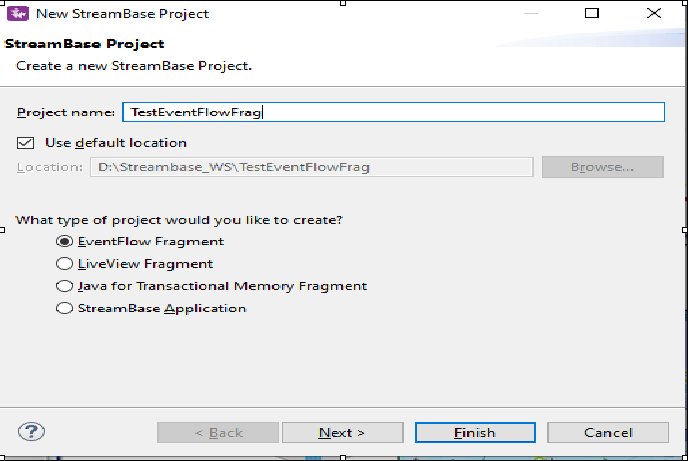
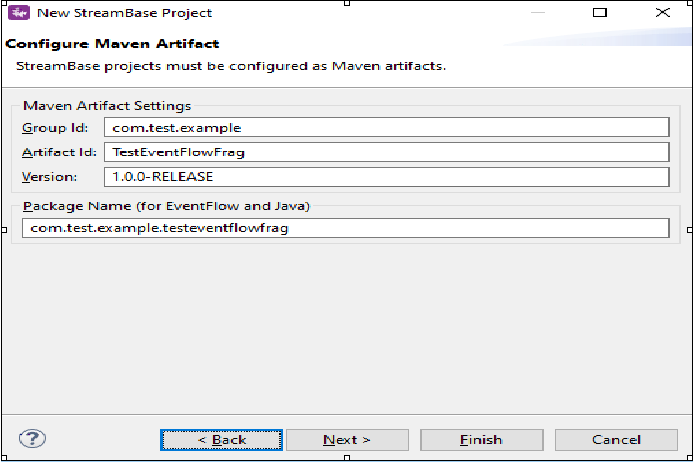
**Create Streambase Studio Project, Create Application Module and Configurations:**

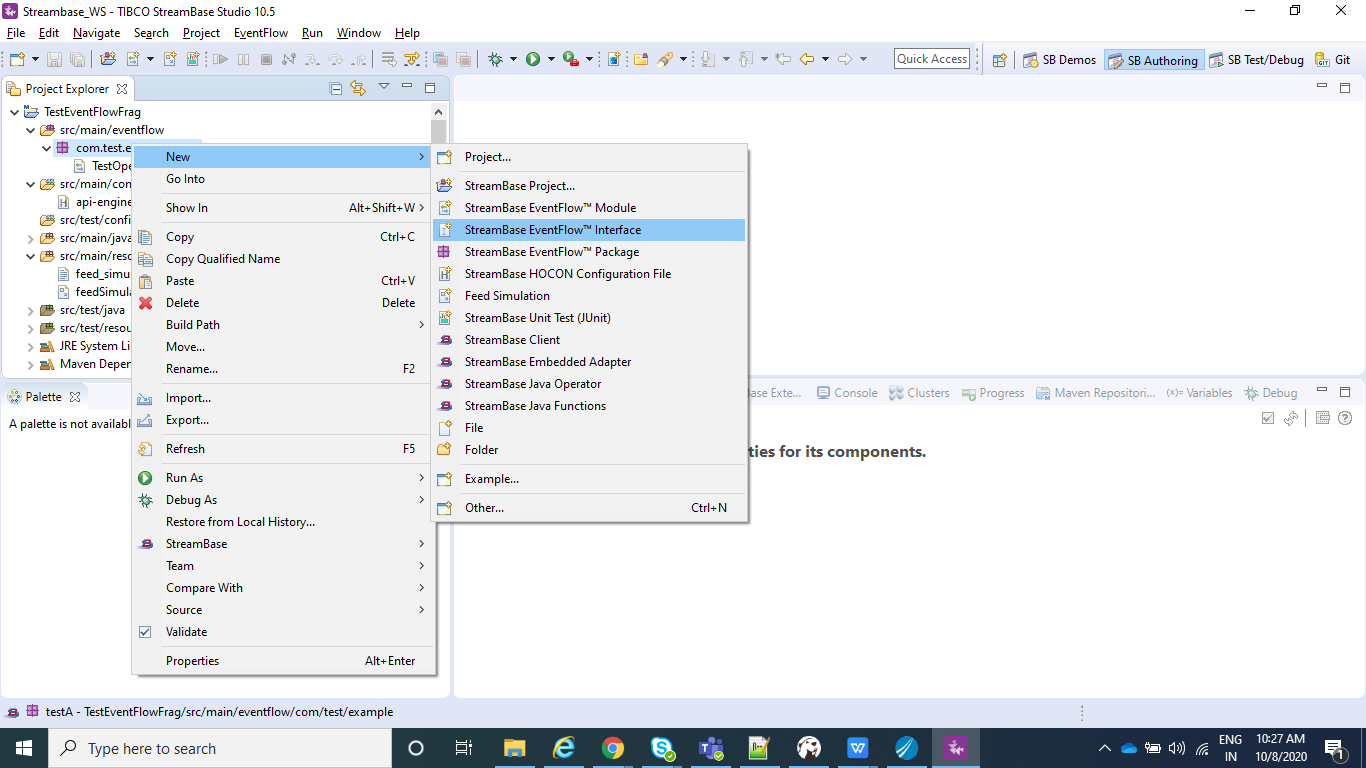
⦁ Create the EventFlow fragment and place all your logic inside this fragment. Select the Radio button as **EventFlow Fragment** as shown in below Screenshot.



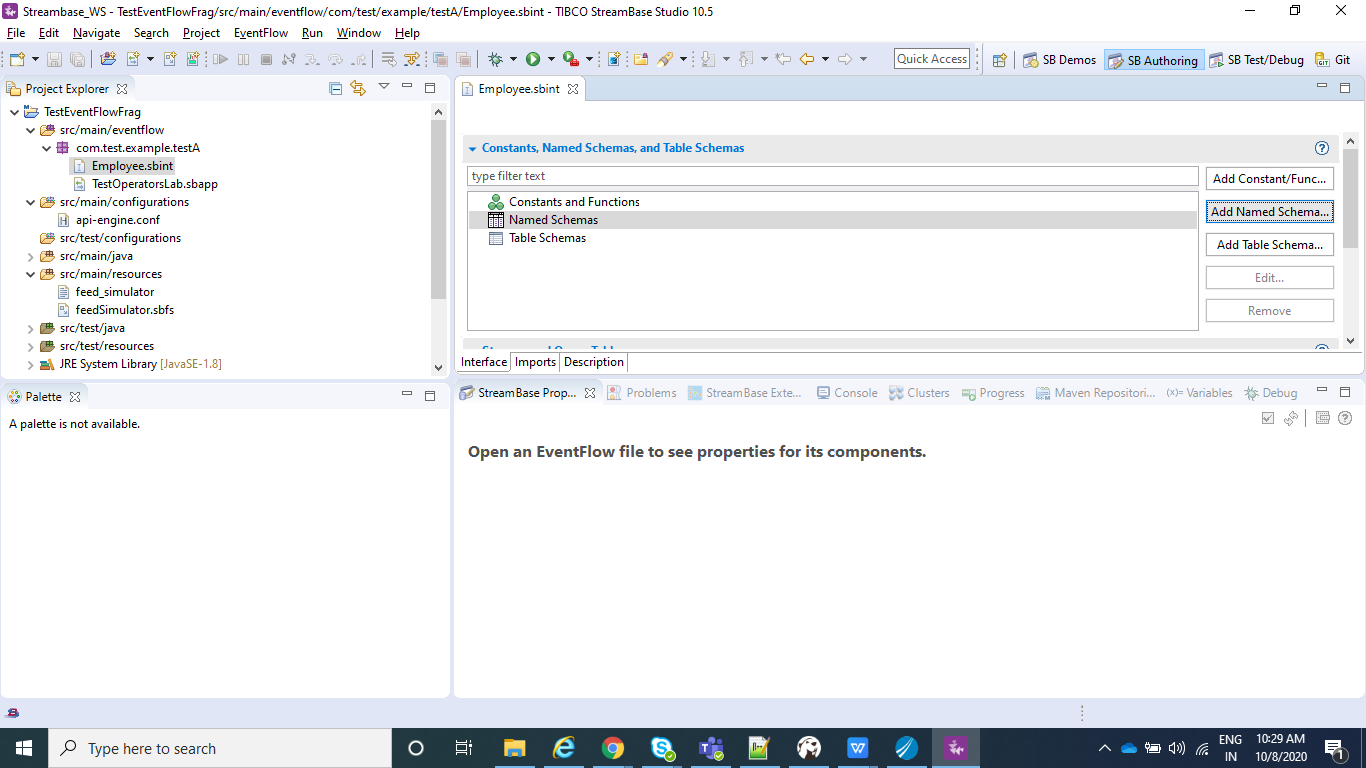
⦁ Click Next and provide the value for **Group Id**, **Artifact Id** and **Version**. Provide the packagename as <<Group Id>>.<<any name>> as shown in below screenshot. Please note down these three parameters and click finish.

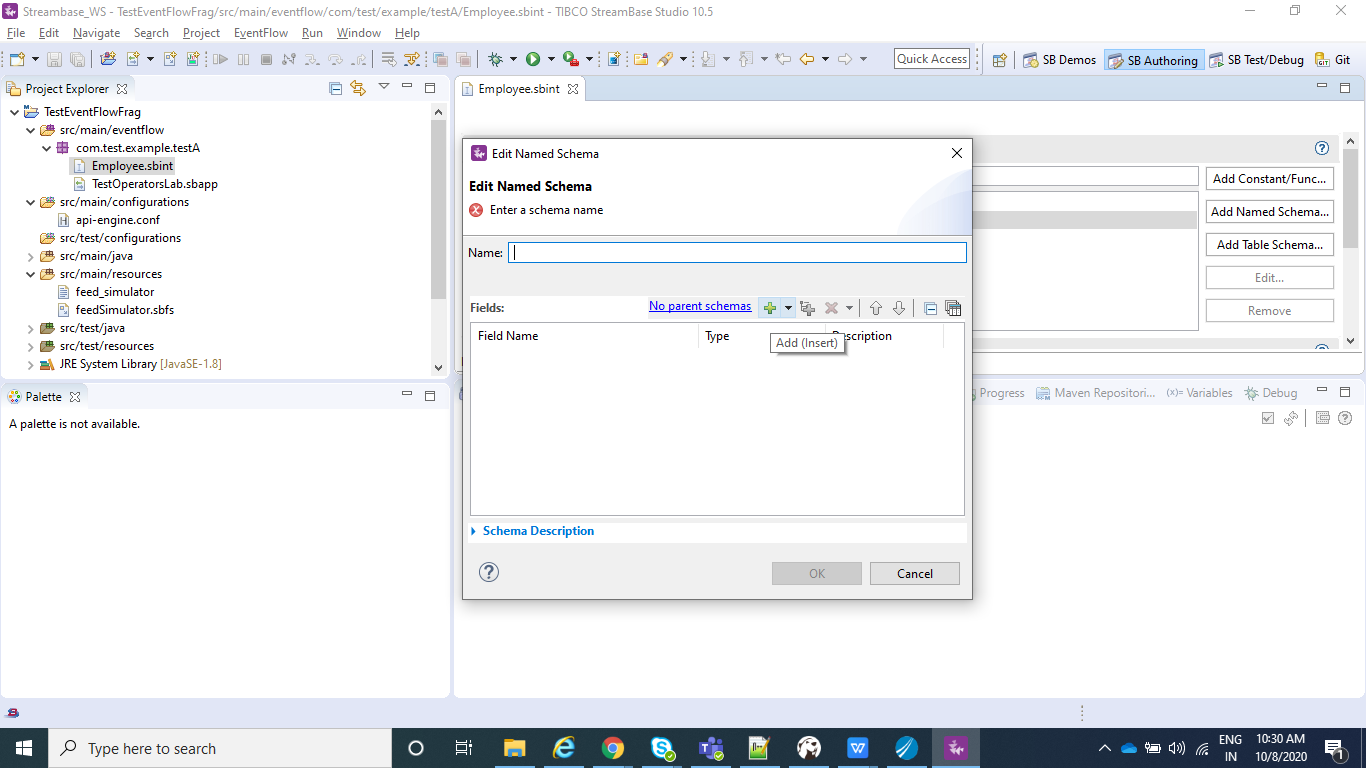


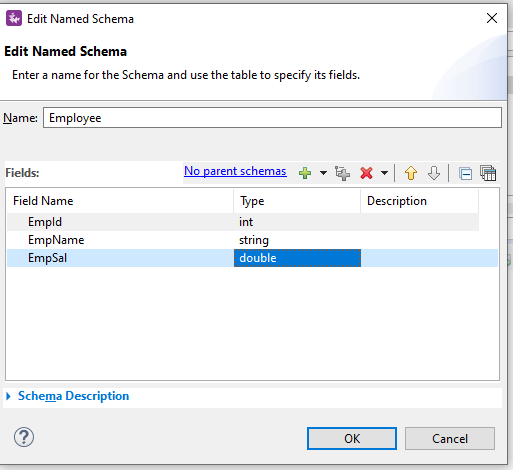
* Create a new event flow fragment and name it as TestOperatorsLab.
* Now go to the folder src/maineventflow and click on TestOperatorsLab.sbapp
* From the palette view click on Input Stream and Drag it into the canvas.
* Name the Input Stream as Customer Information.
* Now create a interface Employee that could be used across the modules.It is like a schema and using it across the main and sub process.



