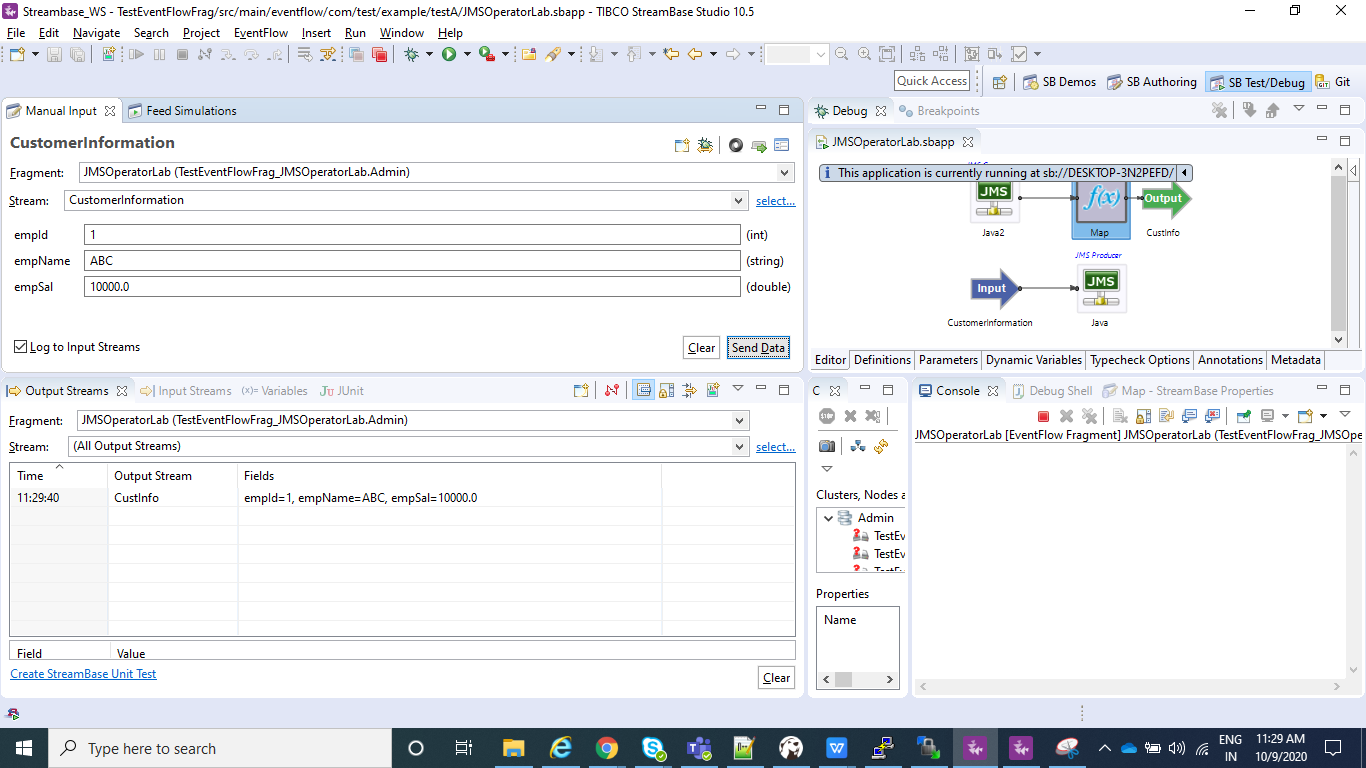


⦁ After you application is running you can see to the console logs as shown below:



