

LAB- Event Processing models

NOTE: **You will be able to understand this lab documentation after you listen my explanation directly or in my video. If not, it may not make sense to you. Please see the videos in my youtube channel (@zerotosiva) before you proceed with this lab.**

SEQUENTIAL PROCESSING

In this exercise, you have to work on **01-sequential-parallel-asynch**

So, import the corresponding jar given to you

Open 01-sequential.xml and understand the flow.

Http listener listening at /addproductssequential is expecting query parameters id, brand and price.

The next Transform Message component is creating a product as Json payload using the query parameters.

We cant to upsert the product into database and send it to soap webservice

Go through each component and understand the flow.

In this flow we are doing sequential processing. Until data is upserted to database, soap webservice call doesn't happen.

Inside upsert product flow, observe the last logger and understand that we are adding a 2000 ms delay. In the soap flow we are adding a delay of 4000 ms. So, as we are processing sequentially, this flow will take minimum 6000 ms for giving response.

Execute soap.bat to start the soap webservice as this flow will make request to this soap webservice

Run this flow and give a request to

<http://localhost:8081/addproductsequential?id=100&brand=Hp&price=2000>

Observe the response time. It should be more than 6000 ms (around 8000 ms)

PARALLEL PROCESSING

As adding a product by calling soap webservice is not dependent on database insert, we can execute them parallelly using scatter-gather

In this exercise, you will be working in same project **01-sequential-parallel-async**

Open 02-parallel.xml and understand how scatter and gather is used.

Give a request to this flow using <http://localhost:8081/addproductparallel?id=200&brand=Hp&price=2000>

Observe that you get response in around 4000 ms

ASYNCHRONOUS PROCESSING

If we don't want the main flow to wait until we get response from soap webservice, we can make soap webservice call asynchronously

In this exercise, you will be working in same project **01-sequential-parallel-async**

Open 03-async.xml and understand how async scope is used.

Give a request to this flow using <http://localhost:8081/addproductasync?id=400&brand=Hp&price=2000>

Observe that you get response in around 2000 ms

IDEMPOTENT PROCESSING

In the above flow using scatter gather, what if one of the route which is make soap call fails?

We can handle the error and send an error response to client so that client can

retry the same operation

If client retries same operation, for database route, it might be a duplicate data .
We don't want to process same data again.

So, we want to make sure that our logic is idempotent

Open 04-idempotent.xml and observe how we are using Idempotent Message validator.

Run it and give a request to <http://localhost:8081/addproductidempotent?id=500&brand=Hp&price=2000>

For the first request, data should be inserted in database. If u give same request again with same data, you should get below response

"Duplicate Message !!! Product was already added"

Quick Acknowledgement Pattern

In this exercise, you will be working in project **02-synchronous-quick-acknowledgement**

So, import the corresponding jar given to you

Open synchronous-quick-acknowledgement.xml and observe flow with name "mainflow".

Observe how we are sending quick acknowledgement and posting message to JMS queue.

Observe the logic in "processingflow".

Observe that we have added a delay of 30 seconds in the logger component to simulate delay.

Observe how the product processing status is updated in database once processing is done successfully

Start Active MQ broker.

Now start the application and give a request to
<http://localhost:8081/addproduct?id=500&brand=Hp&price=2000>

You will get a quick acknowledgement with callbackURL in the response

Give a request to callback url and observe that status is “processing”.

If u give a request to callback url after 30 seconds, you will see that the status is “processed”

Using foreach and parallel foreach

In this exercise, you will be working in project **03-iterativeprocessing-foreach-parallel**

So, import the corresponding jar given to you

Firstly, let us try to understand about foreach.

Open 01-using-foreach.xml and understand the logic in the flow.

Keep a break point on the logger after for each scope

Debug the application and Copy src/main/resources/products.csv under c:\files\input folder

Once the application stops at break point, observe that the payload after foreach is same as the payload before foreach. Also, observe that “vars” which was updated in foreach scope has all the modified products.

So, you understood that variables added in the foreach scope will be available after foreach. But payload will not be modified. Also, the processing is sequential.

Can u tell what is the purpose of “batch Size”, “counter variable name “ and “collection” properties of foreach

Now let us try to understand about “parallel foreach”

Open 02-using-parallel-foreach.xml and observe the logic.
Keep a break point on the logger after parallel foreach .

Debug the application and Copy src/main/resources/products.csv under
c:\files\inputparallel folder

Once the application stops at break point, observe that the variables which we added inside parallel foreach are not available and payload is array of individual modified mule messages

So, you understood that , after parallel foreach, changes made to the variables are lost and payload will be array of mule messages.

Can you tell what is the purpose of “Max concurrency” in the configuration of parallel foreach ?

Using batch job

Open 03-batchprocessing.xml and understand the logic.

Run the the application and Copy src/main/resources/products.csv under
c:\files\inputbatch folder

Observe the logs and analyze the behaviour of batch job.

Can you tell the purpose of Batch Aggregator in step 1?

Can you tell why I didn't use batch aggregator in step2 ?

What is the purpose of configuring “accept expression” in step2?

What is the payload logged by the logger in “on complete” phase?