LAB- Handling Exceptions

In this lab, you will be working on **02-exceptions-start** project under **06-performance-errorhandling**

In this lab you will understand

a) How to Use Handle Exceptions using " Catch Exception Strategy" , "Rollback Exception Strategy", "Choice Exception Strategy","Reference Exception Strategy "

b) You will also understand how to configure your own exception strategy as default global strategy.

c) How to handle transactions

Step 1

Before you start this step, make sure that you have started ActiveMq.

Open ActiveMq console at http://localhost:8161 and delete any queues if present.

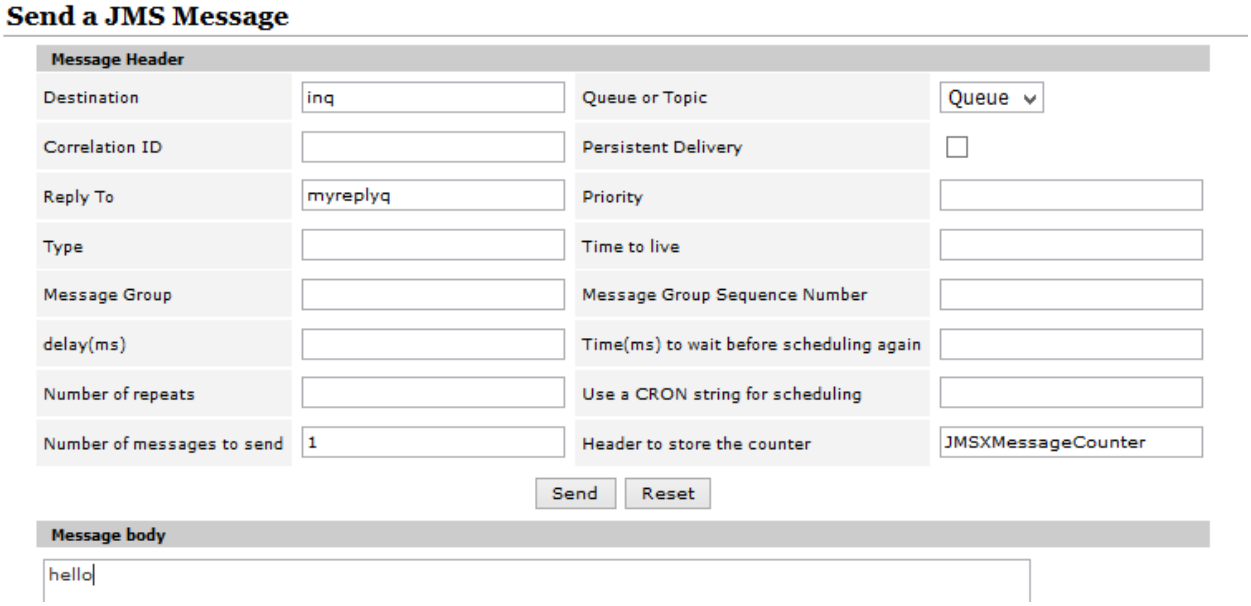
1) Open TestService.java and observe that the method `getMessage` throws exception

2)Open exceptions.xml and observe that JMS inbound endpoint is configured to receive message from "inq".

Observe all of the configuration in the flow.

Goto ActiveMq console and send a test message to "inq" with payload "Hello".

Make sure that you add ReplyTo Header as "myreplyq" as shown below :



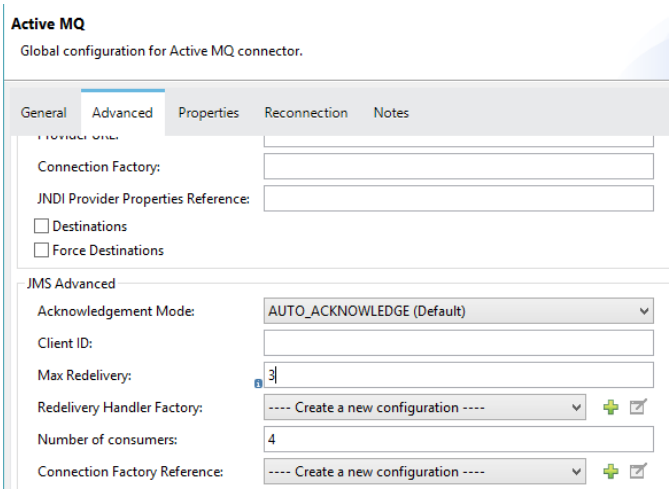
Observe the console that exception is thrown .