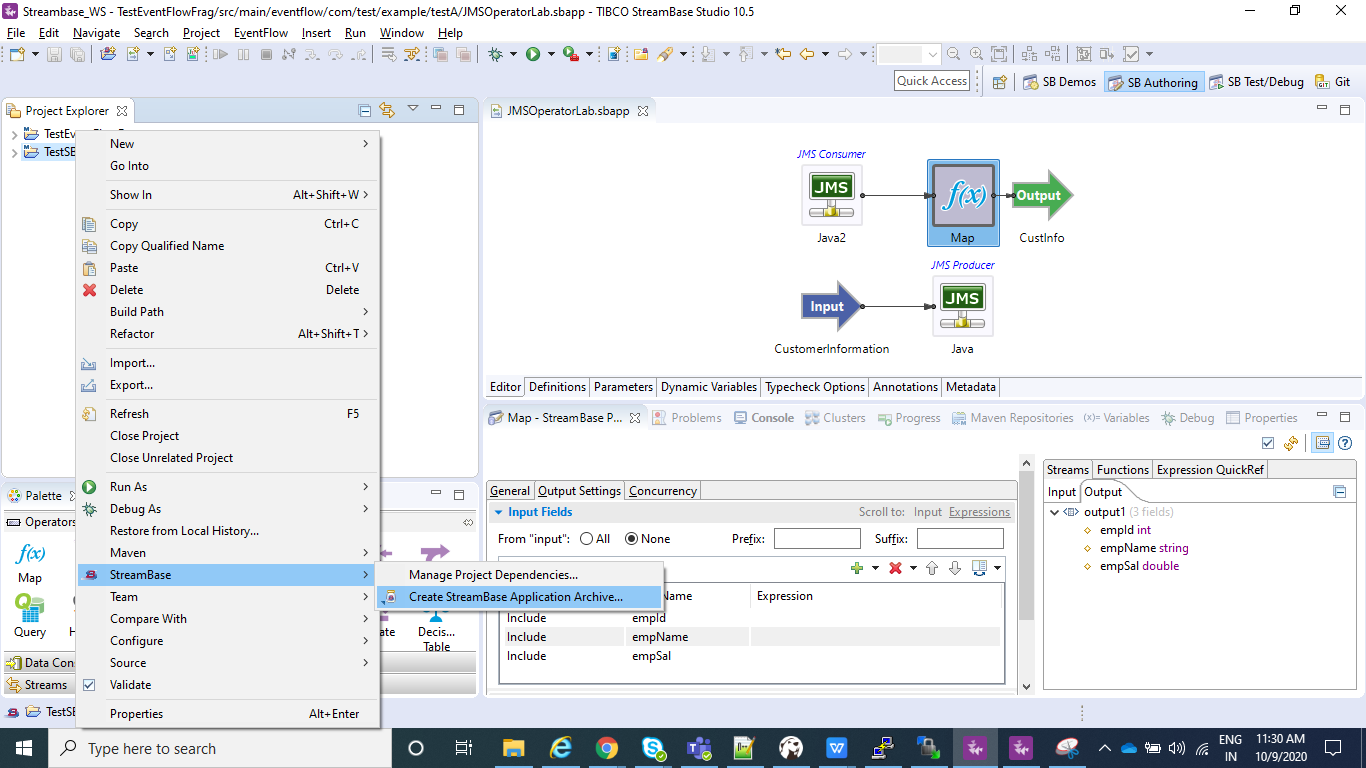
⦁ Now your application is running and same can be tested using Manual Input as shown in below screenshot.Provide the value for empId, empName, empSal and the same than can be seen in output stream.

