LAB- Automatic and Manual Watermarking using object store

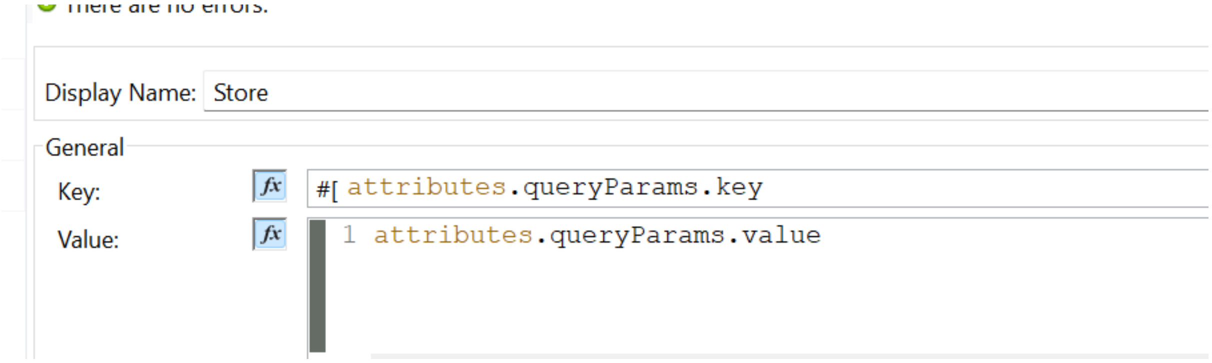
STEP1 – Using Object store to store and retrieve data

Create a project with name objectstoredemo

Drag a Http Listener and configure it to listen at 8081 and path /store

In the Mule palette, add Objectstore module

Drag "Store" Component from Object store module after Http Listener and configure key and value as shown below:



There are no errors.

Display Name: Store

General

Key: #[attributes.queryParams.key

Value:

```
attributes.queryParams.value
```

Rename the flow as "StoreFlow"

Drag Set payload component after Store component and set pay load as "Data Store in Object store"

You are done configuring a flow to store data to object store

Deploy the application and give a request using below URLs:

localhost:8081/store?key=a&value=1

localhost:8081/store?key=b&value=2

So, We store 2 key/value pairs in object store

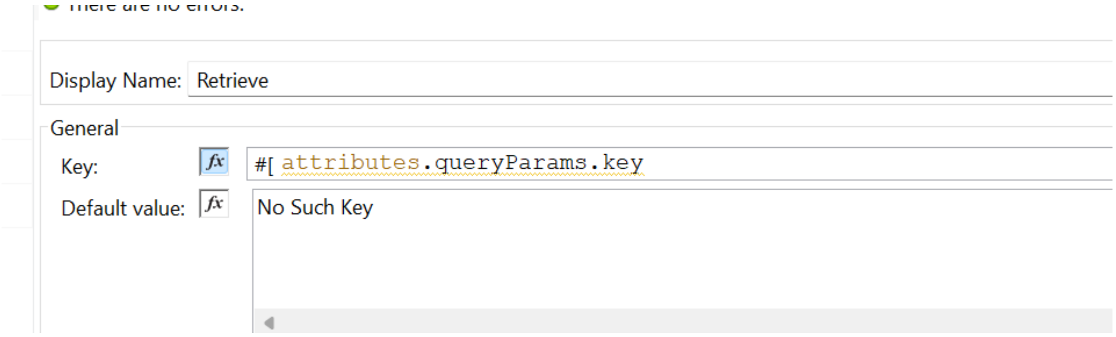
Stop the application.

Now let us configure another flow to retrieve values from object store

Drag a Http Listener into another flow and configure it to listen on 8081 and path /retrieve .   Rename the flow as "retrieveflow"

Drag "Retrieve" component from Object store module after Http Listener

Configure Retrieve component as shown below:



There are no errors.

Display Name: Retrieve

General

Key: #[attributes.queryParams.key

Default value: No Such Key

Create a project with name **01-watermarking-start**

Now run the application and give a request to localhost:8081/retrieve?key=a

You should able to see the value of key "a" which you stored earlier.

Give a request to retrive the value of a key which doesn't exist in Object store. You should see response as "No Such Key"

Try configuring the object store as In Memory Object Store.

