



# LAB- Local and Global Transactions

## STEP1 – Local Transactions

In this step, you will be working on **11-local-transactions** project

execute db1\_product.sql present in src/main/resources of this project

Open localtransactionsdemo.xml in src/main/resources

Observe that this flow has 2 db insert components. One inserting into product table and other into copyproduct of same database db1

Between these 2 insert components, observe that the transform message component is throwing exception if brandname of product is “Samsung”. This is done just for simulating an error between 2 inserts.

Run the application .

Open post man and give a request to

[http://localhost:8081/save?name=Laptop&brand\\_name=Hp&original\\_price=1000](http://localhost:8081/save?name=Laptop&brand_name=Hp&original_price=1000)

You should get 200 status code and data should be inserted into product table and copyproduct table in db1

Now give a request to

[http://localhost:8081/save?name=Laptop&brand\\_name=Samsung&original\\_price=1000](http://localhost:8081/save?name=Laptop&brand_name=Samsung&original_price=1000)

Observe that you get 500 status code and below error message



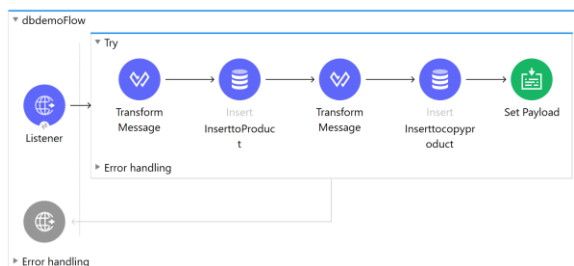
```
Body Cookies (1) Headers (4) Test Results Status: 500 Server Error
Pretty Raw Preview Visualize Text
1 "We dont Accept Samsung
2 Trace:
3   at fail (Unknown)
4   at main (line: 5, column: 2)" evaluating expression: "%dw 2.0
5   output application/java
6   ---
7   if(vars.bname=='Samsung')
8     |dw::Runtime::fail("We dont Accept Samsung")
9   else
10    payload".
```

Also Observe that data first insert into product table is committed even though exception occurred in the flow.

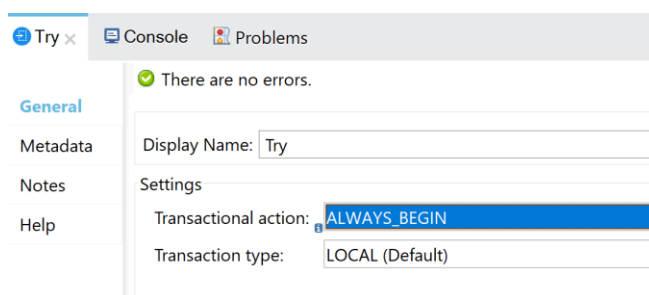
Now we want to run this 2 inserts in a single transaction. So, if exception occurs, transaction should be rolled back.

Now drag try scope after Http Listener and drag all other components into try scope.

The flow should look like below:



Click on Try block and configure as shown below:



Now Click on First Insert component and then click on advanced tab.



Select the transactional action as “ALWAYS\_JOIN”.

Similarly do for second insert component also.

Now restart the application and give a request to

[http://localhost:8081/save?name=Laptop&brand\\_name=Samsung&original\\_price=1000](http://localhost:8081/save?name=Laptop&brand_name=Samsung&original_price=1000)

You should get 500 as status code as brand\_name is Samsung.

Now you should observe that data inserted before exception is rolled back.

## **STEP2 – Global Transactions**

Download activemq from activemq website or use the one given by me

Run bin/win64/activemq.bat to start activemq

Go to <http://localhost:8161> and click on “Manage Activemq broker”. Login using admin/admin

Click on Queues link on the top. You should not see any queues.

In this step, you will be working on **12-global-transactiondemo** project

Open globaltransactionsdemo.xml and understand the flow.

When we get http request, flow expects a query parameter brand and jms publish component is publishing this brand value to a queue and insert component is inserting a row into product table. The transformer is throwing an error if brand is ‘Samsung’.

Deploy the application and give a request to <http://localhost:8081/save?brand=Hp>

You should see a message in queue “someq” and a row is inserted to product table.

Now give a request to <http://localhost:8081/save?brand=Samsung>

You should see that JMS message is committed and then error is thrown. Data is not inserted in database.

We want JMS Send and DB Insert to be in a single global transaction.

Click on try scope and configure Transactional Action as “ALWAYS\_BEGIN” and



“Transaction Type ” as XA.

Click on Publish component and select Advanced tab. Observe that Transactional Action as JOIN\_IF\_POSSIBLE.

Similarly click on Insert component and select Advanced tab. Observe that Transactional Action as JOIN\_IF\_POSSIBLE.

Now deploy the application and give a request to <http://localhost:8081/save?brand=Samsung>

You should get 500 status code and response body as “Cannot start XA transaction”

Observe the console in studio. You should see the below exception:

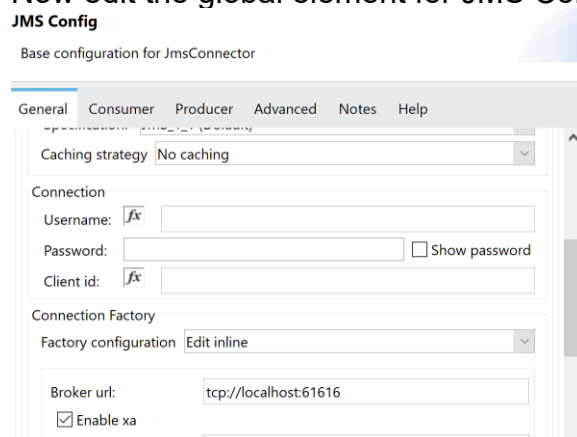
[java.lang.IllegalStateException](#): Object of type "javax.transaction.TransactionManager" with name "Transaction Manager" is not available in the registry

So, we need to configure Transaction Manager.

Click on Global Elements tab of xml , click on create and select “Bitronix Transaction Manager”.

When we want global transaction, we need to enable xa transaction at connector level.

Now edit the global element for JMS Config and select “Enable xa as shown below”



Now edit global element “Database Config “ and select Transactions tab  
Select “Use XA Transactions” check box as shown below:



#### Database Config

Default configuration

General

Advanced

Notes

Help

Name: db1

Connection: MySQL Connection

General

Transactions

Advanced

Connection

Transaction isolation: NOT\_CONFIGURED (Default)

☒ Use XA Transactions

Now redeploy and give a request to <http://localhost:8081/save?brand=Samsung>

U should not see any error.

Now give a request to <http://localhost:8081/save?brand=SOMEBRAND>

You should observe an error and both JMS and db transaction should have been rolled back. See the db table and JMS queue. You should not see any new records.