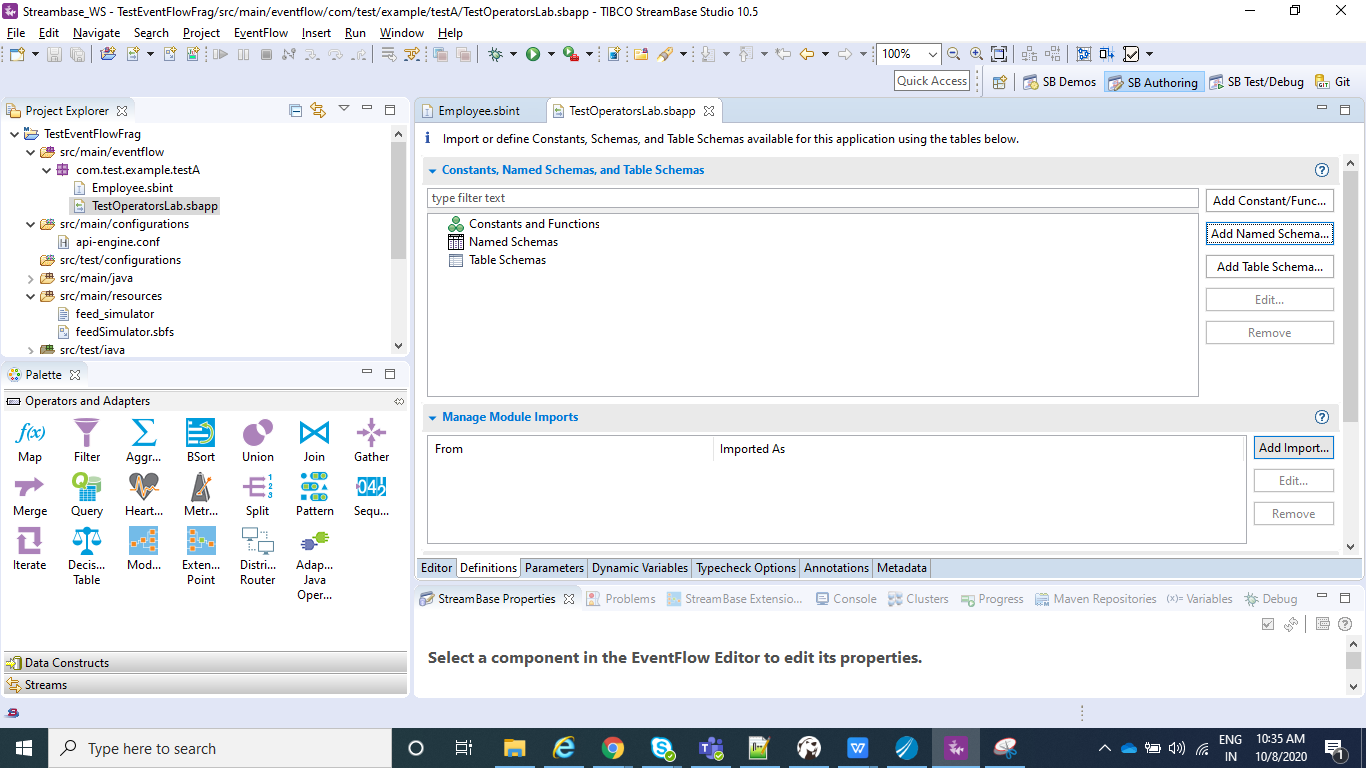
* Now Add Named schema to the interface that can be imported anywhere we require.We will add name as Employee and Fields would be EmpId, EmpName, EmpSal.

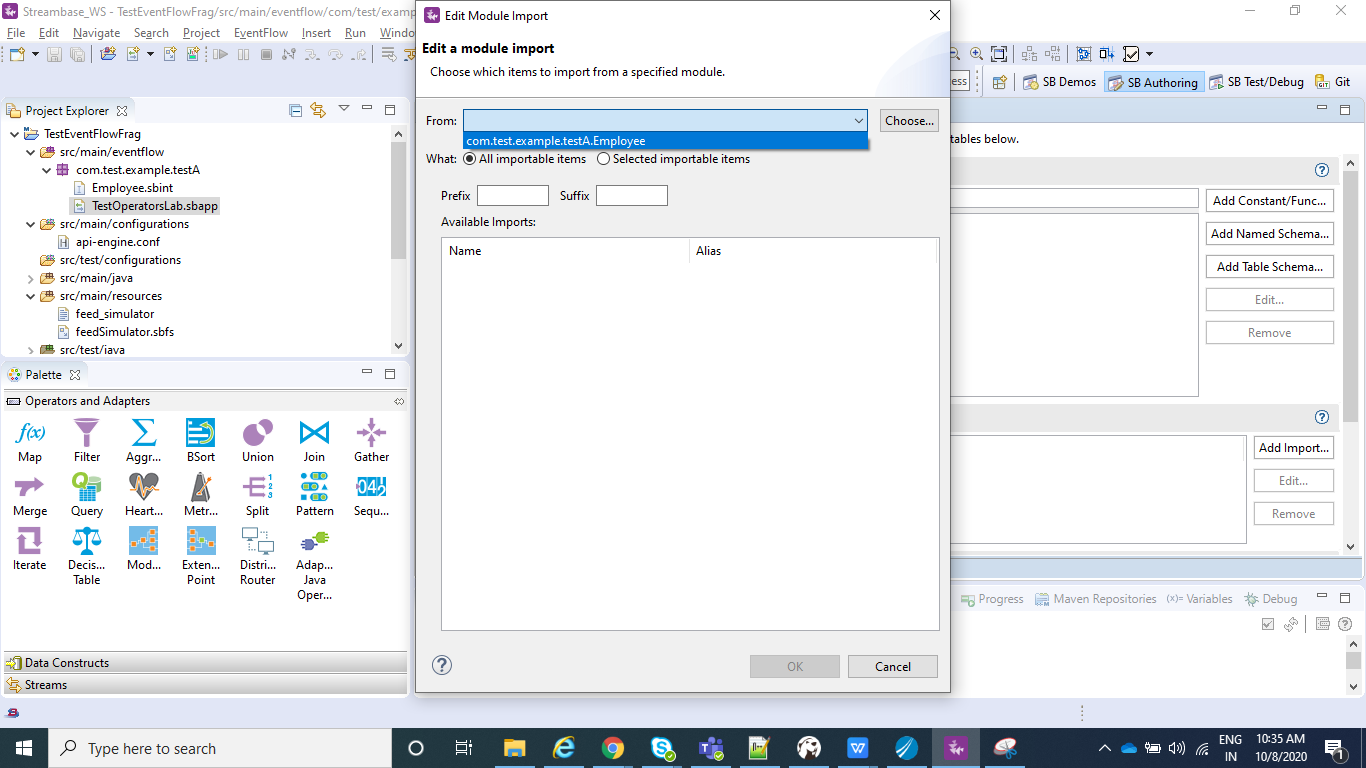




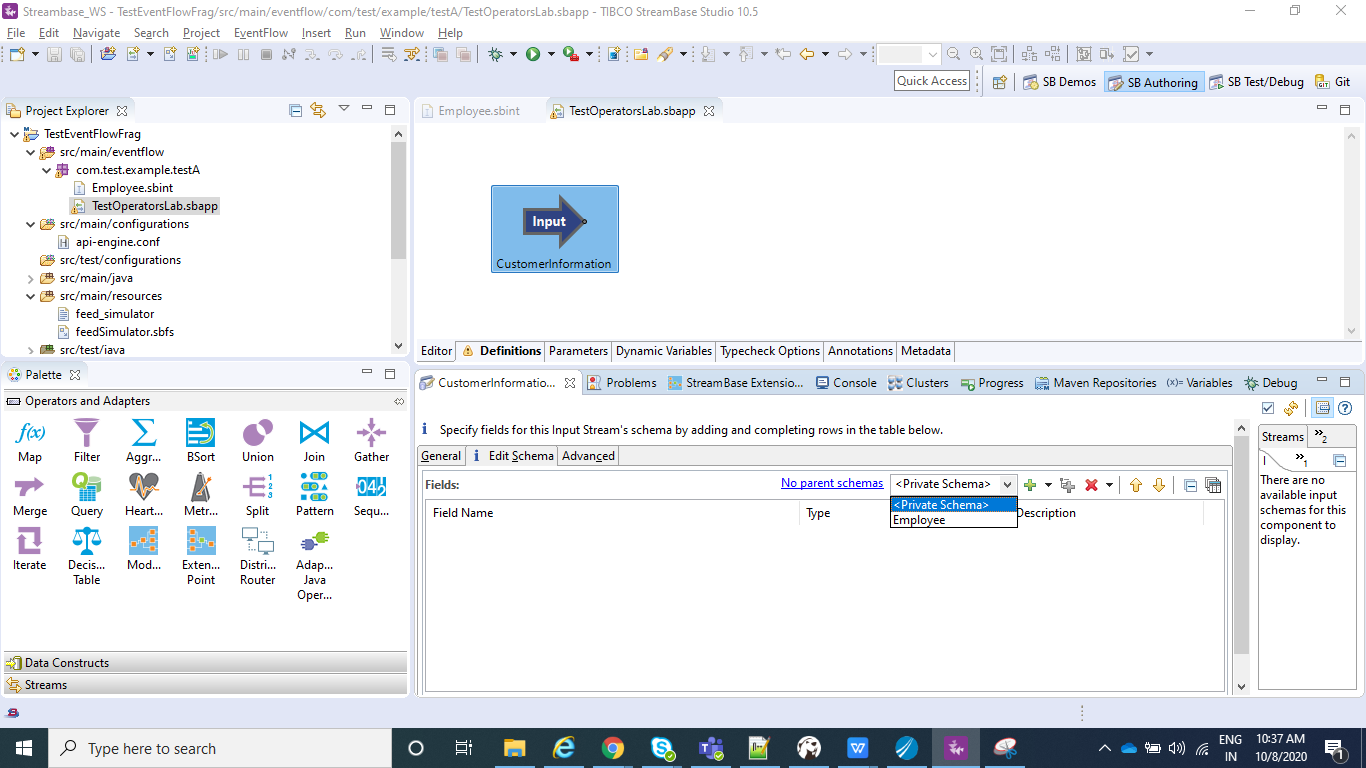


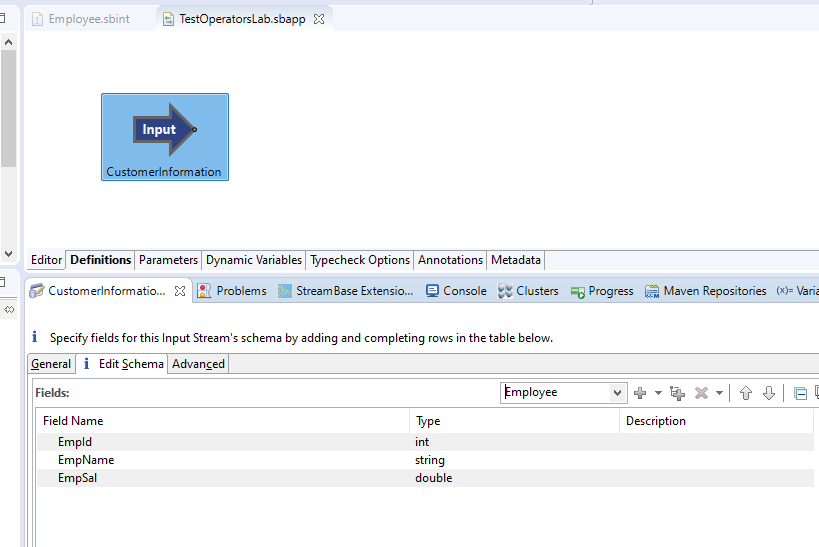
* Now in the Definitions tab, we can select that schema/interface we just created under Manage Module Imports.



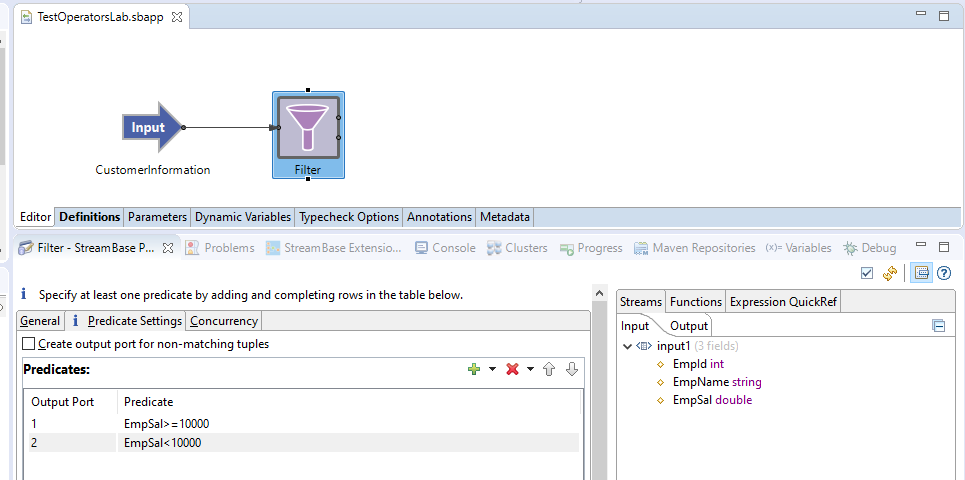


* Now after saving this you will be able to see the Employee schema under Edit Schema in dropdown as shown below.