Give a request to localhost:8081/store?key=a&value=1

Now, restart the application and try to retrive the value of key "a"

You should see response as "No Such Key"

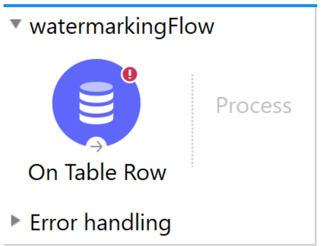
Congratulations!! You understood how to use Object Store

STEP2 – Understanding Automatic watermarking done by "On TableRow"

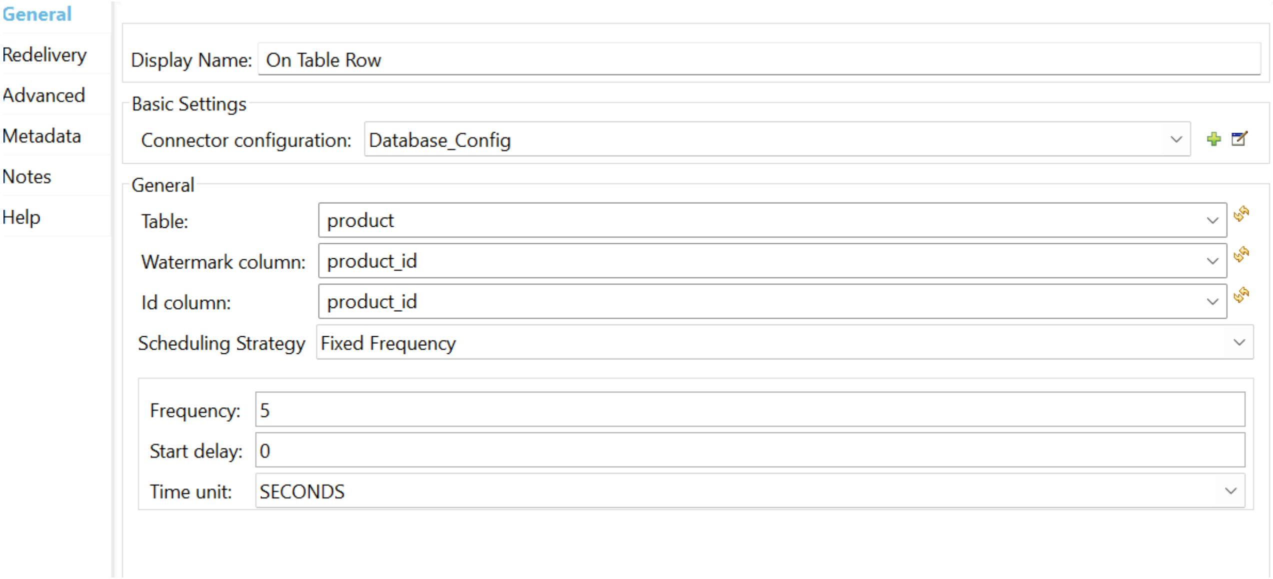
Create a Mule Configuration file with name watermarking.xml .

In the " Mule palette" Add Database module.

Now Drag and drop "On Table row " component .The flow should appear as shown below:



Now Configure On Table row as shown below (Create Database Config just like u have done earlier)



Display Name: On Table Row

Basic Settings

Connector configuration: Database_Config

General

Table: product

Watermark column: product_id

Id column: product_id

Scheduling Strategy Fixed Frequency

Frequency: 5

Start delay: 0

Time unit: SECONDS

Drag “Transform Message” component after “On Table row” and configure it to convert payload to json.

Now drag and drop a logger after “Transform Message” and configure it to log payload.

You should be able to see a log message for every record in the database

Go to MySql workbench and add a row in product table.

You should observe that the new record which you added is logged on the console

If you stop and run the application again, is it fetching the earlier records in database again?

NO.

So, the “On Table row ” component is able to remember about the previously processed records using watermarking.

STEP3 - Custom WaterMarking using Object store connector module

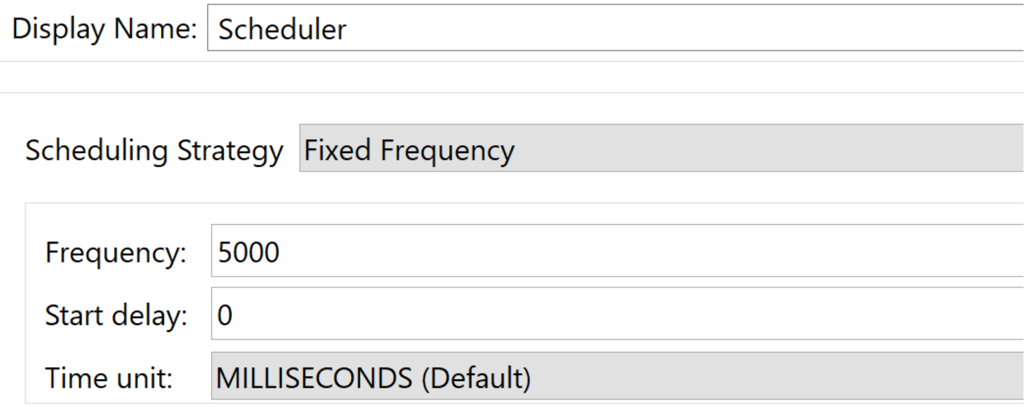
In this lab, you will understand how to use object store to create your own watermark

Create a new project with name “CustomWatermarking”

We want a scheduler to kick off the flow after every 5 secs.

