**QUEUE DATA TRANSMISSION**

Queue: A queue is a container for messages. Business applications that are connected to the queue manager that hosts the queue can retrieve messages from the queue or can put messages on the queue. A queue has a limited capacity in terms of both the maximum number of messages that it can hold and the maximum length of those messages.

Here we are using Active Mq broker for sending the messages into the queue and receiving the messages from the queue.

**Task:**

**->**Passing multiple records as Json payload and performing the addition on the data and assigning an unique UUID for the entire record and calculating the length of the entire record .

->Placing the entire data into the queue and receiving the data from the Queue. From the data taking UUID and length of the record and checking whether the data is present in the database or not and inserting the data into the database if the record does not exist until if the final record has been inserted .

**Requirement:**

1.Active Mq Broker

2.Oracle Database

**Application:**

->Place the Http connector at the source and configure the Http listener i.e its port and path

->Place the Transform message after Http listener for accepting the multiple records

->Generate the UUID for the record by declaring the set variable values as #[function:uuid]

->Take the length of the record and store it in a variable

->Since we want to pass one by one record we have taken for each loop and iterate the payload

->In for each loop we take transform message and calculate its sum and we add the uuid and length of the record as payload to it

->Now we place the entire Json payload into the queue. Configure the queue by giving its destination queue name

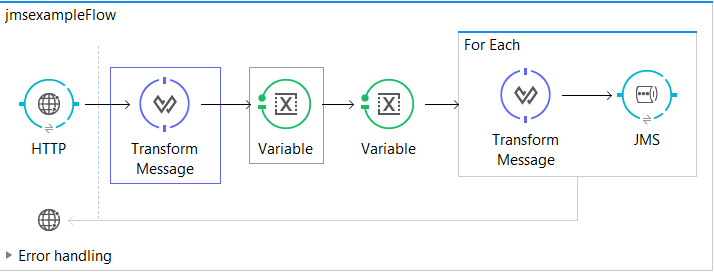


Fig: Flow for placing the Json payload into Queue

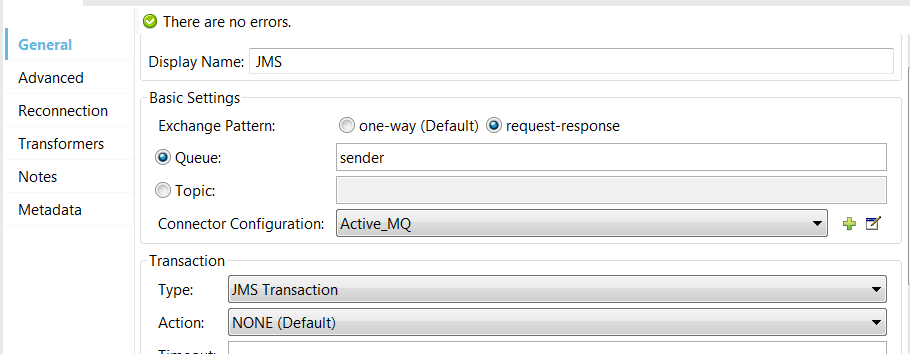


Fig: Configuration of JMS Sender Queue

->Place the Jms connector at the source for receiving the messages placed in the queue

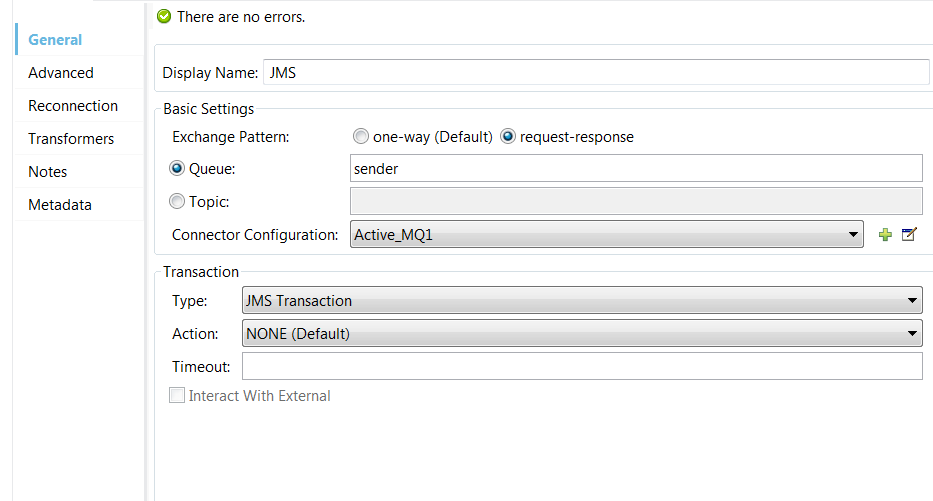


Fig: Configuration of JMS Receiver Queue

->We get the Byte Array payload from the queue. We have to convert it into Json payload by using Byte Array to Object transformer and then Object to Json transformer.

