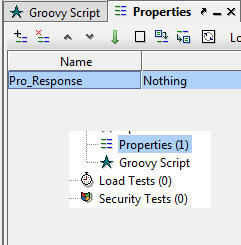
**Use Properties in SoapUI Groovy Script**

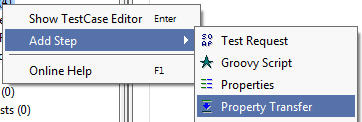
**How to Add Properties in SoapUI:**

Here are the steps.

* Right click on the Teststeps node
* Click **Add Step** and **Properties** option from the context menu
* Enter the property name as desired and click OK
* In the properties screen, click  [properties with groovy Script(1)](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/properties-with-groovy-Script1.jpg)  icon to add property
* Enter your desired property name and click OK button. For example let me enter **Pro\_Response**
* Type any default value for the property if you wish. For example, I enter “Nothing”
* Then, add a [Groovy Script](http://www.groovy-lang.org/) test step next to the property step. Refer below screen shot.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/properties-with-groovy-Script2.jpg)

We can transfer the property data across the test steps during the test execution. For that, [SoapUI](http://www.softwaretestinghelp.com/web-services-api-testing-tool-soapui-tutorial-1/) Pro provides Property Transfer test step. Look at the below screenshot.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/properties-with-groovy-Script3.jpg)

In the groovy script, add the following script. This script will assign a string text to the property and then it will show in the log after executed the test case.

|  |  |
| --- | --- |
| 1 | String testString = "TestString" |
| 2 | testRunner.testCase.setPropertyValue( "Pro\_Response", testString ) | |

|  |  |  |
| --- | --- | --- |
| 3 | def getLocalPropValue = testRunner.testCase.getPropertyValue("Pro\_Response") | |
| 4 | log.info(getLocalPropValue) |

* Once written the above script in the editor, double click on the test case name step.
* Run the test case by clicking on the  [properties with groovy Script](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/properties-with-groovy-Script.jpg)  icon and the see the results in the script log tab.

**Accessing property:**

There are several ways to access test case, test suite and project properties for setting and getting their data through the script. Here are the samples for retrieving the property data.

|  |  |
| --- | --- |
| 1 | def getTestCasePropertyValue = |
| 2 | testRunner.testCase.getPropertyValue( "LocalPropertyName") | |

|  |  |
| --- | --- |
| 3 | def getTestSuitePropertyValue = |
| 4 | testRunner.testCase.testSuite.getPropertyValue | |

|  |  |
| --- | --- |
| 5 | ( " LocalPropertyName " ) |
| 6 | def getProjectPropertyValue = | |

|  |  |  |
| --- | --- | --- |
| 7 | testRunner.testCase.testSuite.project.getPropertyValue | |
| 8 | ( " LocalPropertyName " ) |

**In order to access a global property, this is the script:**

|  |  |
| --- | --- |
| 1 | def getGlobalPropertyValue = |
| 2 | com.eviware.soapui.SoapUI.globalProperties.getPropertyValue | |

|  |  |
| --- | --- |
| 3 | ( "GlobalPropertyName" ) |

These script lines are used to set the value to the local and global property.

|  |  |
| --- | --- |
| 1 | testRunner.testCase.setPropertyValue( " LocalPropertyName ", someValue ) |
| 2 | testRunner.testCase.testSuite.setPropertyValue( " LocalPropertyName ", someValue ) | |

|  |  |  |
| --- | --- | --- |
| 3 | testRunner.testCase.testSuite.project.setPropertyValue( " LocalPropertyName ", someValue ) | |
| 4 | com.eviware.soapui.SoapUI.globalProperties.setPropertyValue |

|  |  |
| --- | --- |
| 5 | ( " GlobalPropertyName ", someValue ) |

Here in these scripts, **testRunner** is common object which might be test suites, test cases or project. **setPropertyValue** and **getPropertyValue** are the methods or functions.

As we mentioned the above script, we can assign data to the properties.

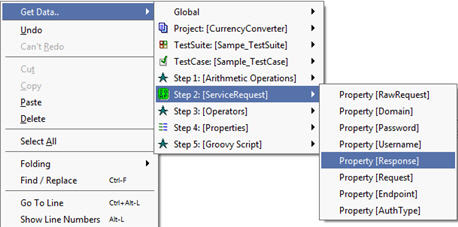
*testRunner.testCase.testSteps[“Properties”].setPropertyValue( “Pro\_Response”, testString )*

After executing the above script, the property will get updated in the property test step. Refer the following screenshot.