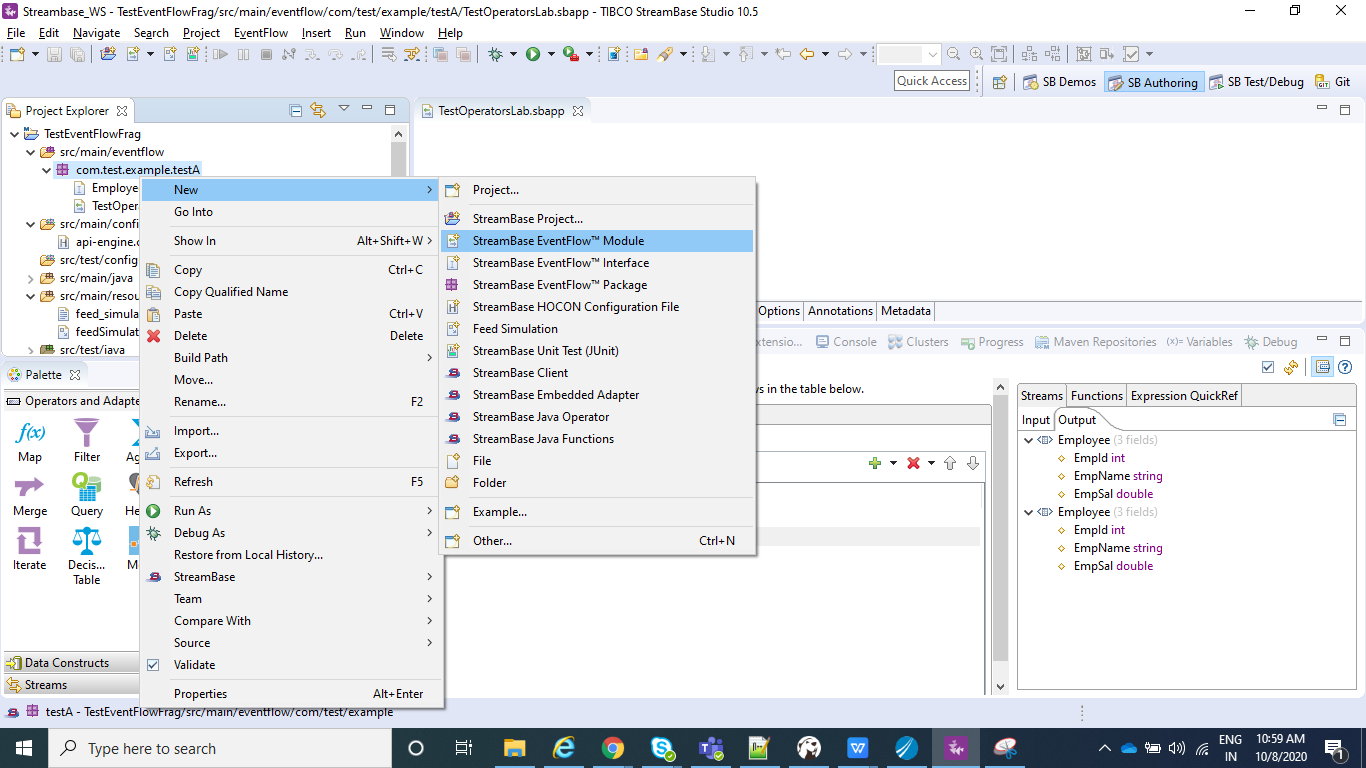


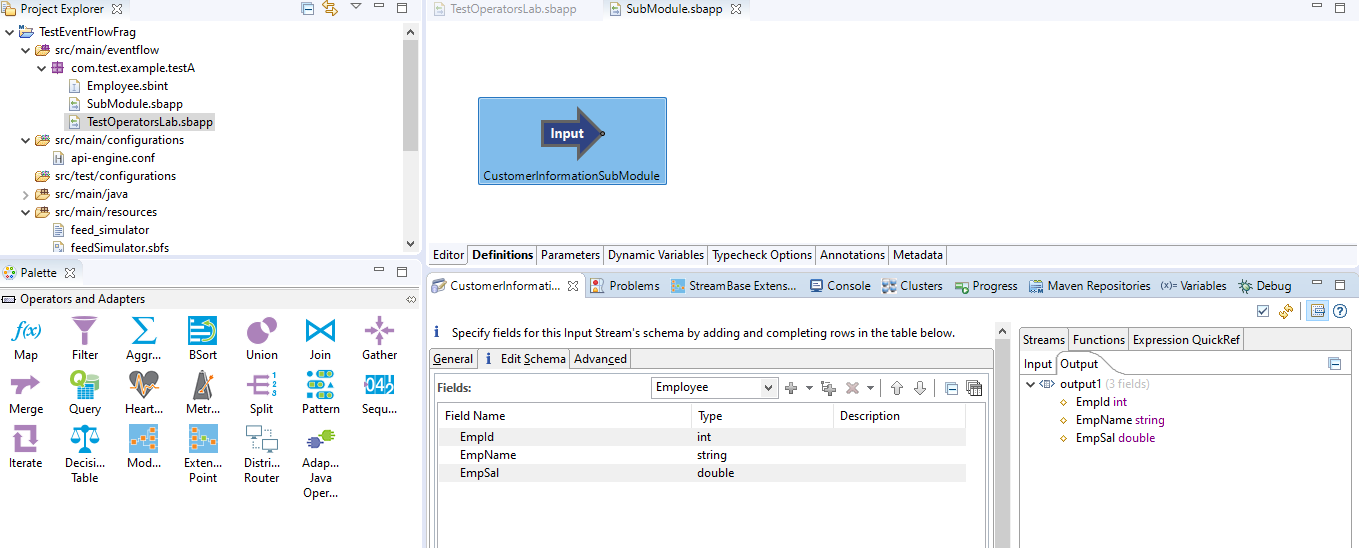


* Now consider a scenario where we want to filter the employees on the basis of salary.So all the employees with salary>=10000 and all the employees with salary<10000 need to be filtered out.For this purpose we will be using the filter operator as shown below.Here in Predicate Setting we will specify the conditions and we can see two dots in the output of the filter as shown below.

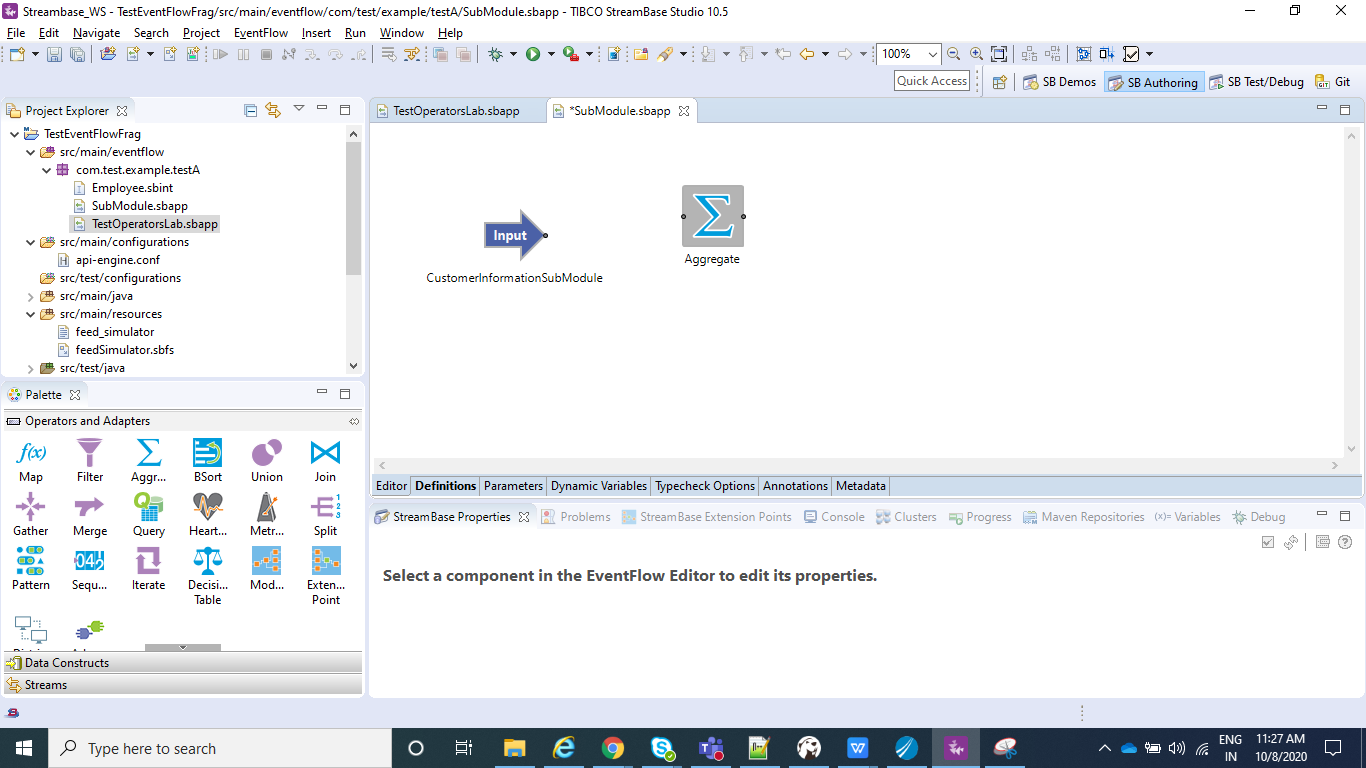


* Now we want to send the data of employee whose salary>=10000 to some other module. For this now create one more Module and name it as SubModule where we will be again importing the same schema Employee as we did for TestOperatorsLab.sbapp(CustomerInformation) and will have the Input stream named as CustomerInformationSubModule.

