LAB- Munit

STEP1

In this lab, you will be working on **01-munit-start** project under **10-munit** working set

1) Open 01-munit-demo.xml and observe the code.

Run the application and give a request to http://localhost:8081/test?url_key=payload_1

You should get response_payload_1 as response.

Now we want to write a Functional Test case for this scenario.

Right Click on 01-munit-demoFlow and select “MUNIT->create new bank test for this flow”

You should see that 01-munit-demo-test-suite.xml is created under src/test/munit folder

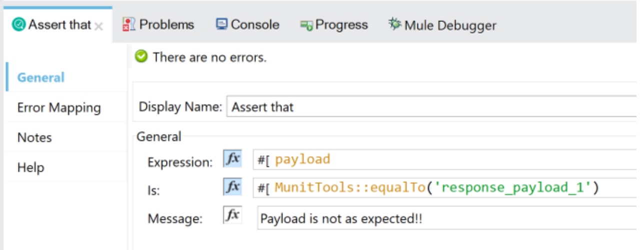
Select Munit module in the Mule palette and drag “SetEvent” before flow reference.

We want to create a event with empty payload and attributes should contain a query parameter url_key with value payload_1

Configure `#[{'queryParams':{'url_key':'payload_1'}}]` as value for attributes

We want to Assert that the payload after invoking the 01-munit-demoFlow as ‘response_payload_1’

So, drag “AssertThat” after Flow reference and configure it as shown below :



There are no errors.

Display Name: Assert that

General

Expression: #[payload

Is: #[MunitTools::equalTo('response_payload_1')

Message: Payload is not as expected!!

Right click on the TestSuite file and select “Run Munit Suite”

You should observe that test case is successful.

Change value of attributes to #[{'queryParams':{'url_key':'payload_2'}}]

Now run the test suite and observe that it fails.

2) We want to write another test case which will verify that “subflow1” is called.

Open 01-munit-demo.xml. Right Click on 01-munit-demoFlow and select “MUNIT->create new 01-munit-demoFlow suite”

One more Test case should be added to same suite xml file.

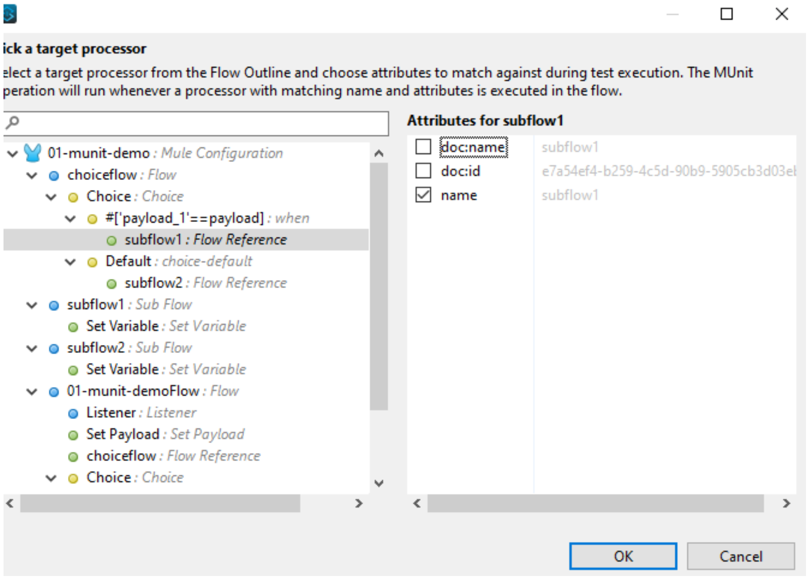
Select Munit module in the Mule palette and drag “SetEvent” before flow reference.