After scheduler triggers a Mule event, we want to check if there is a key lastproductid in object store. If not present, we want to initialize a variable lastproductid with 0. Else if present, lastproductid variable should be initialized with the value from object store.

Now, First drag a scheduler and configure it as shown below:



Display Name: Scheduler

Scheduling Strategy: Fixed Frequency

Frequency: 5000

Start delay: 0

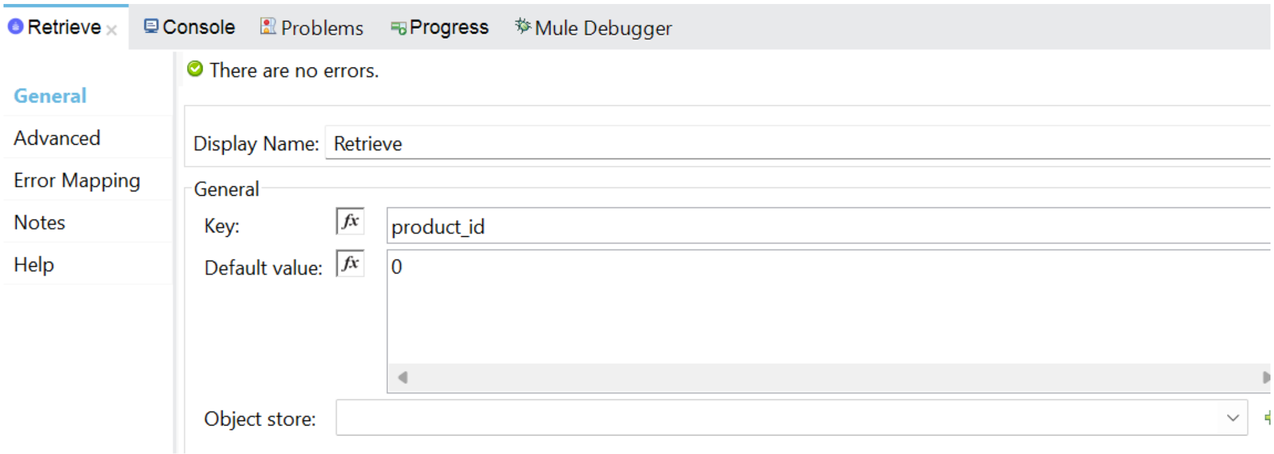
Time unit: MILLISECONDS (Default)

Add Object Store connector module and select it

Now, drag and drop “retrieve” component after Scheduler.

Configure “retrieve” component as shown below:

Configure key as productid and default value as 0.



There are no errors.

Display Name: Retrieve

General

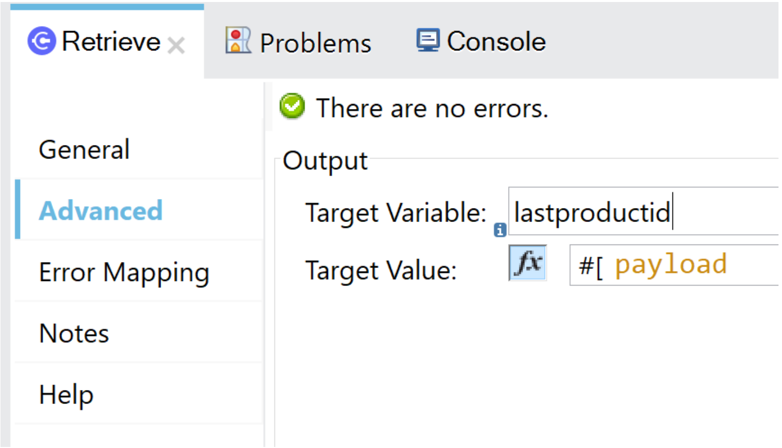
Key: product_id

Default value: 0

Object store:

We want the output of “Retrieve” component to be stored in a variable with name lastproductid.