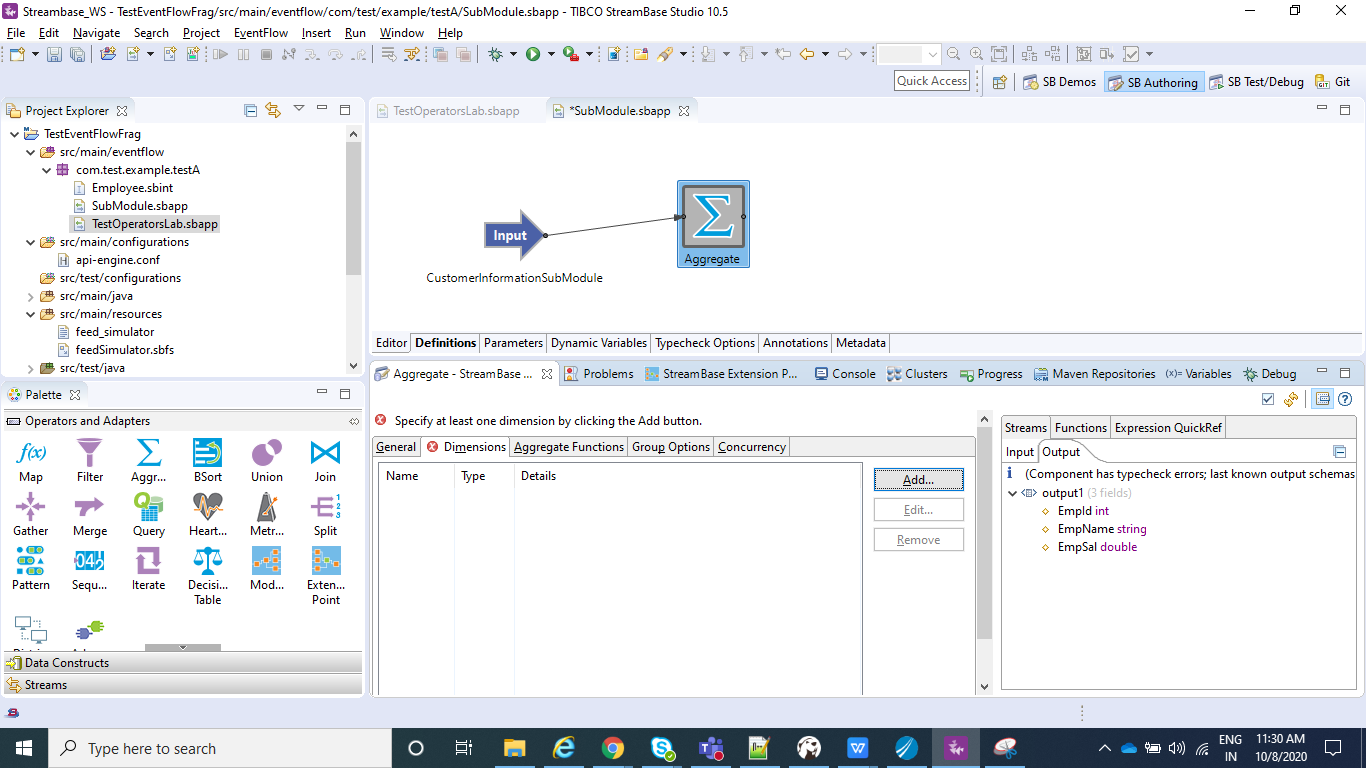


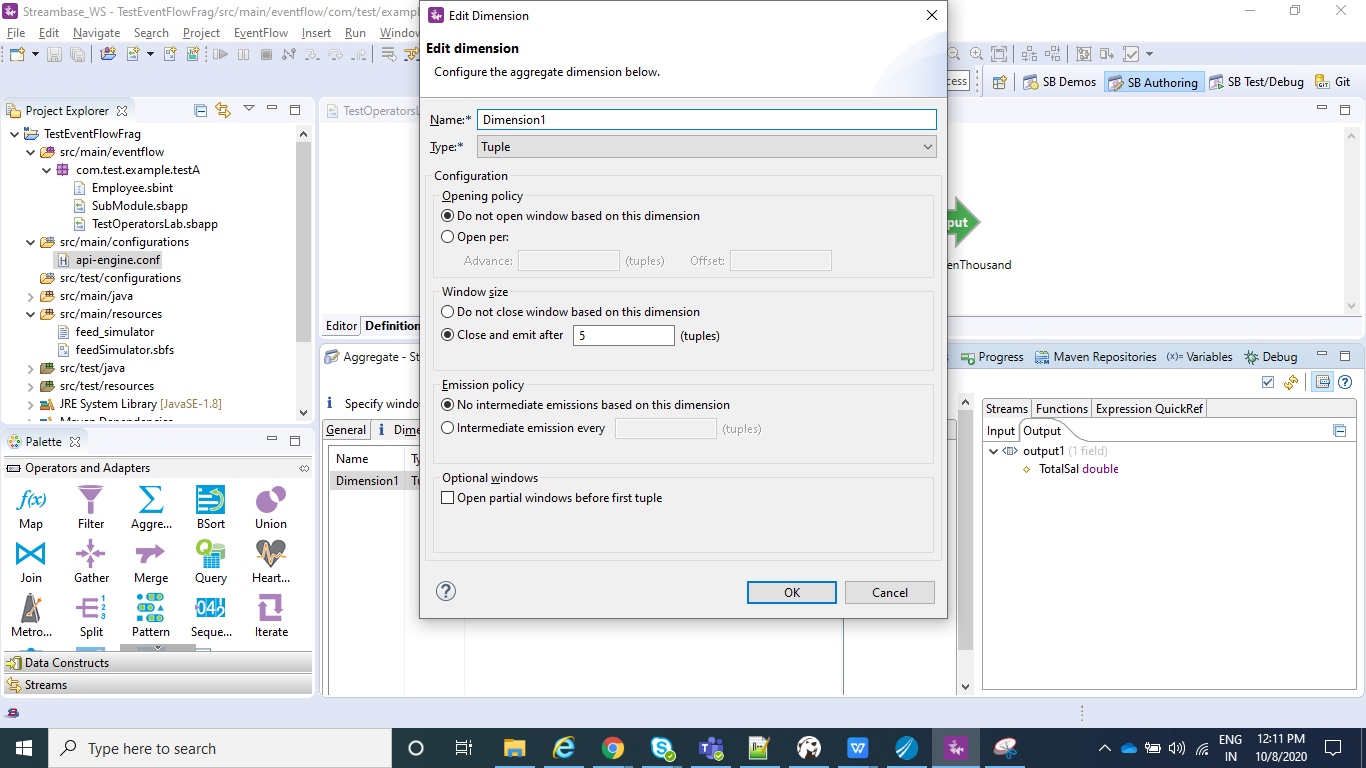


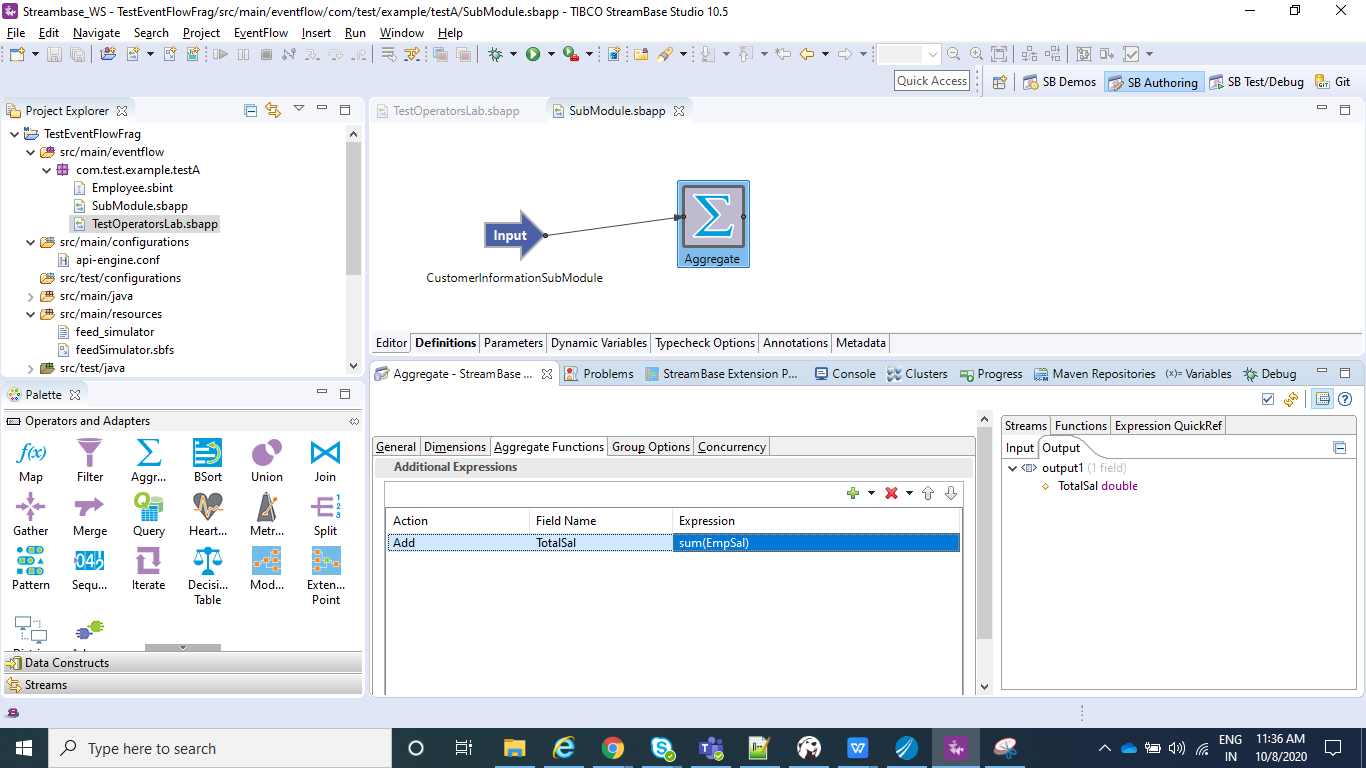
* Now we need to aggregate all the employee data with salary greater>=10000 for which we would be using Aggregate operator and will drag and drop that from Operators and Adapters and connect the Input Stream(CustomerInformationSubModule) to Aggregate.

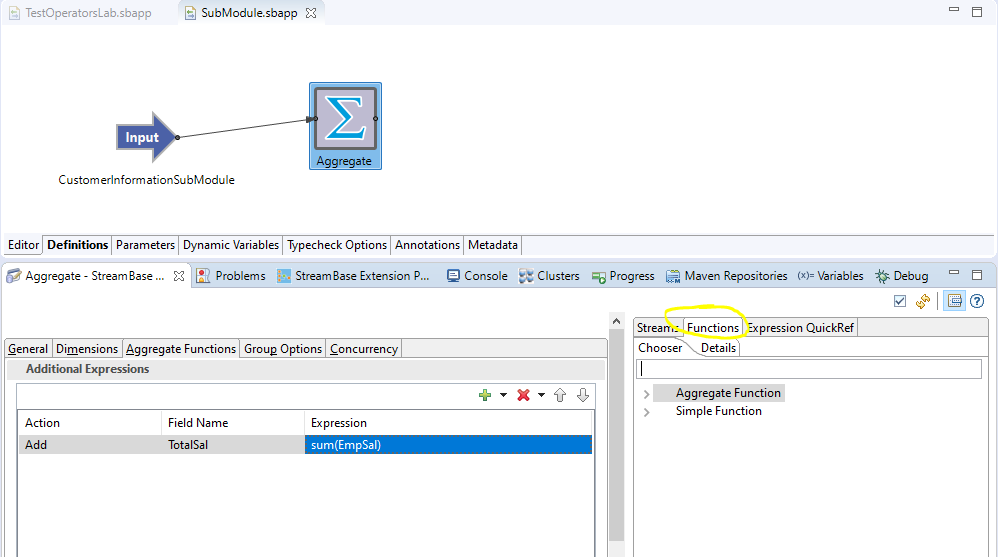


* Now we can add the dimension by clicking on Add button and select ok. To that Dimension we would select Window Size for Close and emit after as 5 or any other value you want. This value would be used to wait for that much number of input records and than only it will emit the result. Once that is done we can go to aggregate function and in FieldName we can write what is expected in output. In our case we need to sum all the salary>=10000. So in filed name we will write any name(TotalSal) and in expression we would do sum(EmpSal) which we are receiving from input.Like we use sum, we can also use many other functions like min, max and all can be found under Functions and same can be seen in below screenshots.

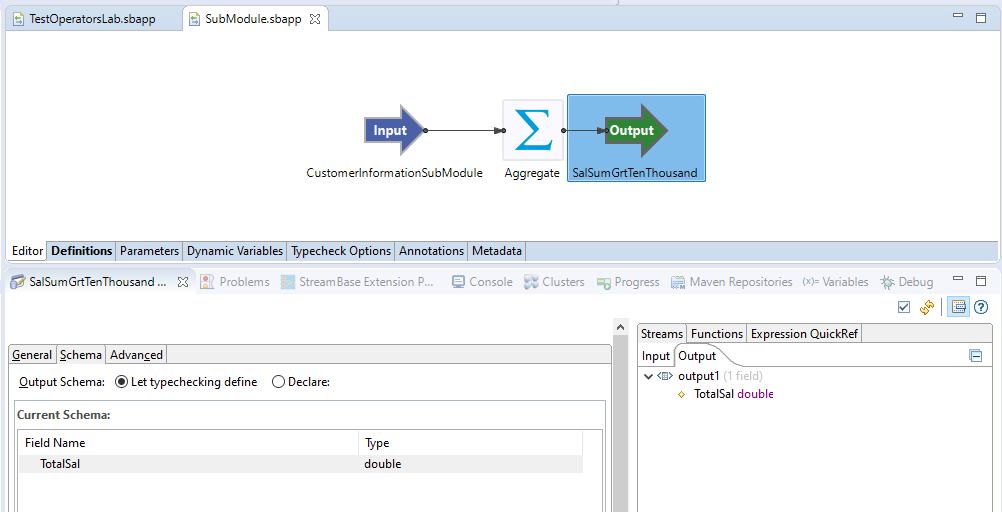




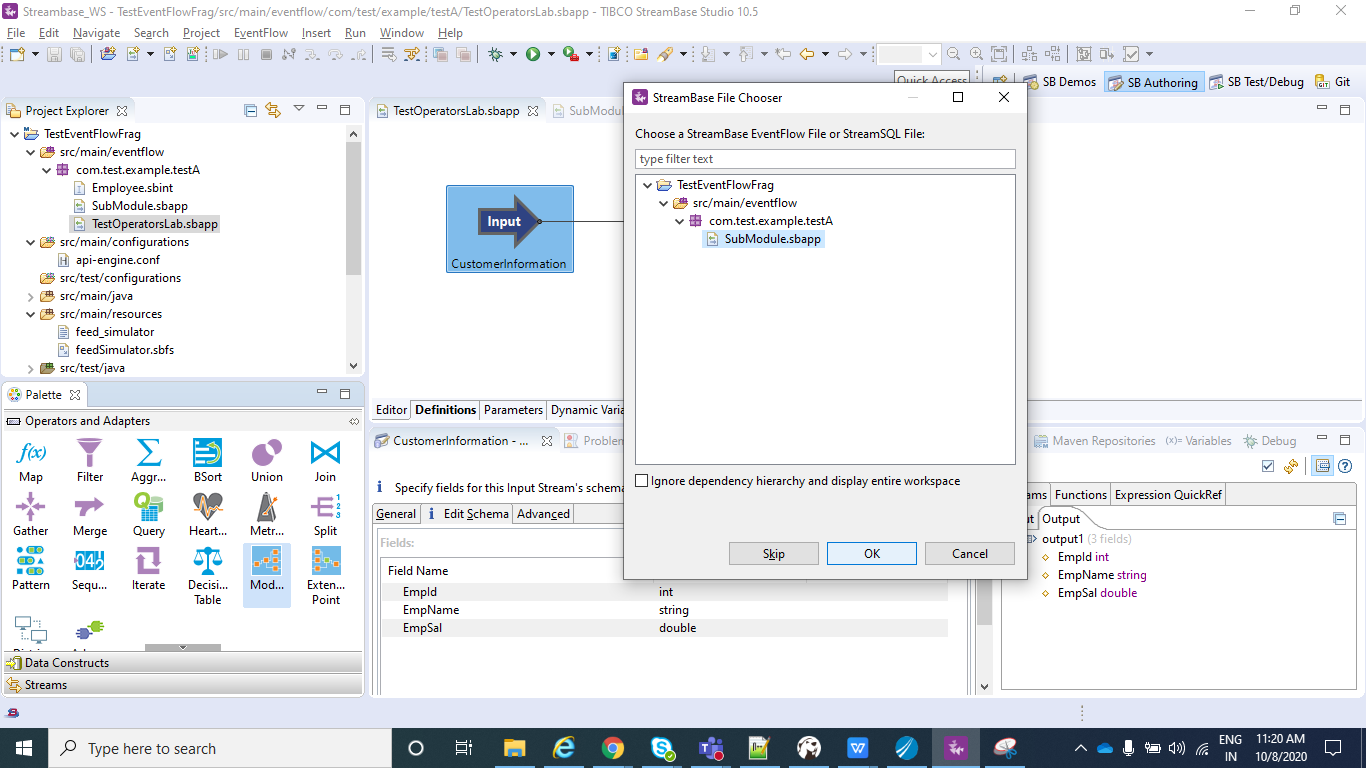




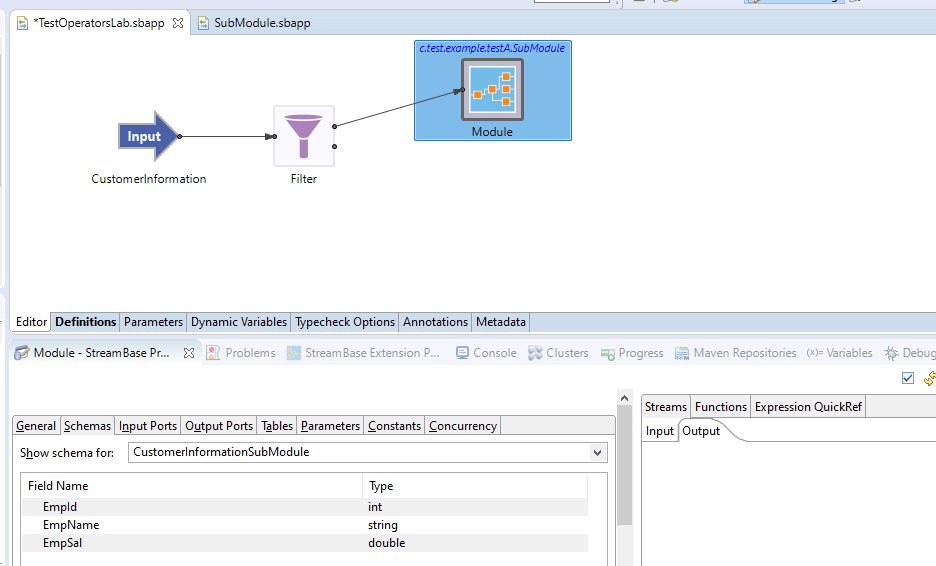
* Now in the output we have the sum of the salary and same we will pass to output stream named as SalSumGrtTenThousand.