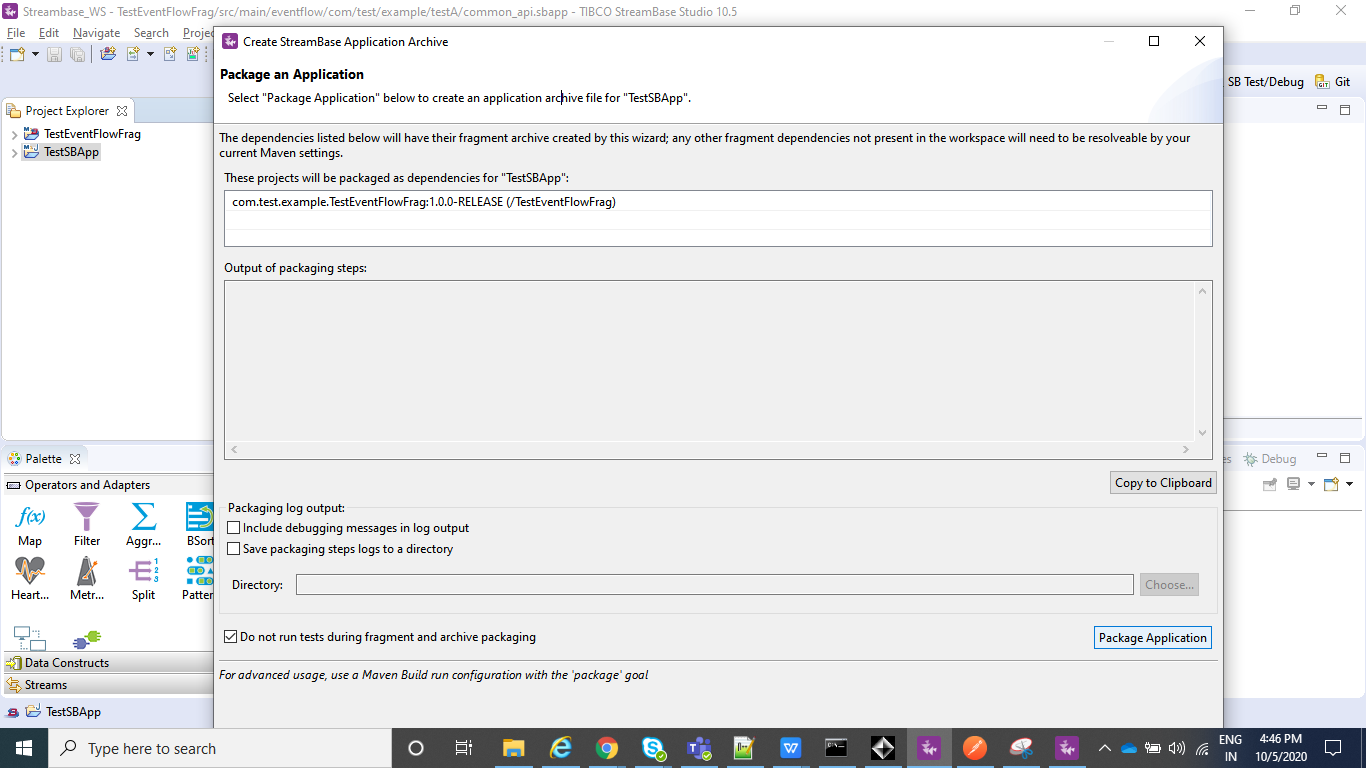




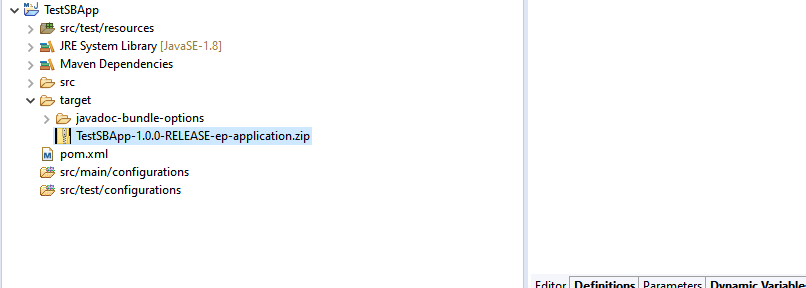
**Create the Application Archive:**

⦁ 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.