So, click on Advanced tab and give the variable name as shown below:



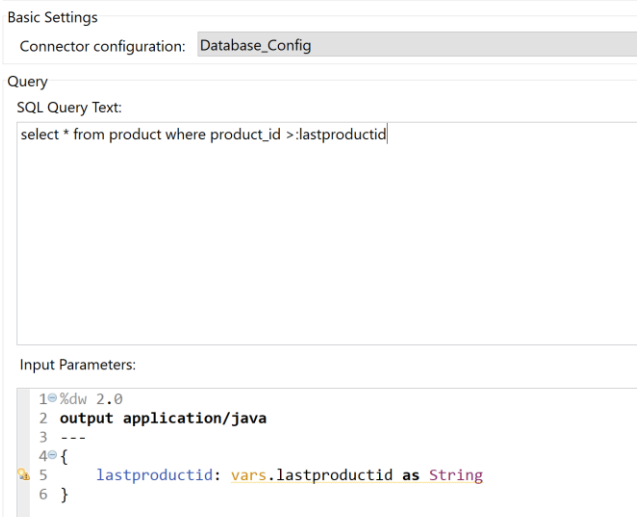
There are no errors.

Output

Target Variable: lastproductid

Target Value: #[payload

Now drag “Select” component of database module and configure it with query as shown below:



Basic Settings

Connector configuration: Database_Config

Query

SQL Query Text:

```
select * from product where product_id >:lastproductid
```

Input Parameters:

```
%dw 2.0
output application/java
---
{
    lastproductid: vars.lastproductid as String
}
```

If there is data from database, we want to transform it to json and log it. After that we want to store the max value of product_id in object store with key product_id.

If there is no data from database, we just want write a log as “No new records!!”

So, drag a choice router after select component.

In the default block of choice router , drag a logger and configure it to log “No New records”

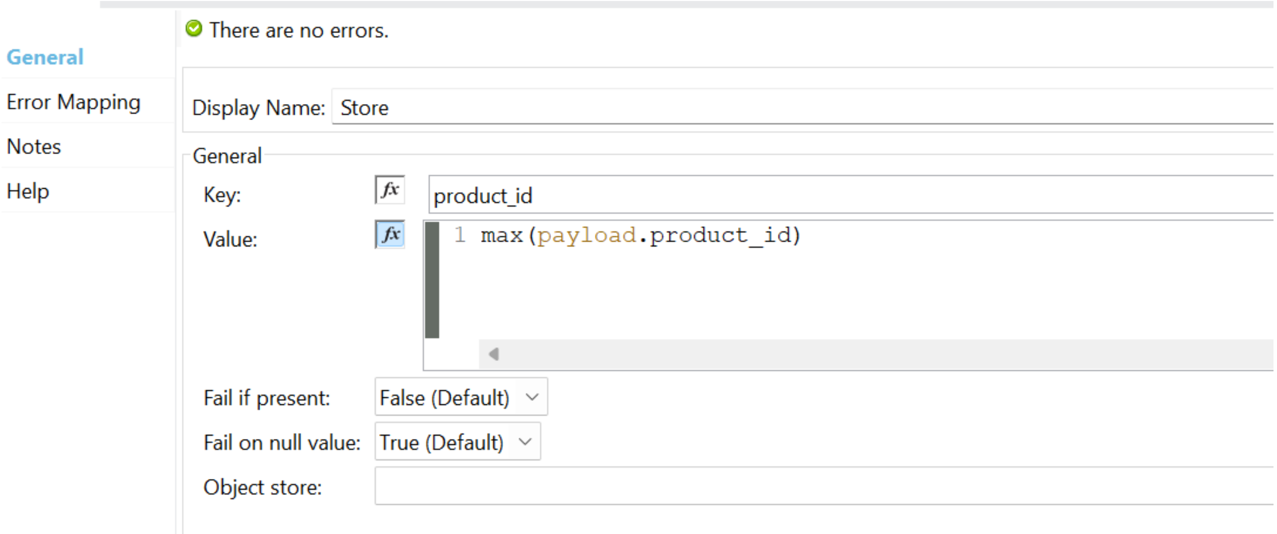
Configure the first route of choice router with condition `sizeOf(payload) >0`

Drag Transform Message component inside first route to transform the payload to json.

Drag a logger after transform message to log the payload

Now, drag store component from Object store connector module

Configure “Store” component properties as shown below :



There are no errors.

Display Name: Store

General

Key: product_id

Value:

```
max(payload.product_id)
```

Fail if present: False (Default)

Fail on null value: True (Default)

Object store:

Now run the application and observe that all the records are logged on console for first time when scheduler runs.

After first time you should observe that “No New Records” will be logged.

This is the end of the Exercise