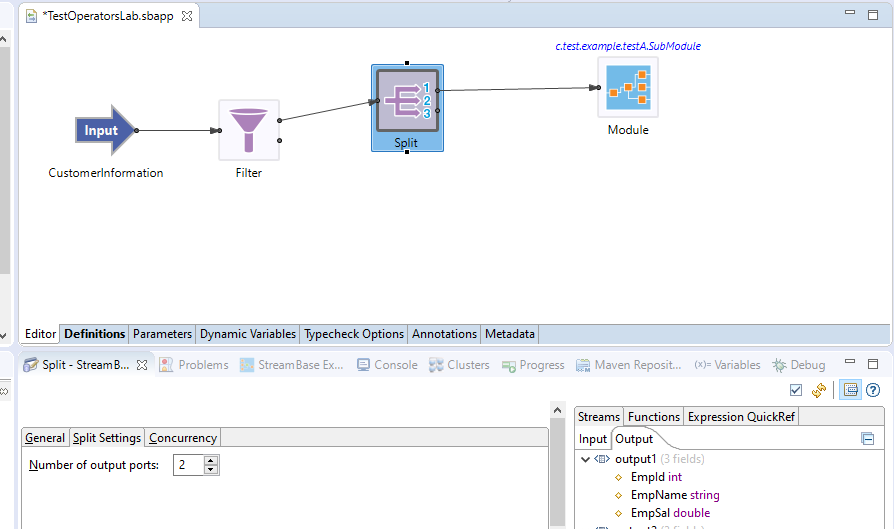
* Now in order to send the data from TestOperatorsLab.sbapp to SubModule.sbapp we need to drag and drop Module reference from Operators and Adapters and need to select the SubModule.sbapp which we just created in above steps.



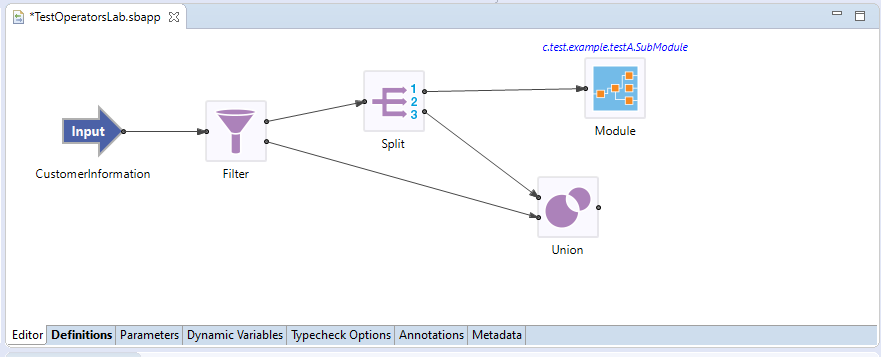
* Now we can connect from filter to Module, so that the data flows for the customer to the SubModule.



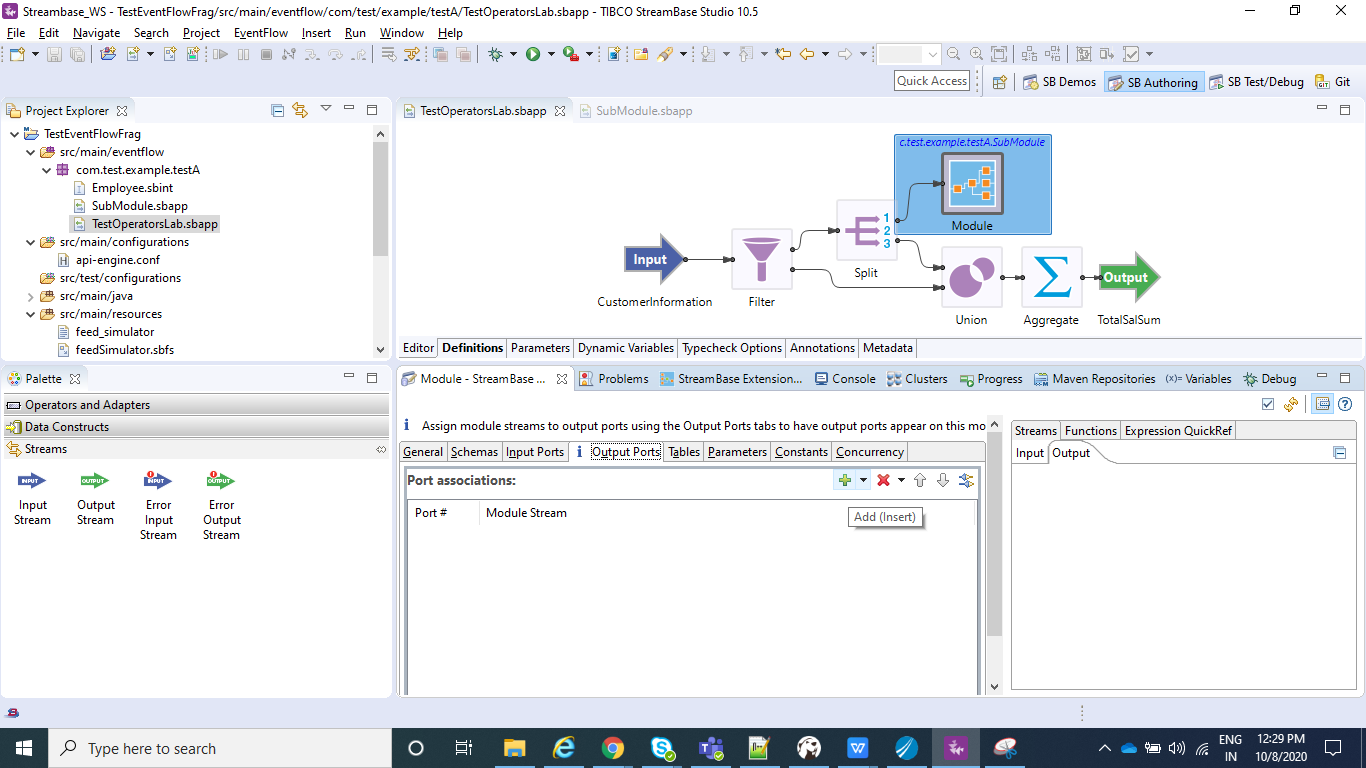
* Now we have passed the employee data with salary>=10000 to sub module, but at the same time we need this data in main process(TestOperatorsLab) as well.So for this purpose we will use split operator from Operators and Adapters, the use of which is to pass the same data to n(2 in our case) number of the outputs as shown in below screenshot.

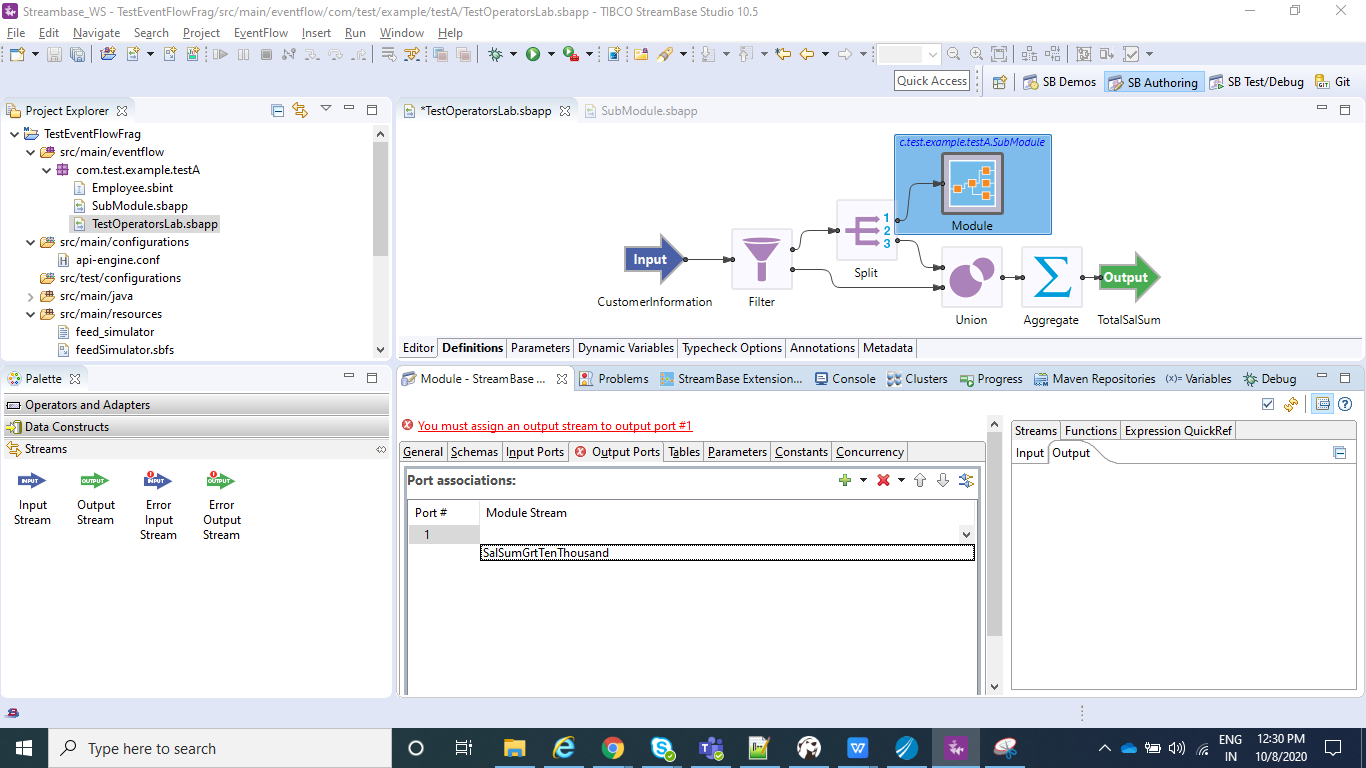


* As we can see the first output is going to sub module and second output is not connected to anyone for now.
* Now since we need to see the sum of salary for all the employees irrespective of whether it is 1 RS or 100000 RS, what we will do is that we will combine this two data(one from split and one from filter) and same we will pass to union operator from Operators and Adapters.

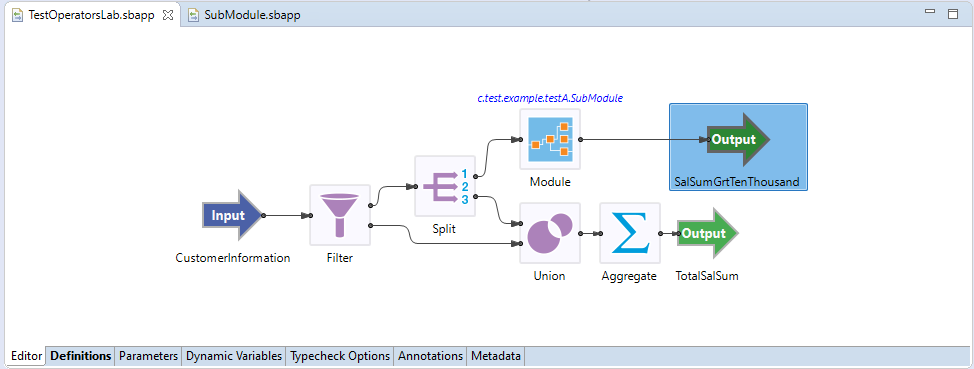


* Now in the output of Union we have all the employee data, so now we can use again the Aggregate operator as we have used above and do a sum(sal) and than we can connect that to output stream named as TotalSalSum.Here in this Aggregate operator we will have the value for n as 7.So this Aggregate operator will emit only when it gets 7 input data.
* Also we want the data of SumModule to have in the Main module(TestOperatorsLab). For this we would edit output ports in sub module and and the output stream.





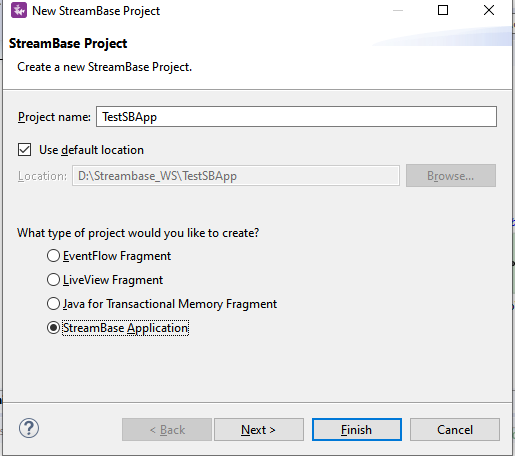
* Now we will copy the output stream(SalSumGrtTenThousand) from SubModule.sbapp and paste the same in TestOperatorsLab.sbapp and connect Module to that output stream.



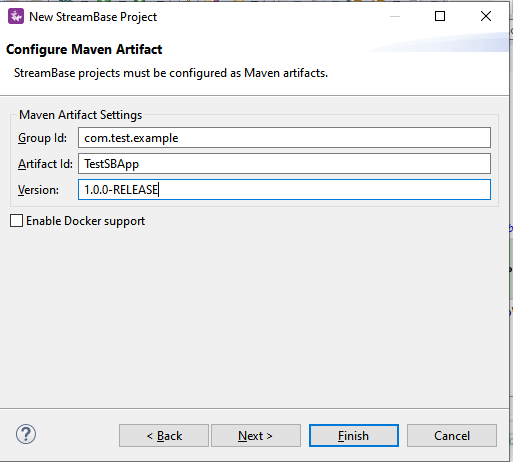
⦁ Now In the Project Explorer view, select and right-click the Streambase Application Project **(TestEventFlowFrag in my case)--> Run As>Maven Build...**

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

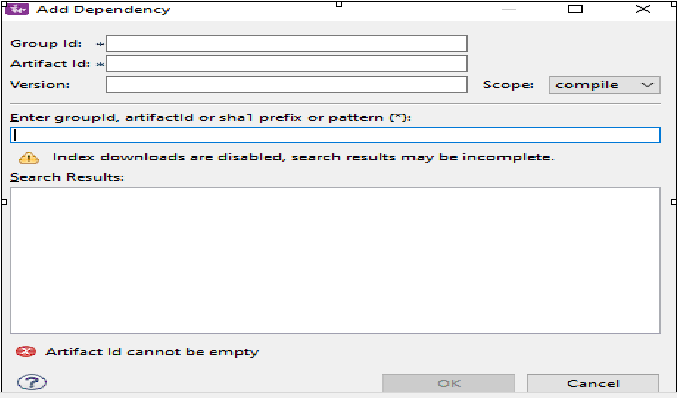
⦁ Now create a new Streambase Project. Select Radio button as **StreamBase Application**.

