Munit Implementation Document

|  |  |  |  |
| --- | --- | --- | --- |
|  | Prepared By | Reviewed By | Approved By |
| Name | Himanshu | Himanshu |  |
| Signature | Himanshu | Himanshu |  |
| Date | 01/28/2019 |  |  |

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Description | Author | Reviewer |
| 01/28/2019 | 1.0 | Initial Version Created | Himanshu | Himanshu |

Contents

[1. Objective 3](#_Toc536481049)

[Appendix 5](#_Toc536481050)

[Creating an MUnit Test Case 5](#_Toc536481051)

# Objective

Our objective is loading expected XML data from a file and comparing it with actual MUnit payload which we got from the fulfilment\_xpression\_isl\_muleFlow flow.

**Steps**

1. Open Mule project (fulfillment\_xpression\_isl\_mule) using Anypoint Studio.
2. Create a MUnit test case for the flow fulfilment\_xpression\_isl\_mule.
3. Inside test/resources, create a XML Request file which holds the sample XML payload.
4. In your MUnit TestCase, add a flow ref to call **fulfilment\_xpression\_isl\_muleFlow**.
5. After the flow ref, create a message payload and load the expected json data from the file using the following MEL: *#[getResource('UHOne\_Dental\_Gen3\_Contract\_Certificate\_20181218\_004555\_37.xml').asStream()]*
6. Now, in your MUnit test case, drag and drop a assert-on-equals component after the flow-ref.



1. Run your MUnit test; it will return a Test Case success if both JSON files/data are identical.

Refer to the code below for the full Flow and MUnit code.

**MUnit Test Case:**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<mule xmlns=*"http://www.mulesoft.org/schema/mule/core"* xmlns:doc=*"http://www.mulesoft.org/schema/mule/documentation"* xmlns:munit=*"http://www.mulesoft.org/schema/mule/munit"* xmlns:spring=*"http://www.springframework.org/schema/beans"* xmlns:core=*"http://www.mulesoft.org/schema/mule/core"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://www.mulesoft.org/schema/mule/munit http://www.mulesoft.org/schema/mule/munit/current/mule-munit.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-current.xsd*

*http://www.mulesoft.org/schema/mule/core http://www.mulesoft.org/schema/mule/core/current/mule.xsd"*>

<munit:config name=*"munit"* doc:name=*"MUnit configuration"*/>

<spring:beans>

<spring:import resource=*"classpath:fulfilment\_xpression\_isl\_mule.xml"*/>

</spring:beans>

<munit:test name=*"fulfilment\_xpression\_isl\_mule-test-suite-fulfilment\_xpression\_isl\_muleFlowTest"* description=*"Test"*>

<munit:set payload=*"#[getResource('UHOne\_Dental\_Contract\_Certificate\_20181228\_054900\_49.xml').asStream()]"* doc:name=*"Set Message"*>

<munit:inbound-properties>

<munit:inbound-property key=*"originalFilename"* value=*"UHOne\_Dental\_Contract\_Certificate\_20181228\_054900\_49.xml"*/>

</munit:inbound-properties>

</munit:set>

<flow-ref name=*"fulfilment\_xpression\_isl\_muleFlow"* doc:name=*"Flow-ref to fulfilment\_xpression\_isl\_muleFlow"*/>

<munit:assert-on-equals expectedValue=*"#[getResource('UHOne\_Dental\_Contract\_Certificate\_response.xml').asString()]"* actualValue=*"#[payload]"* doc:name=*"Assert Equals"*/>

</munit:test>

</mule>

**Files:**

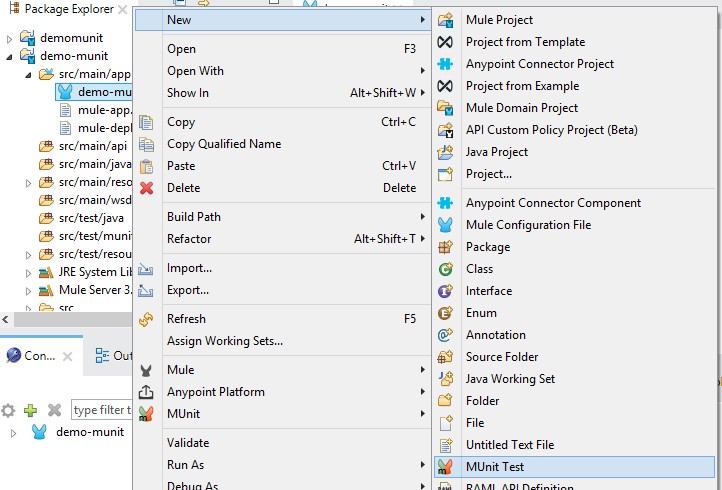
Sample Input :  Response :