->Form the Json payload we take uuid and length and store the entire data into one variable

->We check whether the record is present in the database or not by using UUID

i.e select \* from db where uuid=#[flowVars.daa.uuid] (for example)

->From the database output we take the size of payload and store in a variable

->Using choice we are check whether the size of payload and total length of the records are equal are not. If they are not equal we are going into insert the entire record into the database

->This process is completed until the last record arrives. Once the last record arrives we are going retrieve the entire records from the database and store it in a file in Json Format.

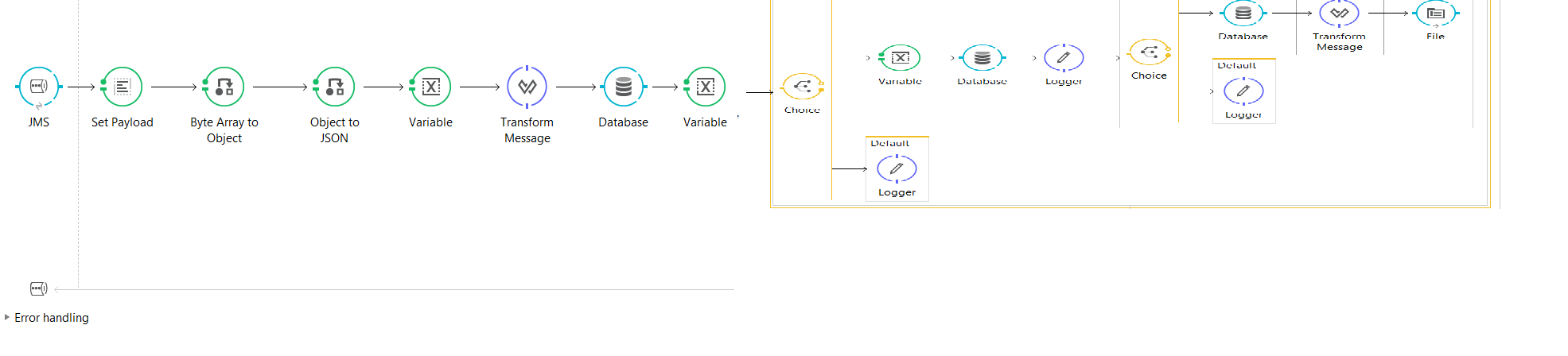


Fig: Receiving the payload from the queue and performing database operations

Sample Inputs:

I am passing the Json payload as the input from the Postman. Here is the sample Input

[

{

"a":19,

"b":20

},

{

"a":41,

"b":32

},

{

"a":1,

"b":2

},

{

"a":111,

"b":2222

},

{

"a":19,

"b":20

},

{

"a":41,

"b":32

},

{

"a":1,

"b":2

},

{

"a":111,

"b":2222

}]

Expected Output:

[

"{\n \"a\": 19,\n \"b\": 20,\n \"uuid\": \"2dfe8e80-2114-11e8-96c8-864b20524153\",\n \"sum\": 39,\n \"length\": 8\n}",

"{\n \"a\": 41,\n \"b\": 32,\n \"uuid\": \"2dfe8e80-2114-11e8-96c8-864b20524153\",\n \"sum\": 73,\n \"length\": 8\n}",

"{\n \"a\": 1,\n \"b\": 2,\n \"uuid\": \"2dfe8e80-2114-11e8-96c8-864b20524153\",\n \"sum\": 3,\n \"length\": 8\n}",

"{\n \"a\": 111,\n \"b\": 2222,\n \"uuid\": \"2dfe8e80-2114-11e8-96c8-864b20524153\",\n \"sum\": 2333,\n \"length\": 8\n}",

"{\n \"a\": 19,\n \"b\": 20,\n \"uuid\": \"2dfe8e80-2114-11e8-96c8-864b20524153\",\n \"sum\": 39,\n \"length\": 8\n}",

"{\n \"a\": 41,\n \"b\": 32,\n \"uuid\": \"2dfe8e80-2114-11e8-96c8-864b20524153\",\n \"sum\": 73,\n \"length\": 8\n}",

"{\n \"a\": 1,\n \"b\": 2,\n \"uuid\": \"2dfe8e80-2114-11e8-96c8-864b20524153\",\n \"sum\": 3,\n \"length\": 8\n}",

"{\n \"a\": 111,\n \"b\": 2222,\n \"uuid\": \"2dfe8e80-2114-11e8-96c8-864b20524153\",\n \"sum\": 2333,\n \"length\": 8\n}"

]