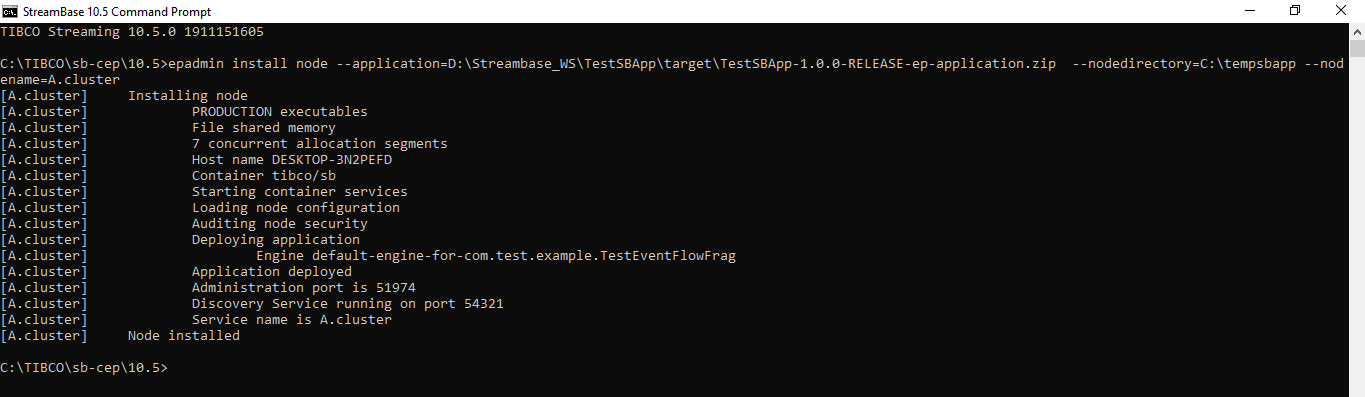
**Deploy and Run the Application:**

⦁ 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**

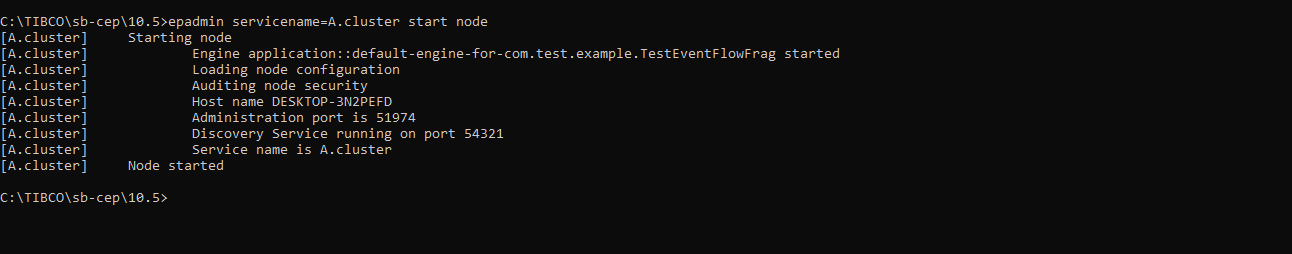


⦁ Now after your Node is installed, we can **start**, **stop** or **remove** node using