## Appendix

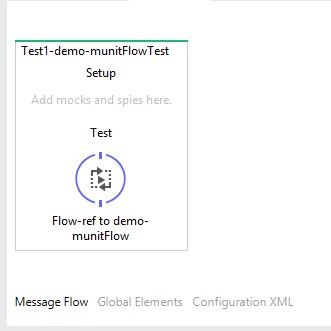
## Creating an MUnit Test Case

We will create an MUnit test case as follows.

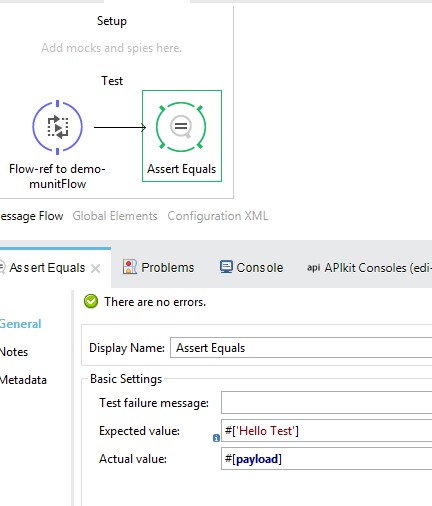
* Right-click the project and select new MUnit Test as below.



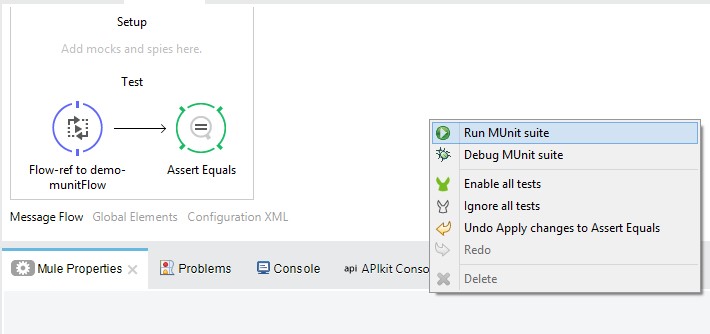
* The initial flow should look like this one:



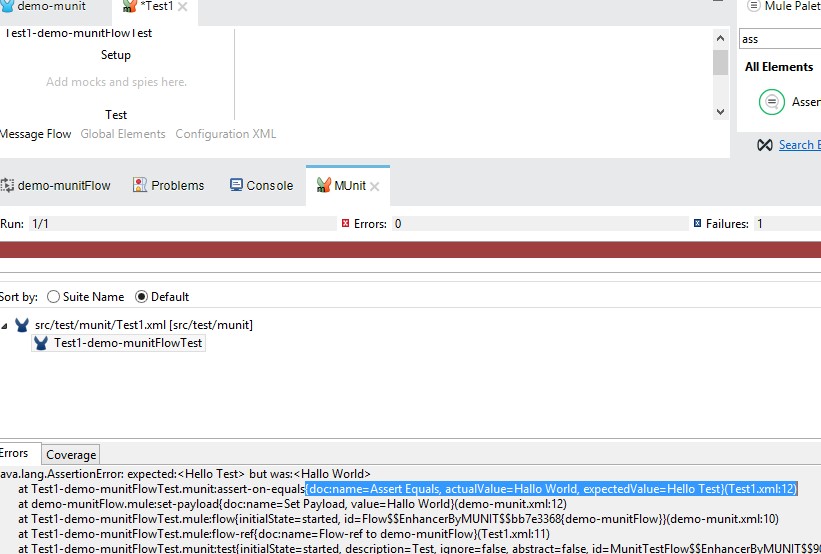
* Add an assert Equals as shown below.



* Run the test case as shown below.



* You should see a screen as below – here, the highlighted text says that it is expecting “Hallo Test” but got “Hallo world.”



* Now let’s just make a small change in the test case. Let’s go to Configuration file to and change the value of “Hello Test” to “Hallo World” and re-run the test case.

<munit:test name="Test1-demo-munitFlowTest" description="Test">

<flow-ref name="demo-munitFlow" doc:name="demo-munitFlow"/>

<munit:assert-on-equals expectedValue="#['Hallo World']" actualValue="#[payload]" doc:name="Assert Equals"/>

</munit:test>

