## 

**TABLE OF CONTENT**

[**Get Current User Directory**](#_uk2b15eit8sv) **1**

[**Copy Response value and set in path level**](#_adl6w7qt5pco) **2**

[**Random Number Generation**](#_xxqjt3no734q) **2**

[**Reading Property File**](#_odaas2oo35q1) **3**

[**SetUp End Point IN SoupUi**](#_qk8qhee8j2kl) **3**

[**Generate Random Number**](#_x87kdqlq021w) **4**

[**Get and Set properties**](#_u3wmy3c62b27) **4**

[**Logging results**](#_2qesetjd8xvf) **5**

[8.1. Saving all TestStep's results into files](#_9fnm7kwasv4h) 5

[8.2. Logging the result messages of all TestSteps in all failing TestCases](#_hpze11d5lset) 5

[**Add custom HTTP header to requests**](#_94g39de5ctkk) **5**

[**Run a TestCase located in another project**](#_amilqkvfkv3j) **6**

[**Script Assertion**](#_h2aq4wvixyby) **6**

[**Date and Time with in the request :**](#_mec0dd54wbk) **7**

## 

## 

## 

## 

## Get Current User Directory

def groovyUtils = new com.eviware.soapui.support.GroovyUtils(context)

def projectPath = groovyUtils.projectPath //gets the path of the project root

def response = new File(projectPath, "/etest.xml").text;

log.info response;

## Copy Response value and set in path level

import groovy.json.JsonSlurper

def project\_level = testRunner.testCase.testSuite.project

def responseContent = testRunner.testCase.getTestStepByName("Get Access Token").getPropertyValue("response")

def resp = new JsonSlurper().parseText(responseContent)

//Get Asset Id & Asset Iteam Details

def access\_token = resp.access\_token

project\_level.setPropertyValue("access\_token",access\_token)

log.info "access\_token:" +access\_token

## Random Number Generation

import com.eviware.soapui.impl.wsdl.testcase.WsdlTestRunContext

def project\_level = testRunner.testCase.testSuite.project

Random ran =new Random()

def ran\_number=ran.nextInt(10099802)

def random\_number = ran\_number.toString()

project\_level.setPropertyValue("random\_number",random\_number)

log.info random\_number

## Reading Property File

def groovyUtils = new com.eviware.soapui.support.GroovyUtils(context)

def projectPath = groovyUtils.projectPath //gets the path of the project root

Properties props = new Properties()

File propsFile = new File(projectPath, "/env.properties");

props.load(propsFile.newDataInputStream())

log.info props.getProperty('env')

Integer rand = new Random().next(100)

props.setProperty('porcupine', rand.toString())

props.store(propsFile.newWriter(), null)

props.load(propsFile.newDataInputStream())

log.info props.getProperty('porcupine')

## SetUp End Point IN SoupUi

//def result = com.eviware.soapui.support.UISupport.prompt("Please select the enviornment", "Environment", ['SIT', 'UAT'])

def testSuite\_level\_cirrus = testRunner.testCase.testSuite.project.testSuites["GetEmployeeDetails"]

def testcases = testSuite\_level\_cirrus.getTestCaseList()

testcases.each { testcase ->

def teststeps = testcase.getTestStepList()

teststeps.each { teststep ->

teststep.setPropertyValue('endpoint','https://ictoncloud.com')

}

}

## Generate Random Number

uuid = context.expand('${=java.util.UUID.randomUUID()}')

testRunner.testCase.setPropertyValue("correlationID", uuid)

## Get and Set properties

Getting properties from groovy is straight-forward; all objects that contain properties have a getPropertyValue / setPropertyValue methods. The following examples are from inside a **Groovy Script step**:

// get properties from testCase, testSuite and project  
def testCaseProperty = testRunner.testCase.getPropertyValue( "MyProp" )  
def testSuiteProperty = testRunner.testCase.testSuite.getPropertyValue( "MyProp" )  
def projectProperty = testRunner.testCase.testSuite.project.getPropertyValue( "MyProp" )  
def globalProperty = com.eviware.soapui.SoapUI.globalProperties.getPropertyValue( "MyProp" )  
  
// setting values is equally straigh forward  
testRunner.testCase.setPropertyValue( "MyProp", someValue )  
testRunner.testCase.testSuite.setPropertyValue( "MyProp", someValue )  
testRunner.testCase.testSuite.project.setPropertyValue( "MyProp", someValue )   
com.eviware.soapui.SoapUI.globalProperties.setPropertyValue( "MyProp", someValue )

From inside a **Script Assertion**, you have to access testCase in another way:

def testCaseProperty = messageExchange.modelItem.testStep.testCase.getPropertyValue( "MyProp" )

## Logging results

### **8.1. Saving all TestStep's results into files**

To save the result (request data, response data, response time, etc.) of every executed TestStep in a Project, create a Project EventHandler (Project Window > Events tab > Add new EventHandler) of type *TestRunListener.afterStep* with the following content:

filePath = 'c:/users/henrik/soapUI-results/'  
fos = new FileOutputStream( filePath + testStepResult.testStep.label + '.txt', true )  
pw = new PrintWriter( fos )  
testStepResult.writeTo( pw )  
pw.close()  
fos.close()

### **8.2. Logging the result messages of all TestSteps in all failing TestCases**

This is intended to run in a TestSuite Teardown Script.

for ( testCaseResult in runner.results )  
{  
 testCaseName = testCaseResult.getTestCase().name  
 log.info testCaseName  
 if ( testCaseResult.getStatus().toString() == 'FAILED' )  
 {  
 log.info "$testCaseName has failed"  
 for ( testStepResult in testCaseResult.getResults() )  
 {  
 testStepResult.messages.each() { msg -> log.info msg }  
 }  
 }  
}

## Add custom HTTP header to requests

Add “RequestFilter.filterRequest” event with the following example code:

def headers = request.requestHeaders  
headers.put( "X-tokenHeader", "value" )  
request.requestHeaders = headers

## Run a TestCase located in another project

def prj = testRunner.testCase.testSuite.project.workspace.getProjectByName("ProjectName")  
tCase = prj.testSuites['TestSuiteName'].testCases['TestCaseName']  
tStep = tCase.getTestStepByName("TestStepName")  
def runner = tStep.run(testRunner, context)  
log.info ("runner status ....... : " + runner.hasResponse())

## **Script Assertion**

import groovy.json.JsonSlurper

def response = messageExchange.response.responseContent

def slurper = new JsonSlurper()

def json = slurper.parseText(response)

//log.info json

def random\_number1 = context.expand('${#Project#random\_number}');

def Assetscancode = json.response.result.assetItemDetailsVOs[0].scanCode

log.info random\_number1;

def test\_Status=false

if (Assetscancode.contains(random\_number1)) {

test\_Status=true

}

else {

test\_Status=false

}

assert test\_Status==true

## **Date and Time with in the request :**

${=new java.text.SimpleDateFormat("MM-dd-yyyy HH:mm:ss").format(new Date())}