below commands and once you have started the node you can test your application using postman in the same way we did before:

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

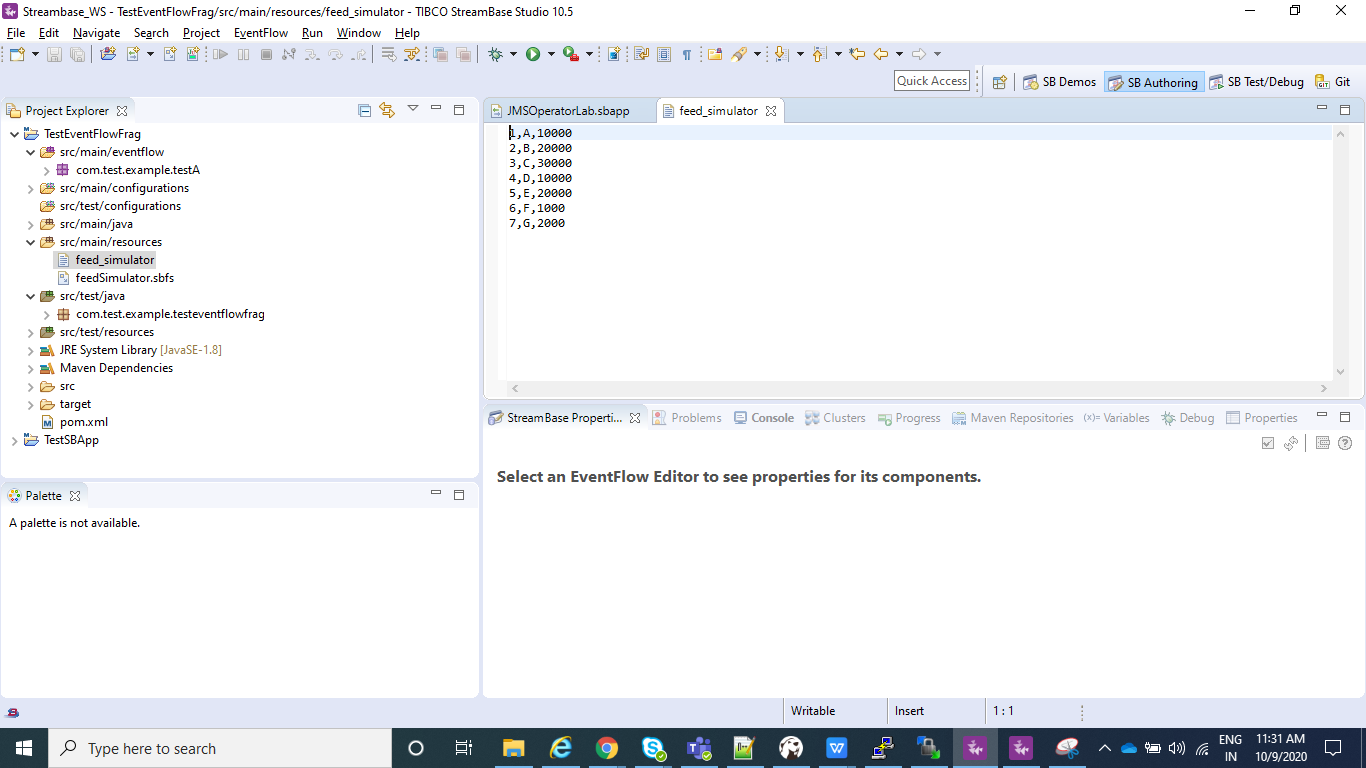
**epadmin servicename=A.cluster stop node**