We want to create a event with empty payload and attributes should contain a query parameter url_key with value payload_1

Configure #[{'queryParams':{'url_key':'payload_1'}}] as value for attributes

Now Drag “VerifyCall “ in the Validation part

Click on Pick Processor button and select “subflow1” flow reference and select name attribute as shown below



ick a target processor

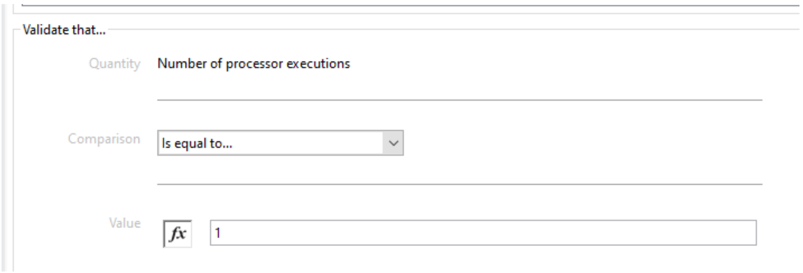
elect a target processor from the Flow Outline and choose attributes to match against during test execution. The MUnit peration will run whenever a processor with matching name and attributes is executed in the flow.

- 01-munit-demo : Mule Configuration
  - choiceflow : Flow
    - Choice : Choice
      - #['payload_1'==payload] : when
        - subflow1 : Flow Reference
      - Default : choice-default
        - subflow2 : Flow Reference
  - subflow1 : Sub Flow
    - Set Variable : Set Variable
  - subflow2 : Sub Flow
    - Set Variable : Set Variable
  - 01-munit-demoFlow : Flow
    - Listener : Listener
    - Set Payload : Set Payload
    - choiceflow : Flow Reference
    - Choice : Choice

Attributes for subflow1

☐ doc:name	subflow1
☐ doc:id	e7a54ef4-b259-4c5d-90b9-5905cb3d03el
☑ name	subflow1

Configure it to validate the number of processor executions is equal to 1 as shown below:



Validate that...

Quantity: Number of processor executions

Comparison: Is equal to...

Value: 1

Run the suite and make sure that both the test cases pass

3)If you have observed, both the test cases have “SetEvent” as same.

We want this “Set Event ” to be called before every Test case.

So, drag “Before Test” from the Mule palette and drag “set Event” into it.

Make sure that both the flow don’t contain Set event now.

Run the suite and make sure that both the cases are passed.

4) Till now, we have written Integration Tests. Now, we want to write Unit Test case

for 01-munit-demoFlow

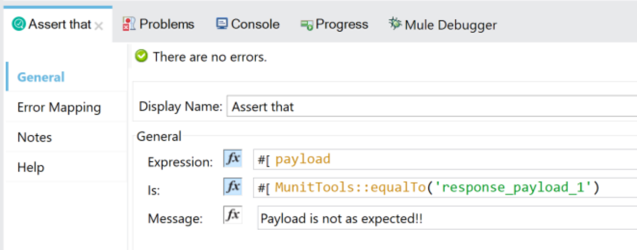
When we write Unit test cases, all the external calls to other flows have to be mocked.

Open 01-munit-demo.xml. Right Click on 01-munit-demoFlow and select "MUNIT->create new 01-munit-demoFlow suite"

One more Test case should be added to same suite xml file.

We want to Assert that the payload after invoking the 01-munit-demoFlow as 'response_payload_1'

So, drag "AssertThat" after Flow reference and configure it as shown below :



There are no errors.

Display Name: Assert that

General

Expression: #[payload

Is: #[MunitTools::equalTo('response_payload_1')

Message: Payload is not as expected!!