**Receiving response data:**

Now let us discuss how to get the response data through the script. To do this,

* Execute the service request once and verify the result
* Go to Groovy script editor and then right click on the editor as shown in the below screenshot

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/properties-with-groovy-Script5.jpg)

Now SoapUI Pro generates the script as below after specifying the property name.

def response = context.expand( ‘${ServiceRequest#Response}’ )

As we know, “**def**” is a groovy script keyword that represents defining properties / objects. By default, SoapUI Pro has the property name as “**response**” in the **Get Property** popup. If we want we can change this name. Remaining portions of the script are auto generated.

Let us merge the above script in our earlier discussed script. Here’s what you would see:

|  |  |
| --- | --- |
| 1 | def response = context.expand( '${ServiceRequest#Response}' ) |
| 2 | testRunner.testCase.setPropertyValue( "Pro\_Response", response ) | |

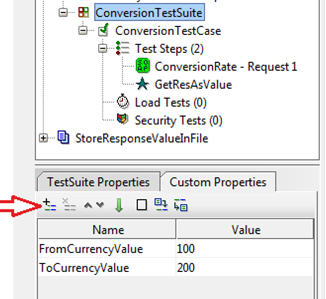
|  |  |  |
| --- | --- | --- |
| 3 | def getLocalPropValue = testRunner.testCase.getPropertyValue("Pro\_Response") | |
| 4 | log.info(getLocalPropValue) |

If we execute the above script separately, it will log the entire response data in the **log** section. Even when execute this along with the test case, it will show the same output in the **script log**.

**Creating properties from the navigator pane:**

There is another way to create properties locally through the property panel which will be appear when we click on the nodes under the project tree. Let’s see how:

* Add currency converter service request and a groovy script test step under the test suite ConversionTestSuite.
* Click on the **TestSuite** name under the project (i.e. **ConversionTestSuite**)
* At the bottom of the Navigation panel, we can see a Property panel. It contains TestSuite **Properties** and **Custom Properties** tabs.
* Go to **Custom Properties** tab by clicking on it
* Then click on the plus ( + ) icon to add property as shown below:

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/properties-with-groovy-Script6.jpg)

* Enter property name and provide default input value as shown in the above screen shot.
* Now execute the currency converter service request once. Only then we can get the property information when right click on the editor.
* Enter the following script in the editor

 defgetPropValue = context.testCase.testSuite.getPropertyValue(“FromCurrencyValue”)

* Click on the **Run** icon

This script gets the property value and assign to the variable “getProValue”. To print the value of the property, we can write the following script :

Log.info (getPropValue);

**Global Properties:**

Now let us discuss about global properties. These properties are defined in one place and we can access them across the project components like test suite, test case, test steps etc.

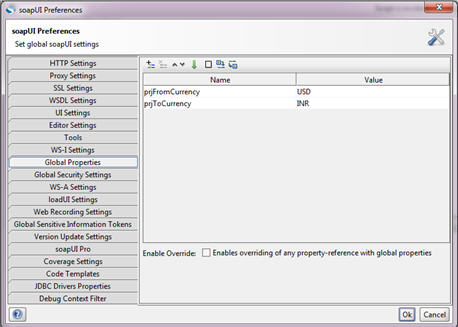
Here are the scripts for writing data to the global properties.

|  |  |  |
| --- | --- | --- |
| 1 | com.eviware.soapui.SoapUI.globalProperties.setPropertyValue | |
| 2 | ( "prjFromCurrency", "USD" ) |

|  |  |  |
| --- | --- | --- |
| 3 | com.eviware.soapui.SoapUI.globalProperties.setPropertyValue | |
| 4 | ( "prjToCurrency", "INR" ) |

Once we execute the above test step script, the mentioned properties will be created and the respective values will be assigned to those properties. Let us see how we can verify it.

* Click on the **File** menu
* Then, choose **Preferences** option
* In the left side, click on the **Global Properties** tab.
* Verify the properties in the property sheet on the right side. Refer the screenshot below:

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/properties-with-groovy-Script7.jpg)

### ****Conclusion:****

Properties are helpful for transferring the data between the test steps such as test suites, test steps and test cases. Property can be defined through groovy script. We can also assign and retrieve data of the properties through the script. And, just like other test steps we can rename or delete or disable the property test step by right click and then choosing the respective options from the context menu.