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

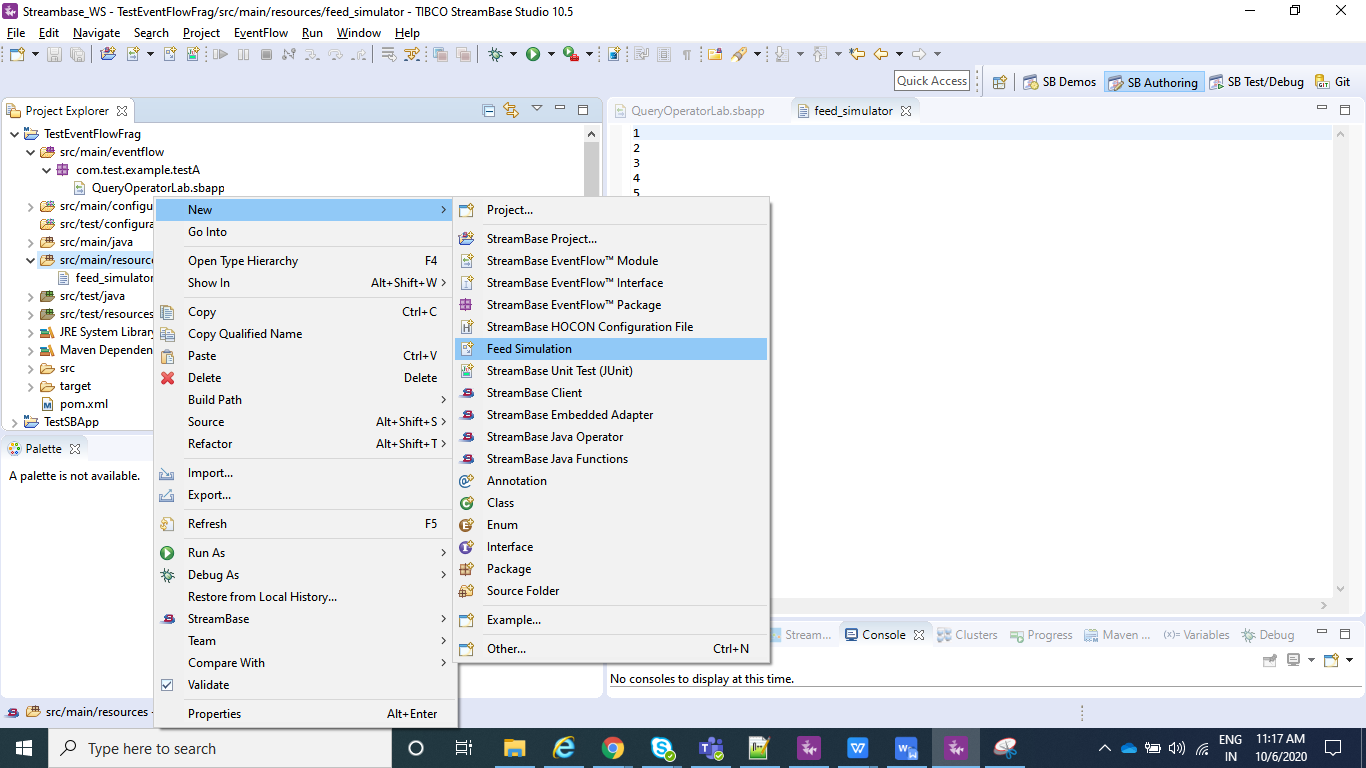


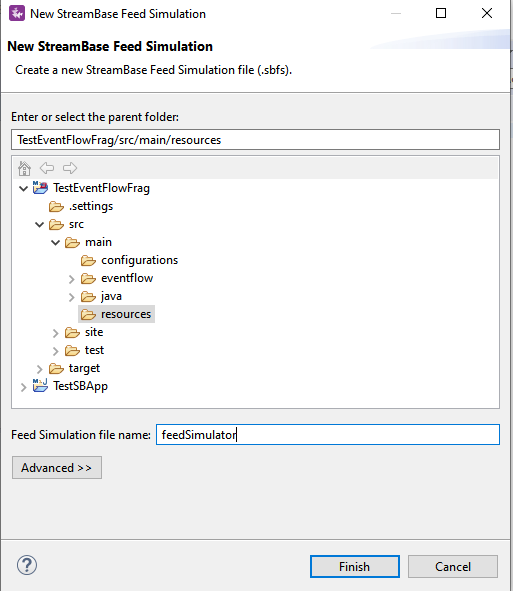
**Test the application (Using Feed Simulator file from command line):**

* Import a feed\_simulator.txt file with below entries under src/main/resources.

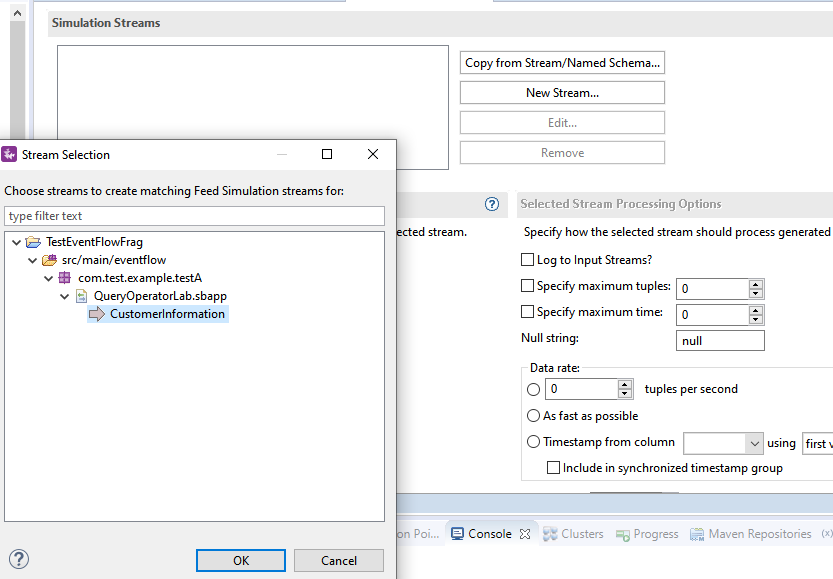


* To Configure feed simulator please follow below screenshots :