DefaultExceptionStrategy has handled it and dumped the stack trace.

Is Redelivery done?

No. Because you didnt configure it.

Now edit the ActiveMq Connector configuration and click on Advanced tab.
Modify the Max delivery to 3 and number of consumers as 1 as shown below :



Now deploy the application

Goto ActiveMq console and delete all the existing queue.

send a test message to "inq" with payload "Hello".

Make sure that you add ReplyTo Header as "myreplyq"

Observe the console that message has been redeilvered 3 times as per our configuration.

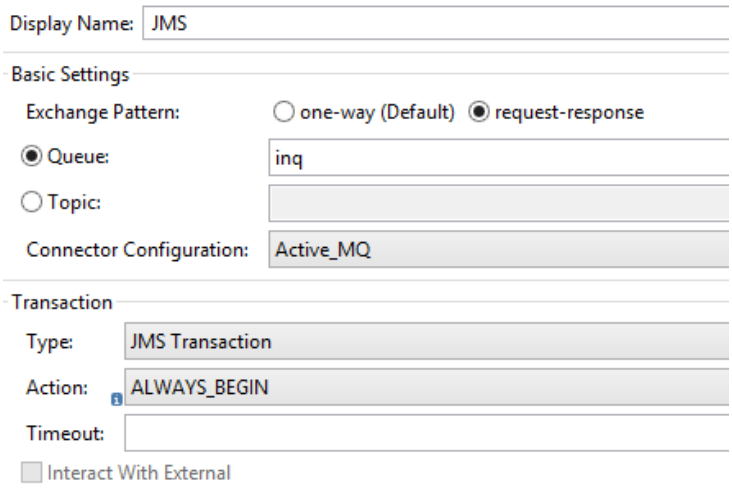
Can you guess how many messages will be in outq?

There will be 4 messages in outq even though our message is not processed successfully due to exception.

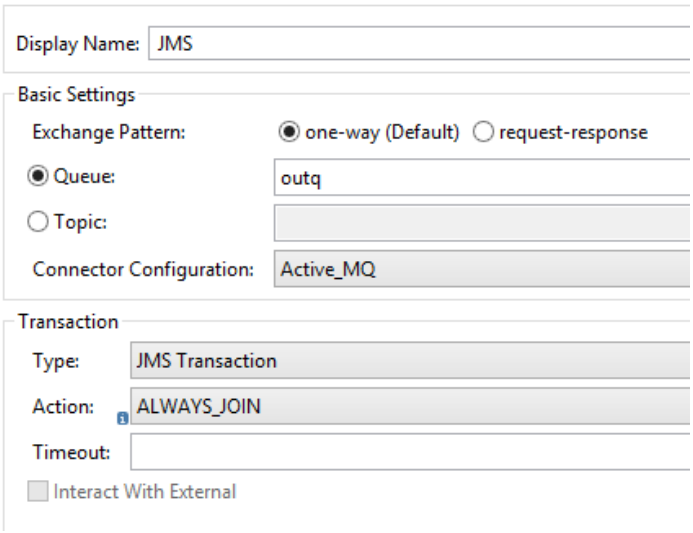
This is because we didn't configure out flow to run in a transaction.

We want to start a JMS transaction at the beginning of the flow and commit the transaction only if there are no exceptions on the flow. If there are exceptions in flow, transaction should be rolled back.

Double click on JMS inbound endpoint and configure Transaction Type as "JMS transaction " and Action as "ALWAYS_BEGIN" as shown below :



Double click on JMS Outbound endpoint and configure it to join transaction as shown below :



Now deploy the application .

Goto ActiveMq console and delete all the existing queues.

Send a test message to "inq" with payload "Hello".

Make sure that you add ReplyTo Header as "myreplyq"

You should observe that message is redelivered 3 times.