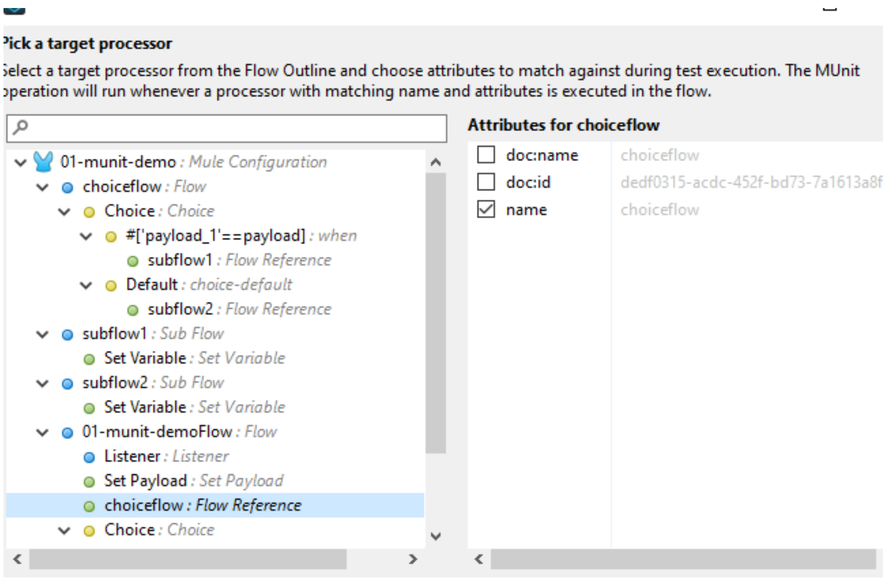
Now drag "Mock when" in to Behaviour part of Test case.

We want to mock the call to 'choiceflow'.

Actually, invoking choice flow with payload 'payload_1' results in setting a variable 'myvariable' with value 'myvariable1'.

So, let us mock this behaviour

Click on Pick Processor and select flow reference with name "choiceflow" as shown below



Pick a target processor

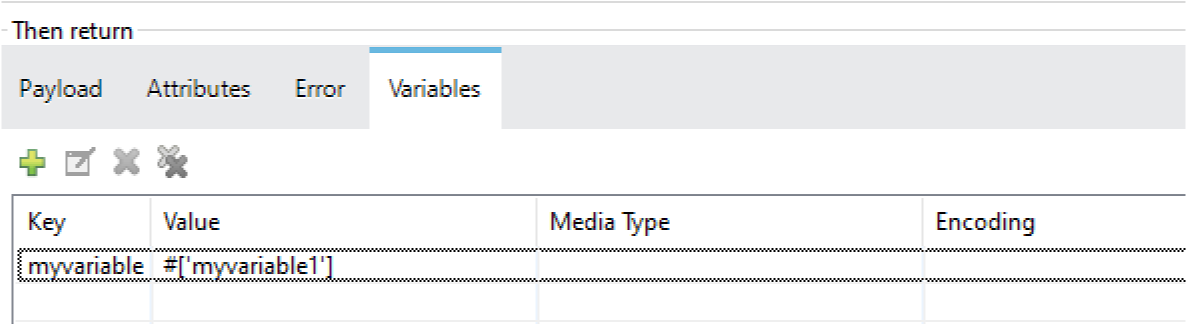
Select a target processor from the Flow Outline and choose attributes to match against during test execution. The MUnit operation will run whenever a processor with matching name and attributes is executed in the flow.

```
01-munit-demo : Mule Configuration
  choiceflow : Flow
    Choice : Choice
      #['payload_1'==payload] : when
        subflow1 : Flow Reference
      Default : choice-default
        subflow2 : Flow Reference
  subflow1 : Sub Flow
    Set Variable : Set Variable
  subflow2 : Sub Flow
    Set Variable : Set Variable
  01-munit-demoFlow : Flow
    Listener : Listener
    Set Payload : Set Payload
    choiceflow : Flow Reference
    Choice : Choice
```

Attributes for choiceflow

□ doc:name	choiceflow
□ doc:id	dedf0315-acdc-452f-bd73-7a1613a8f
☑ name	choiceflow

Inside “Then return ” section, select variables tab and configure a variable myvariable as shown below:



Then return

Key	Value	Media Type	Encoding
myvariable	#['myvariable1']		

Thas all. Now you mocked the call to choice flow.

Run the suite and make sure that all the test cases pass.

This is the end of the Exercise