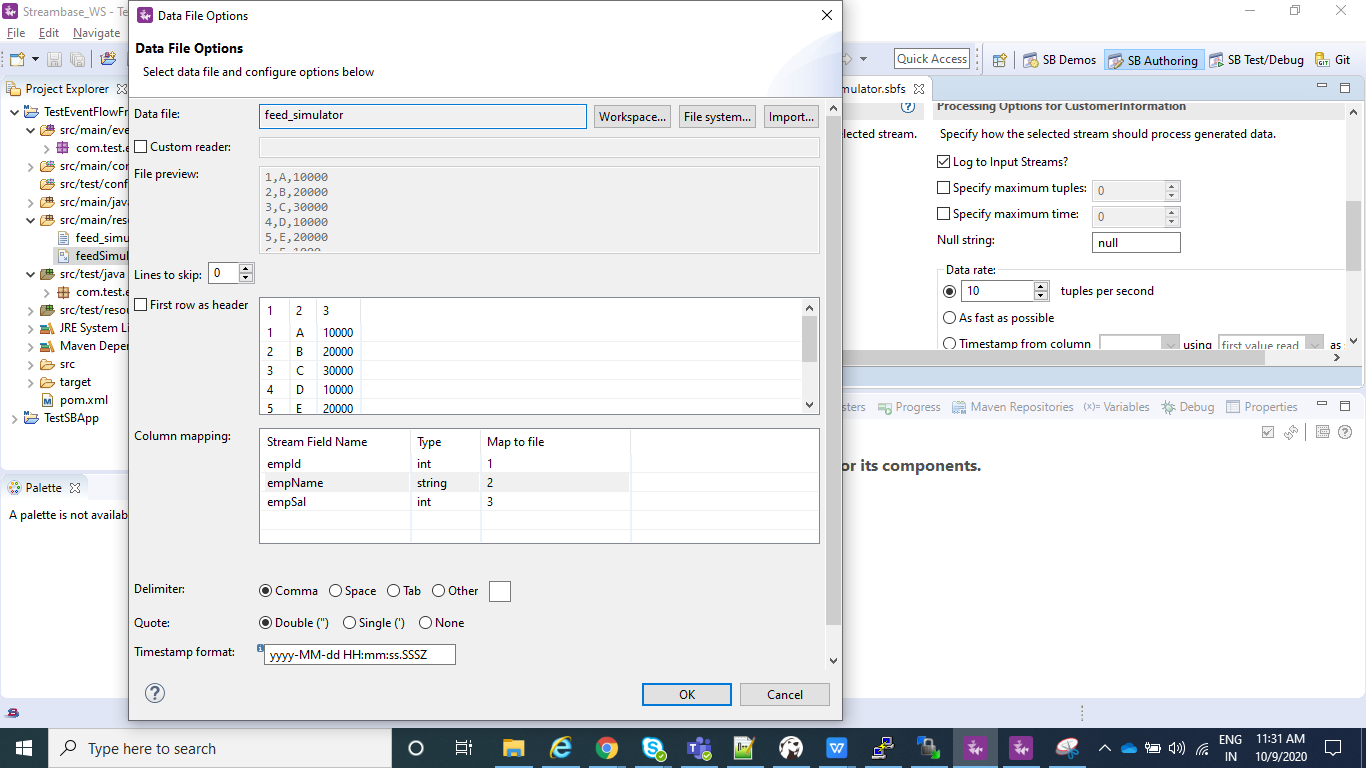




* Click on Copy from Stream/Named Schema and go to your Input Stream(CustomerInformation in my case) and click OK.



* Now select Data File option and select the file you just created , i.e, feed\_simulator.txt file where you have your data loaded.



* Create a configuration file which will contains information on which of your .sbapp need to run when an streambase application is deployed.Sample for the same is :

name = "sample-StreamBaseEngine-document"

type = "com.tibco.ep.streambase.configuration.sbengine"

version = "1.0.0"

configuration = {

StreamBaseEngine = {

jvmArgs = [

"-Xmx8g"

"-Xms512m"

"-XX:+UseG1GC"

"-XX:MaxGCPauseMillis=500"

"-XX:ConcGCThreads=1"

]

}

EventFlowDeployment = {

modules = [

{

moduleName="com.test.example.testA.JMSOperatorLab"

containerName="JMSOperatorLab"