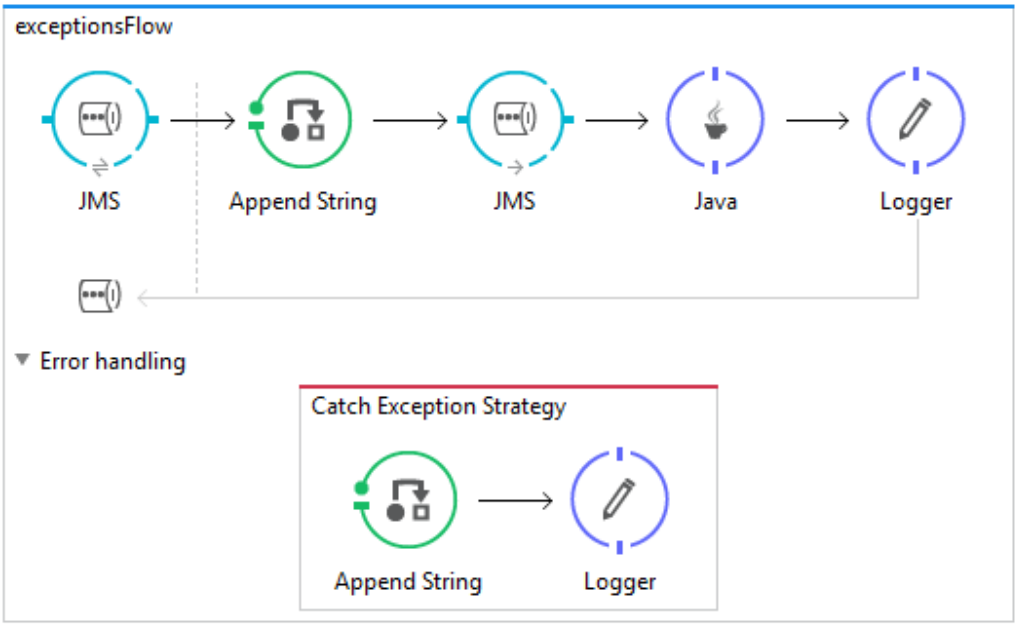
But this time, the number of messages in outq will be zero.

What about Respone? How many messages will be present in "myreplyq"?

WAY2LEARN

HANDLING EXCEPTIONS USING CATCH EXCEPTION STRATEGY

Drag a catch exception strategy into the flow and Drag Append String Transformer and Logger into it as shown below :



Now deploy the application .

Goto ActiveMq console and delete all the existing queues.

Send a test message to "inq" with payload "Hello".

Make sure that you add ReplyTo Header as "myreplyq"

Will there be redeliveries?

What about Reply?

Is the transaction commited?

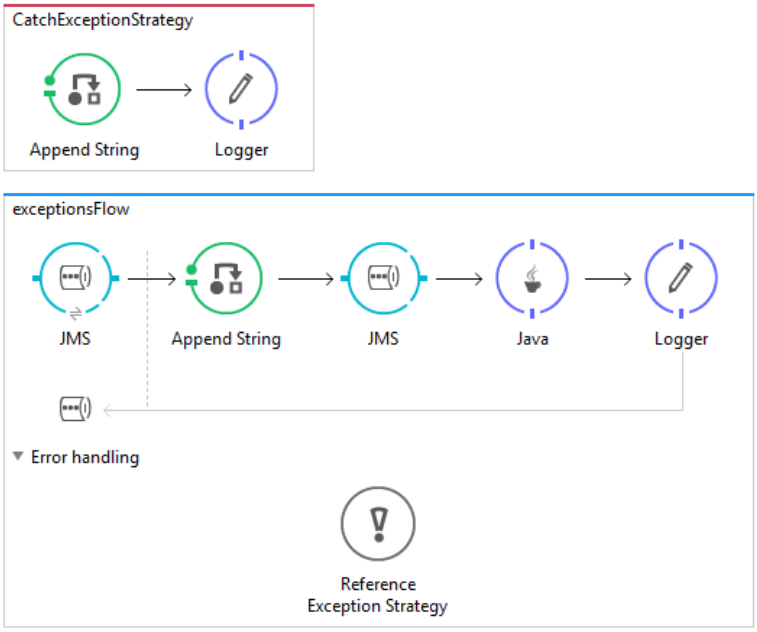
MAKING CATCH EXCEPTION STRATEGY AS GLOBAL STRATEGY

Drag the catch Exception strategy outside of Flow.

It shows error because name of catch exception strategy cannot contain spaces.

So, remove spaces.

Now Drag "Reference Exception Strategy" into the flow and reference the Global strategy .



Now deploy the application .

Goto ActiveMq console and delete all the existing queues.

Send a test message to "inq" with payload "Hello".

