LAB- Error Handling

**STEP1**

In this lab, you will be working on **01-exceptions-start** project under **08-errorhandling** working set.

1) Open exceptions-demo.xml

2) Observe that there are 3 flows in it with names MainFlow, ValidateAndTransform, and getproductsbyname.

In ValidateAndTransform flow, there is “IsTrue” validator which validates the expression `#("Hp","Mac") contains vars.productname` and throws exceptions if the value of query param productname is not Hp or Mac.

The error message thrown by “IsTrue” is “Invalid ProductName!!!!”

In postman, give a request to [http://localhost:8081/error](http://localhost:8081/error) without passing queryparameter productname.

You will see “Invalid ProductName!!” as Response ans status code is 500.

This is because in Http Listener, error response body is configured with `output text/plain --- error.description`

Now, Click on responses tab in properties of HttpListener and change the error response body as

`output application/json --- error.errorType`

Now give a request to [http://localhost:8081/error](http://localhost:8081/error) with out passing queryparameter productname.
You should see the error type displayed on browser as

```
{
  "parentErrorType": {
    "parentErrorType": {
      "parentErrorType": null,
      "namespace": "MULE",
      "identifier": "ANY"
    },
    "namespace": "MULE",
    "identifier": "VALIDATION"
  },
  "namespace": "VALIDATION",
  "identifier": "VALIDATION"
},
  "namespace": "VALIDATION",
  "identifier": "INVALID_BOOLEAN"
}
```

By seeing the error type above we can understand that the error thrown is “VALIDATION:INVALID_BOOLEAN”. Its parent is “VALIDATION:VALIDATION” whose parent is “MULE:VALIDATION” and its parent is “MULE:ANY”.

3) Now, we want to handle exception in “ValidateAndTransform” flow.

Drag and drop “On Error Continue” inside error handling block of “ValidateAndTransform” flow. Drag a Logger inside it and configure it to log “Inside on Error continue of ValidateAndTransform flow”.

Using postman, Now give a request to http://localhost:8081/error with out passing queryparameter productname

Now observe the console for the log message from ValidateAndTransform flow and also the log message from main flow. Observe that status code is 200.

Now Delete “On Error Continue” from “ValidateAndTransform” flow.

Drag and drop “On Error Propagate” into ValidateAndTransform flow. Drag a logger into it and configure it to log “Inside on error propagate of ValidateAndTransform”.

Start the application.
From postman, give a GET request to http://localhost:8081/error with out passing query parameter productname.

You should observe that Logger is not executed in mainflow and error type is displayed on the browser.

Observe the status code is 500.

Now make sure that soap.bat is running.

In postman, give a request to http://localhost:8081/error?productname=Hp observe that you get proper response from Soap Webservice.

Now, Stop the Soap Webservice and give a request to http://localhost:8081/error?productname=Hp you should see the error type displayed as shown below:

```
{
  "parentErrorType": {
    "parentErrorType": null,
    "namespace": "MULE",
    "identifier": "ANY"
  },
  "namespace": "WSC",
  "identifier": "CANNOT_DISPATCH"
}
```

Now Drag “On Error Continue ” inside errorhandling block of getproductsbyname flow. Drag “Setpayload” transformer inside it and set the payload as ‘In getproductsbyname after error from webservice----on error continue’

Drag a logger after “SetPayload” and configure it to log “In getproductsbyname after error from webservice----on error continue”

Now, Give a request to http://localhost:8081/error?productname=Hp you should see the following output

```
In getproductsbyname after error from webservice----on error continue
```
Now, Double click on Consume component, click on “Error Mapping” and configure as shown below (Make sure that error identifier(SIVACONNECTIVITYERROR in this case) is given in capitals only)

Now Drag “On Error Propagate” above “On Error Continue”.

Now Double click on “On Error Propagate” and configure it as shown below:

Drag “Set Payload” inside “On Error Propagate” and configure it with below payload:

'In getproductsbyname after error from webservice----on error Propagate error type: APP:SIVACONNECTIVITYERROR'

Drag a logger after “SetPayload” and configure it to log following:

'In getproductsbyname after error from webservice----on error Propagate'

Now run the application and Give a request to http://localhost:8081/error?productname=Hp you should see the following output
Drag “Raise Error” after logger and configure it as shown below:

```
{
    "identifier": "SIVACONNECTIVITYERROR",
    "parentErrorType": {
        "identifier": "ANY",
        "parentErrorType": null,
        "namespace": "MULE"
    },
    "namespace": "APP"
}
```

Now run the application and give a request to http://localhost:8081/error?productname=Hp you should see the following output

```
{
    "identifier": "SIVACUSTOMERROR",
    "parentErrorType": {
        "identifier": "ANY",
        "parentErrorType": null,
        "namespace": "MULE"
    },
    "namespace": "APP"
}
```

Now we want to configure a Global Error Handler.
Drag Error Handler on the top of all the flows.
Now drag “On Error Continue” inside it.

Drag a logger inside “On Error Continue” and configure it to log “In Global Error Handler”
Also Drag “Set Payload” inside it and set payload as “In Global Error Handler”

We want to make it default Error Handler for all flows.

Now click on Global Elements tab on the xml

Click on Create. Select “Configuration” under “Global Configurations”

Select your Exception handler as default error handler

Now restart the application and give request to
http://localhost:8081/error?productname=Hp

Observe that u get the response payload what ever has been set in global error handler
This is the end of the Exercise