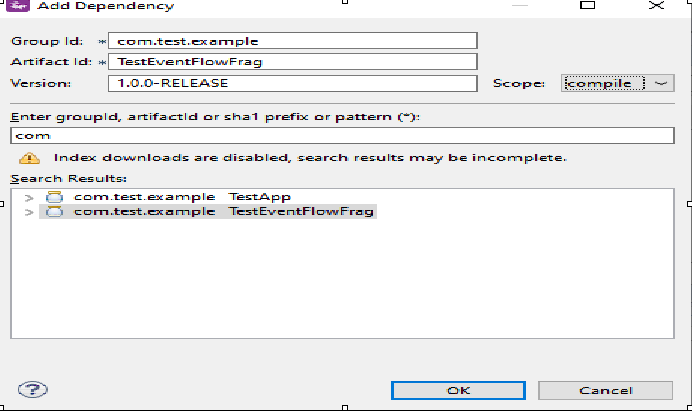


⦁ Now click next and make sure you provide the same **Group Id** and **Version** that you specified for the EventFlow Fragment created Earlier and click **Finish**.



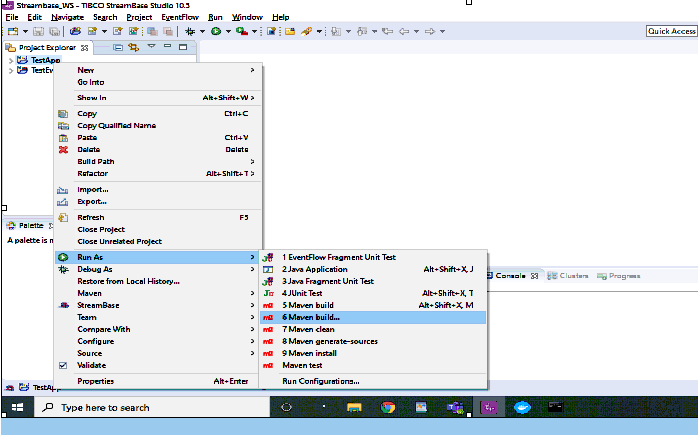
⦁ Add the EventFlow Fragment Project depedency to our main Streambase Application. Right click on Streambase Application Project **(TestSBApp in my case) --> StreamBase>Manage Project Dependencies**. Add the depedency for the EventFlow Fragment and click ok.

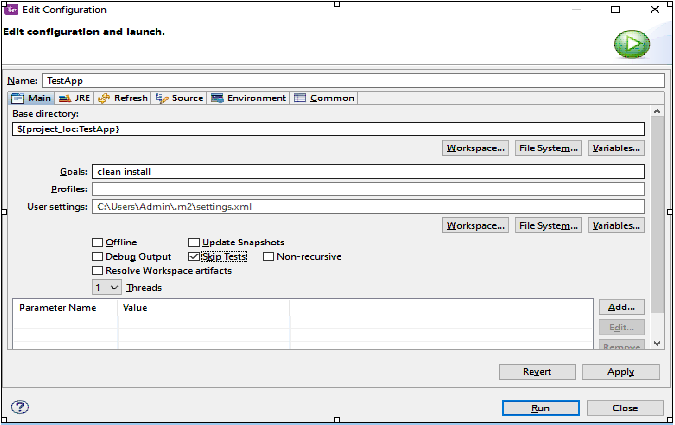


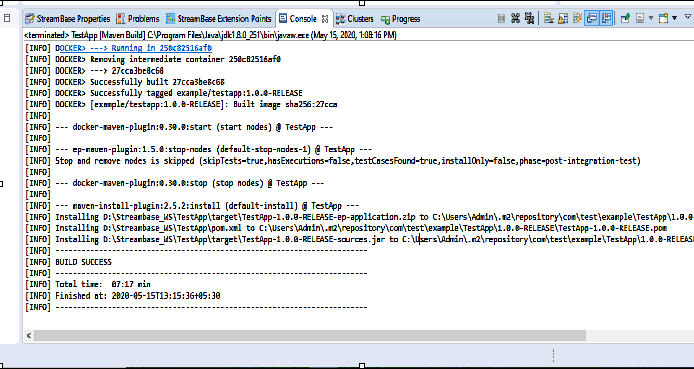


⦁ Now In the Project Explorer view, select and right-click the Streambase Application Project **(TestSBApp in my case)--> Run As>Maven Build...**

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

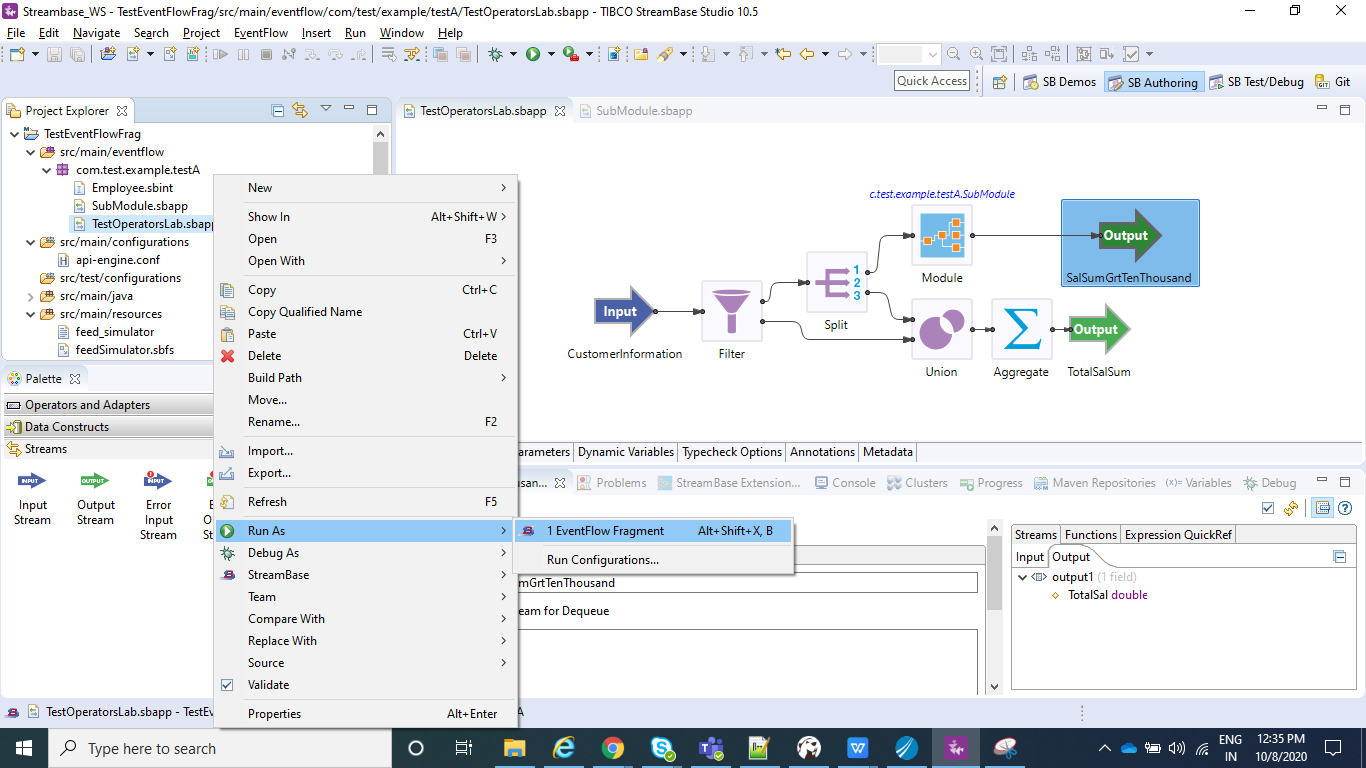




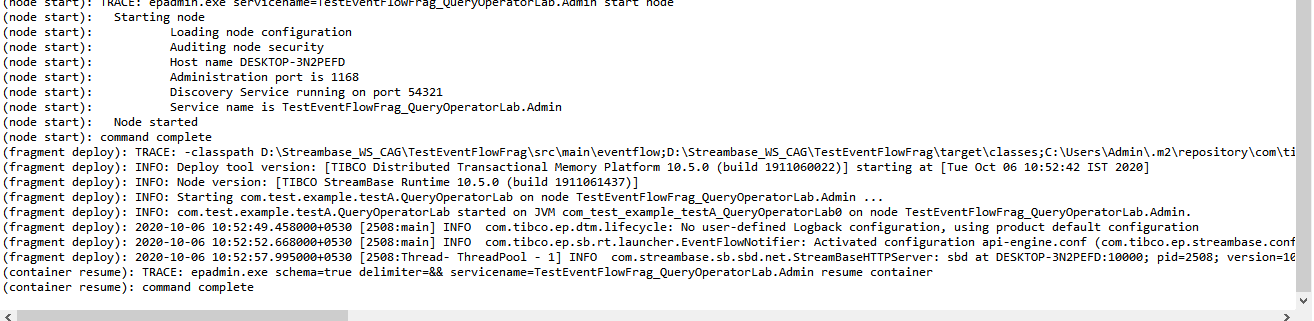
⦁ After the above step check the console and it should be showing BUILD SUCCESS message.

**Run the application in Studio:**

⦁ Right click on your .sbapp(TestOperatorsLab) 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 we will send data using manual input and after we send the 5th data, we can see one of the output streams which is SalSumGrtTenThousand .The reason for this is we have configured 5 in our Aggregate operator under Dimension. Now the question would be why only SalSumGrtTenThousand output stream and not TotalSalSum one. The answer is because for the aggregate operator used in TestOperatorsLab we have n=7. So we need 7 inputs, as we pass another 2 inputs we can see data in the stream TotalSalSum .