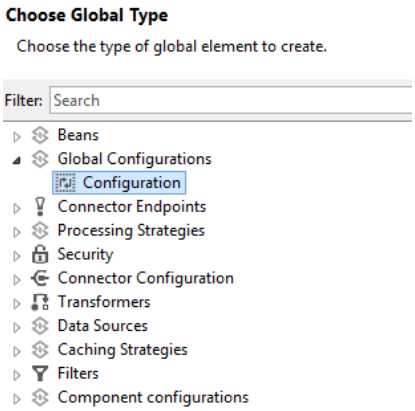
Make sure that you add ReplyTo Header as "myreplyq"

Observe that the global strategy is executed when exception occurs and the behaviour is same as earlier.

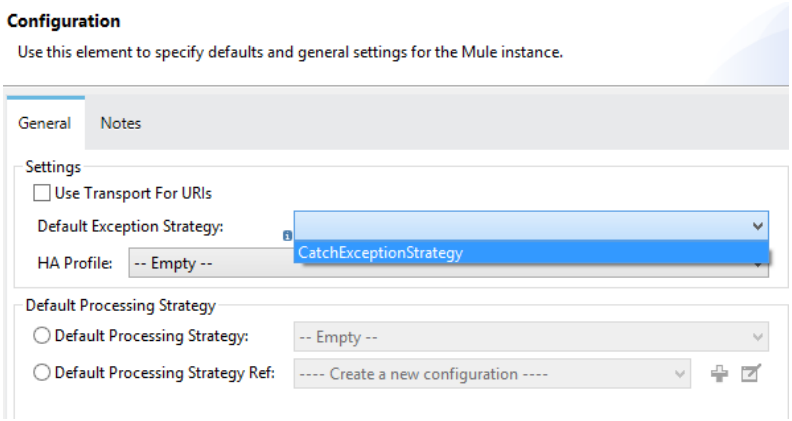
MAKING CATCH EXCEPTION STRATEGY AS DEFAULT GLOBAL STRATEGY

Delete the reference exception strategy from the flow.

Go to global elements tab and click on create . Under Global Configurations,select Configuration as shown below :



Configure default exception strategy as shown below :



Now deploy the application .

Goto ActiveMq console and delete all the existing queues.

Send a test message to "inq" with payload "Hello".

Make sure that you add ReplyTo Header as "myreplyq"

Is the default strategy executed?

Are redeliveries done?

What about transactions?

.

HANDLING EXCEPTIONS USING ROLLBACK EXCEPTION STRATEGY

We want redeliveries to be done and everytime before redelivery, transactions if any , should be rolled back.

So, delete Catch Exception strategy .

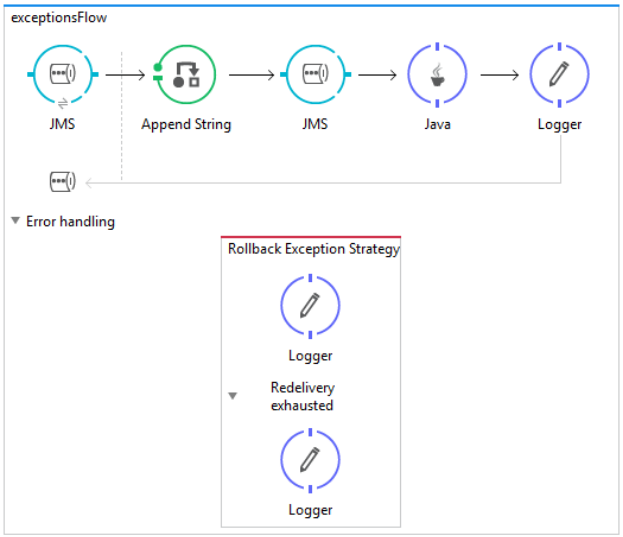
Also go to Global elements and delete the Configuration element,

Now Drag Rollback exception strategy in to the error handling block of the flow.

There are 2 sections in Rollback exception strategy.

Drag Logger into each section and configure them to log payload.

It should look like below :



Now deploy the application .

Goto ActiveMq console and delete all the existing queues.

Send a test message to "inq" with payload "Hello".

Make sure that you add ReplyTo Header as "myreplyq"

Are there redeliveries?

How many messages will be present in outq?

Will there be a reply message in "myreplyq"?

In this case after all redeliveries are exhausted, the payload in "redeliveries exhausted" section will be original ActiveMq Message.

If you dont want to lose message, you can drag a JMS endpoint and configure it to send the message to DLQ.