}

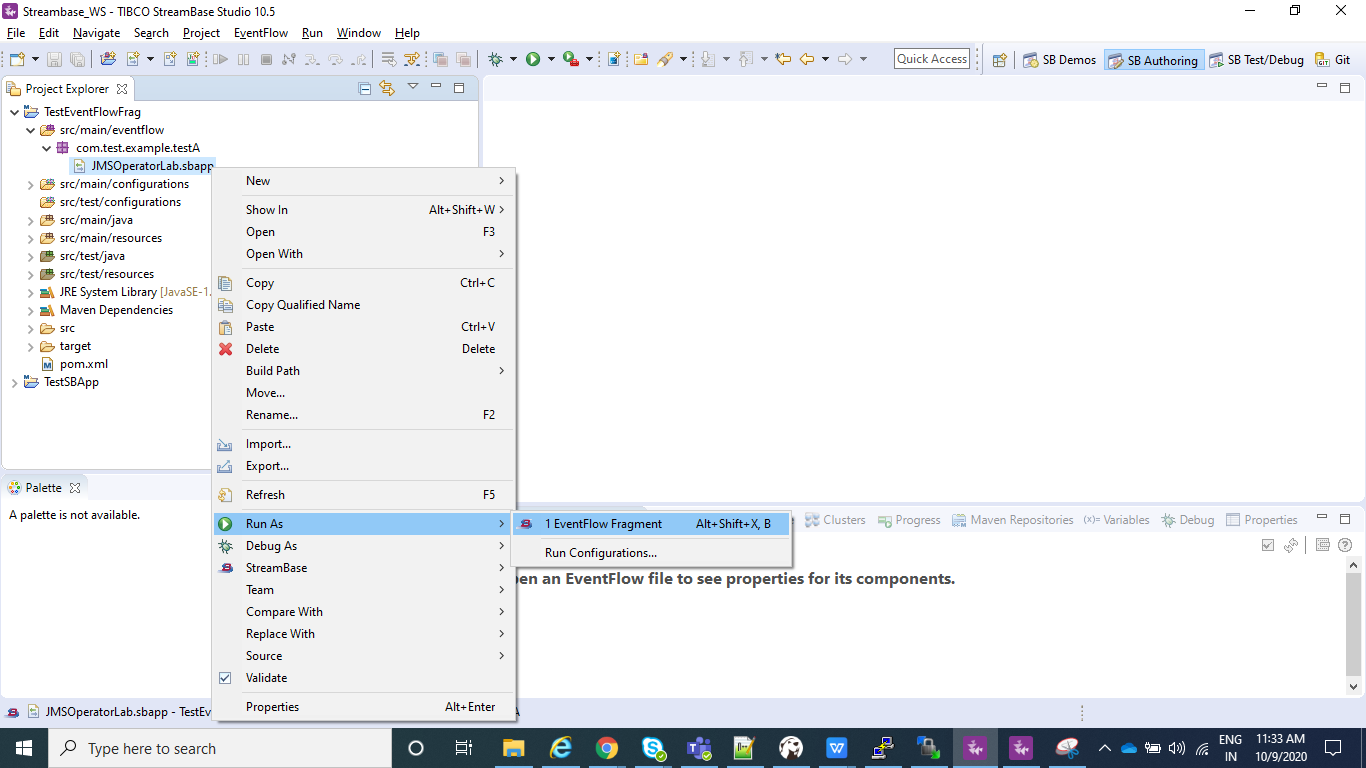
] }

}

Note : moduleName="com.test.example.testA.JMSOperatorLab" is nothing but my .sbapp which I wish to run when my application is deployed and containerName is just any name that you wish to give.



* Now make sure you do a clean install as explained above for the EventFlowFragment(TestEventFlowFrag in my case) and than for your Streambase Application(TestSBApp in my case) and than create Archive and Deploy the node as explained in above steps.
* In StreamBase command prompt navigate to TesteventFlowFrag project workspace location where your **feedSimulator.sbfs** is located and use **sbfeedsim** utility command and enter as below as **sbfeedsim feedSimulator.sbfs** and we can see Data from feed simulator.
* Also using the Streambase Studio we can run Feed Simulation.Just Right click on your .sbapp and run as eventFlowFragment.



* After your application is running select Feed Simulations instead of manual input as shown in below screenshot and create a new Feed Simulation and provide any name and after it is created configure in the same way we configured in above steps for feedSimulator.sbfs and save and than just do Run.