1,A,10000

2,B,20000

3,C,30000

4,D,10000

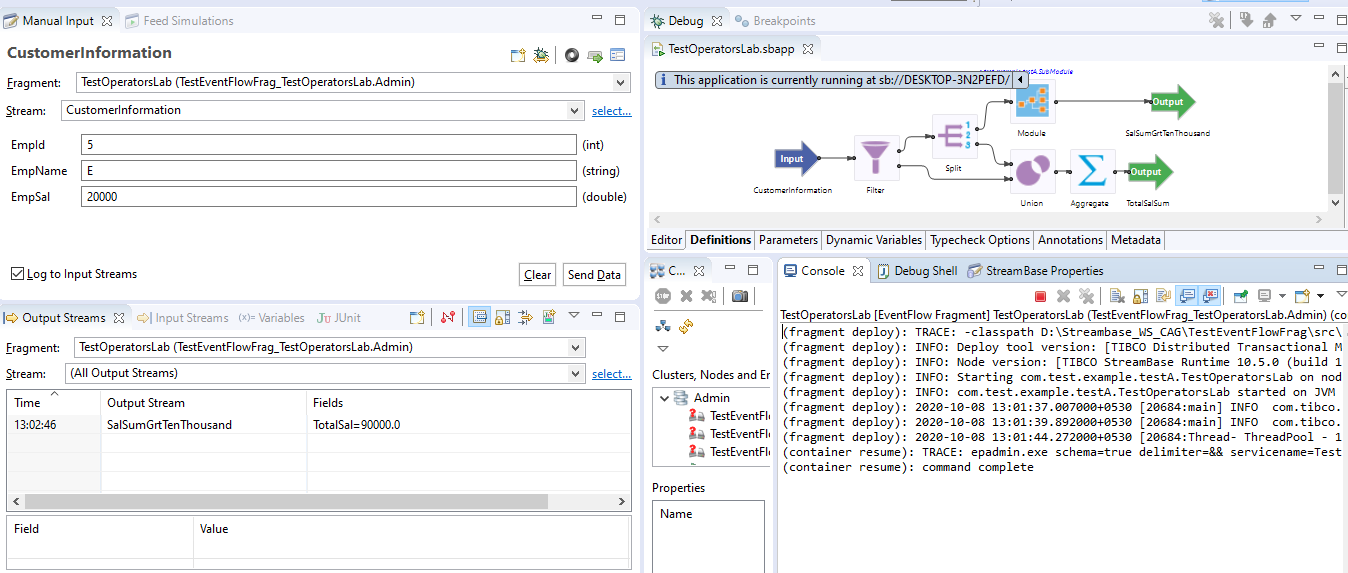
5,E,20000

**6,F,1000**

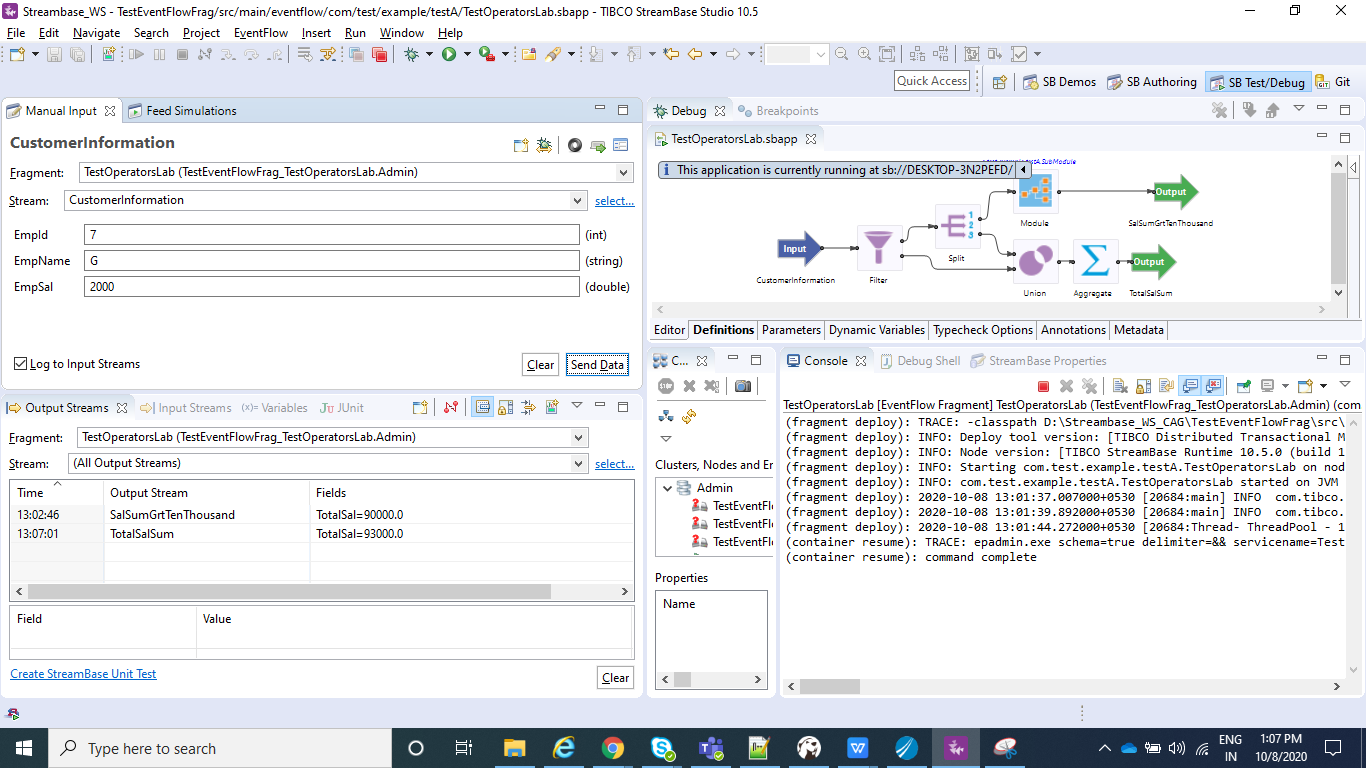
**7,G,2000**

* Now we can see two streams with output value as 93000(sum of total salary for all the employees) and 90000(sum of salary for the employee whose salary >=10000)

When we send first 5 data:

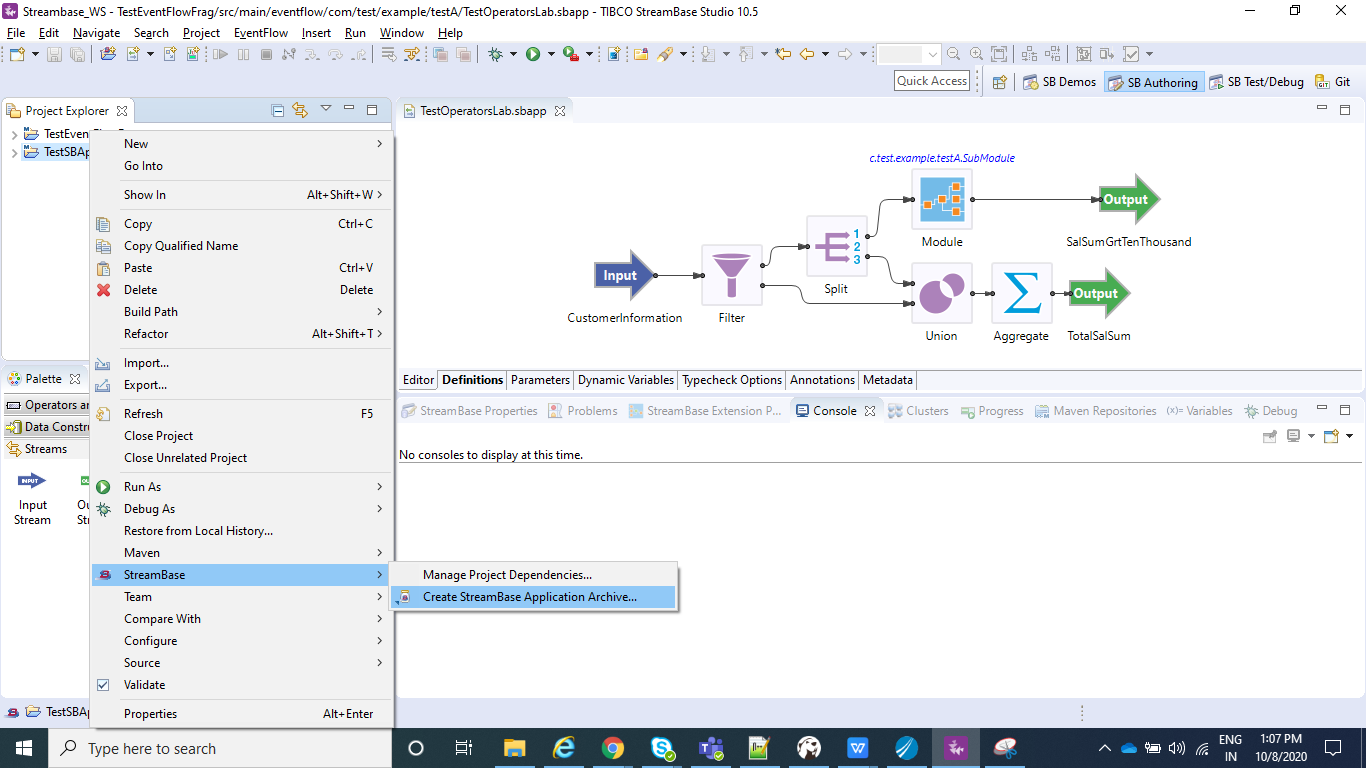


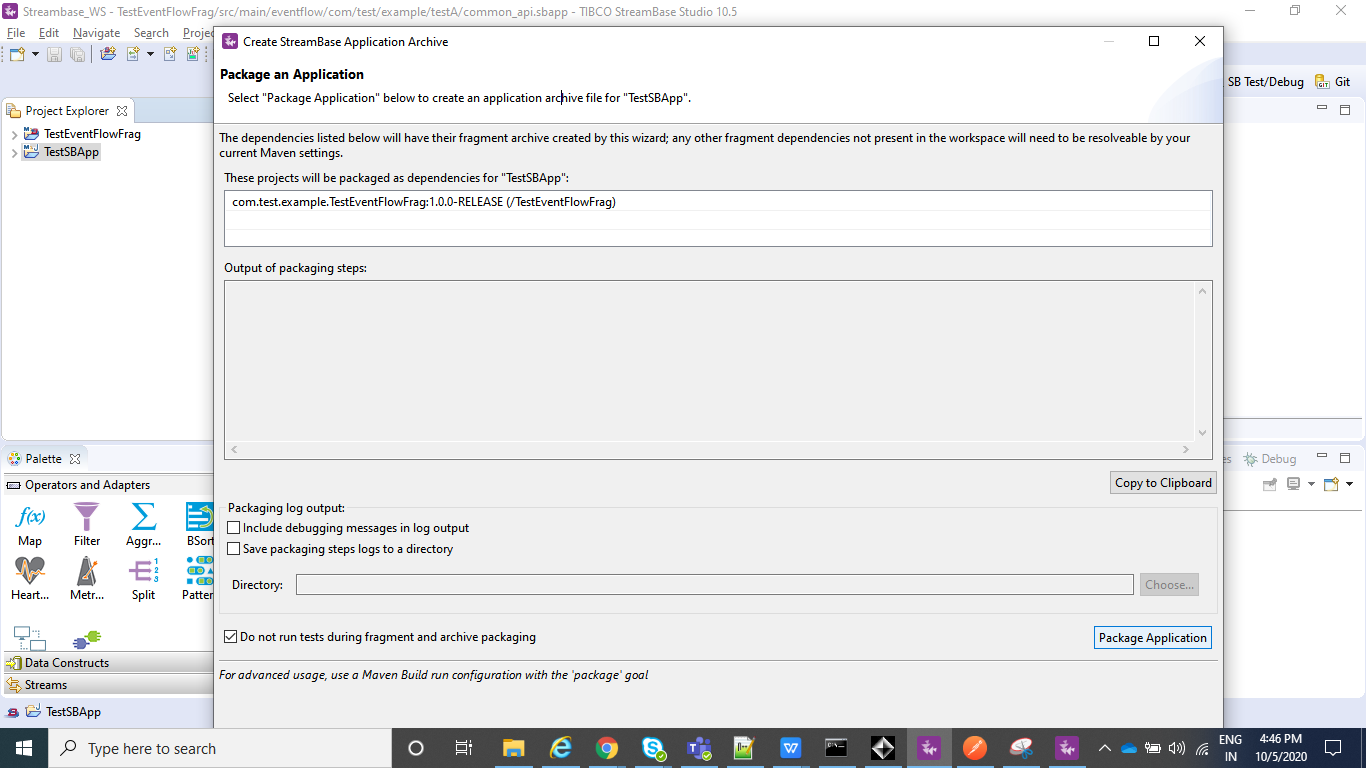
After we send another 2 more inputs:



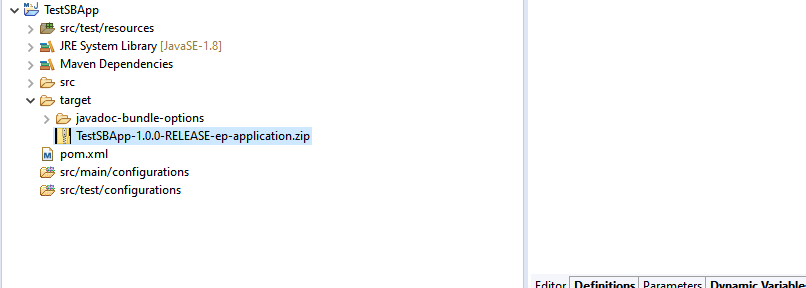
**Create the Application Archive:**

⦁ 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.



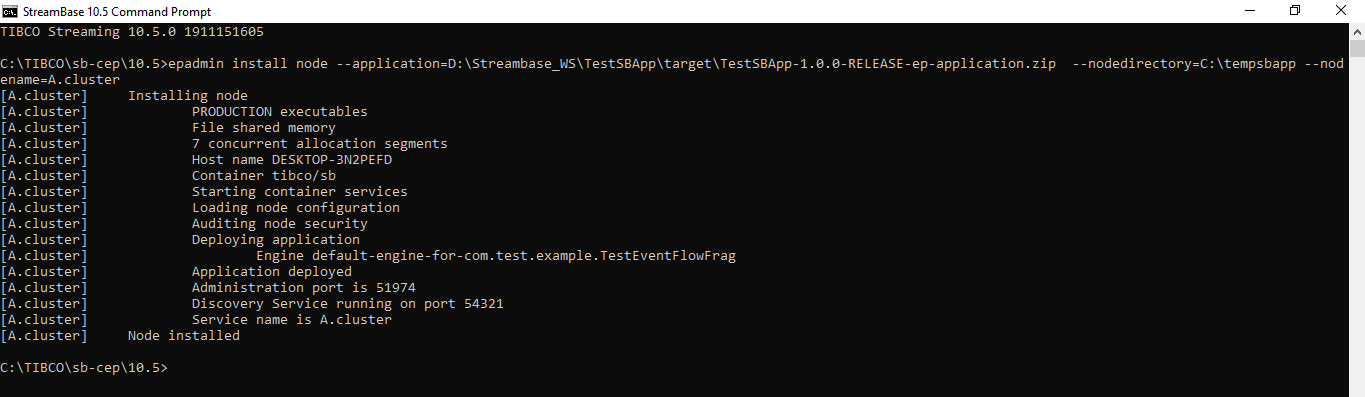
**Deploy and Run the Application:**

⦁ 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**

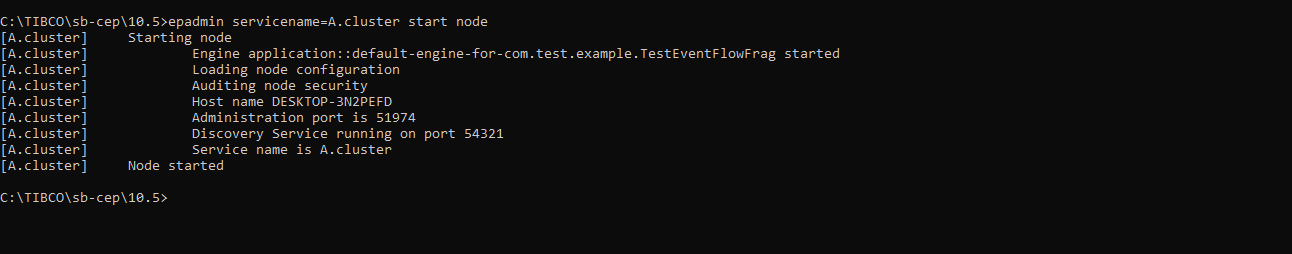


⦁ Now after your Node is installed, we can **start**, **stop** or **remove** node using below commands and once you have started the node you can test your application using postman in the same way we did before:

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

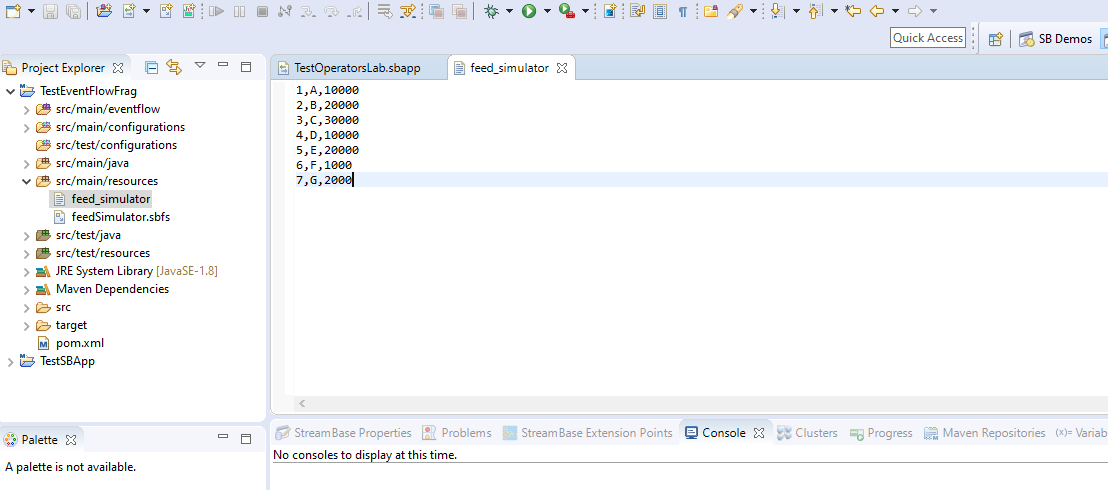
**epadmin servicename=A.cluster stop node**

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

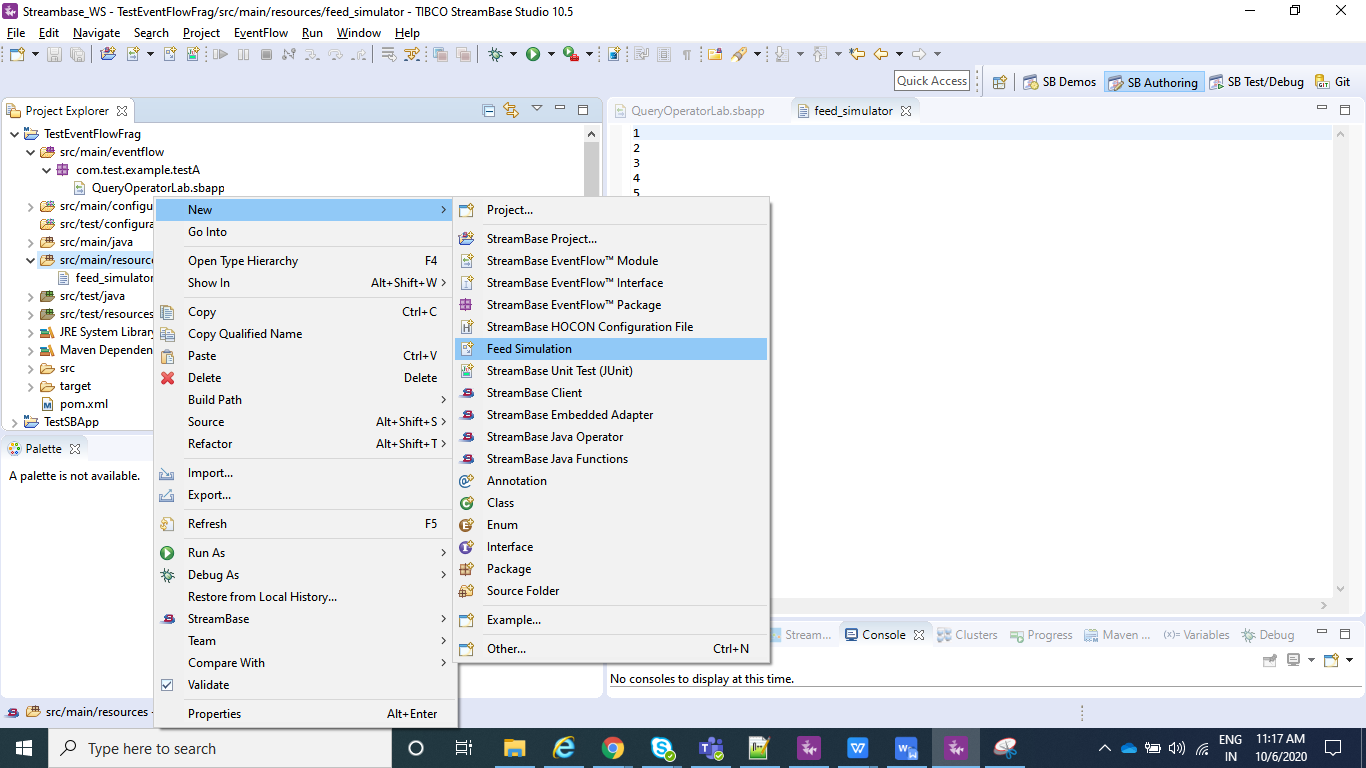


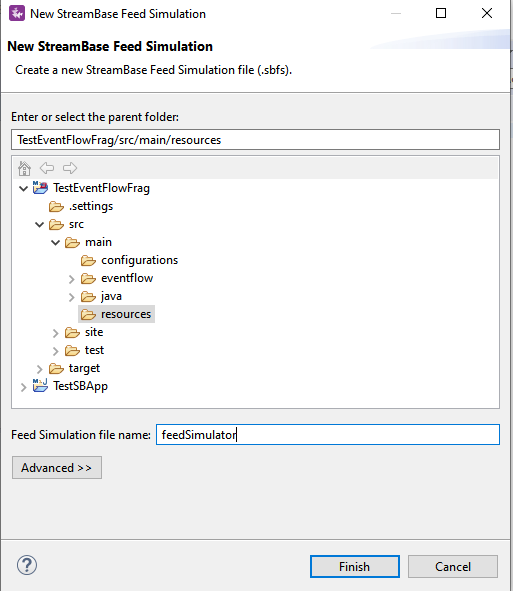
**Test the application (Using Feed Simulator file from command line):**

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

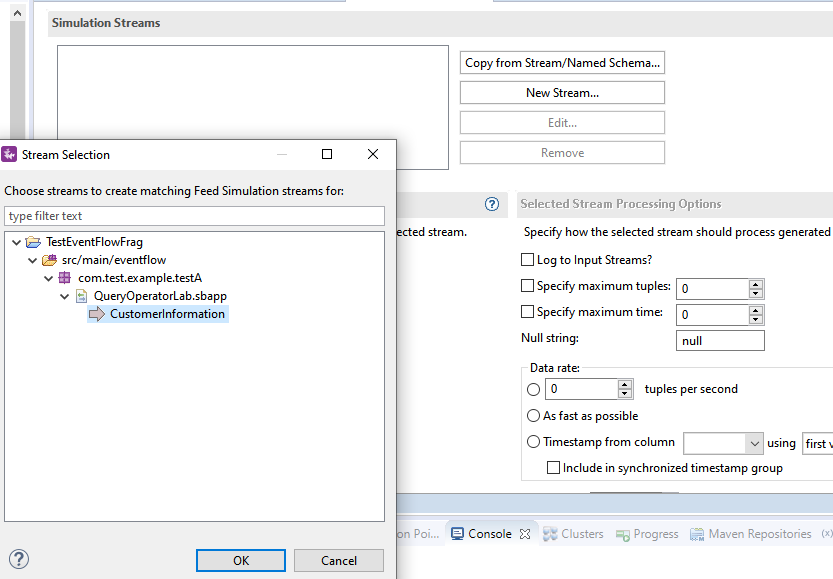


* To Configure feed simulator please follow below screenshots :

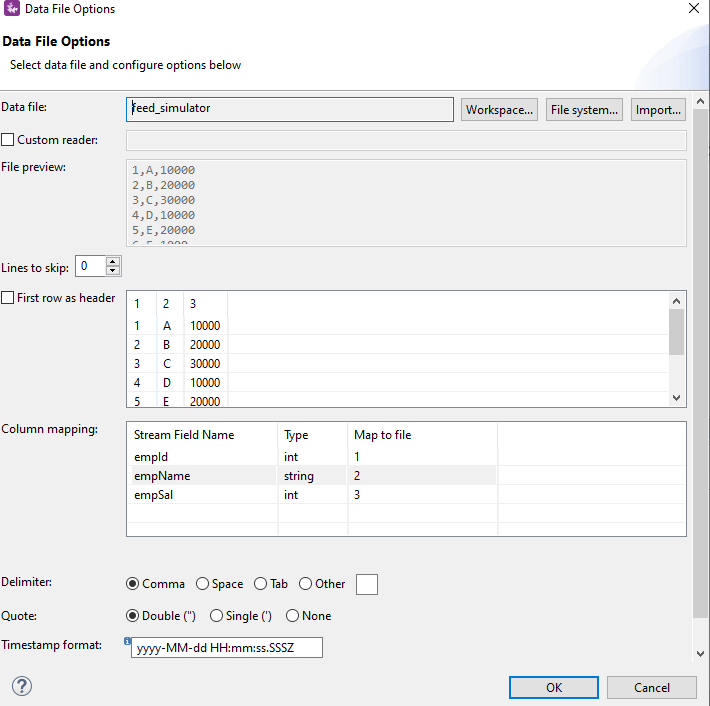




* Click on Copy from Stream/Named Schema and go to your Input Stream(CustomerInformation in my case) and click OK.



* 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.TestOperatorsLab"

containerName="TestOperatorsLab"

},

{

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

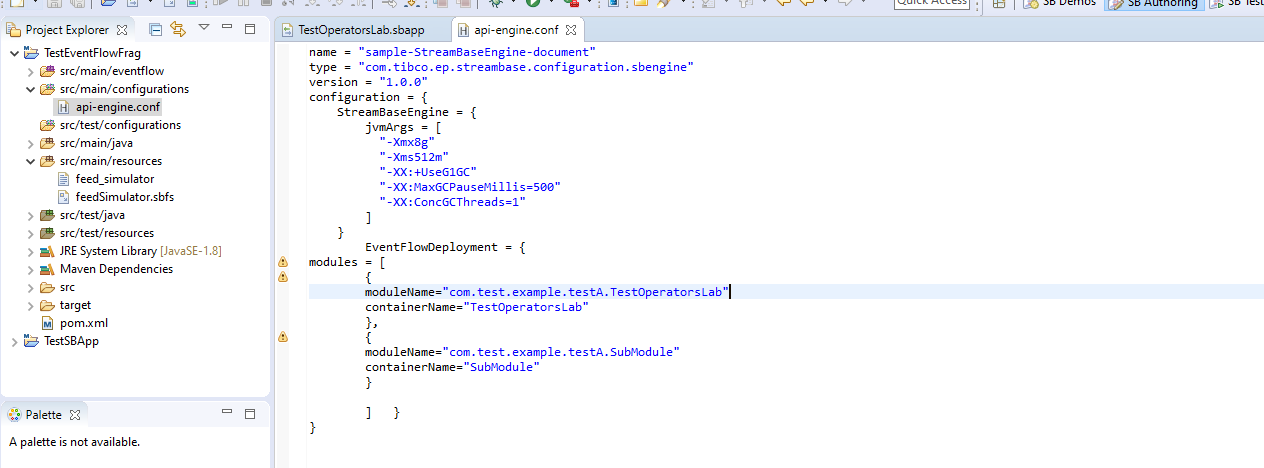
containerName="SubModule"

}

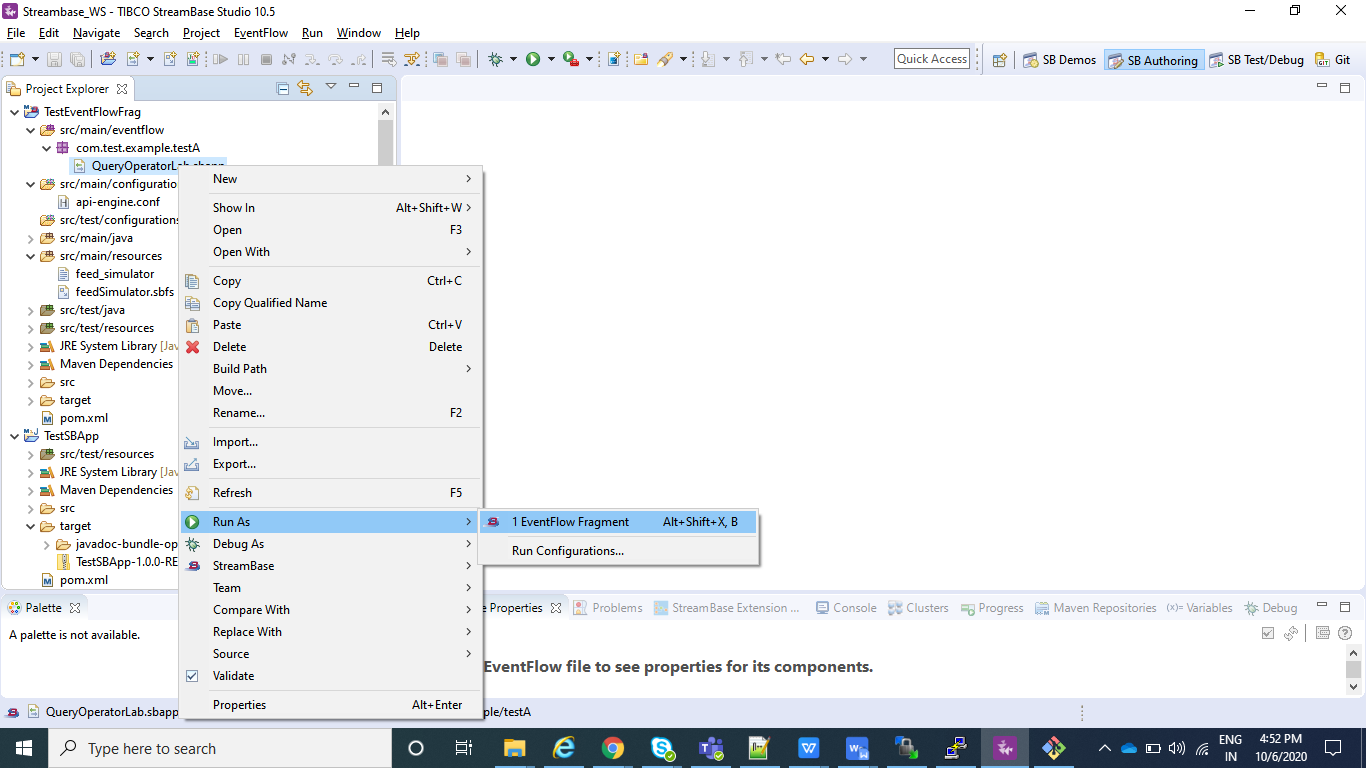
] }

}

Note : moduleName="com.test.example.testA.QueryOperatorLab" 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.

