**What is SoapUI**

API Testing Tool For manual and automation testing of SOAP and REST APIs

**Why to use SoapUI**

1. To create quick and efficient API tests

2. To create API functional, performance and security tests

3. To create API Testing automation framework

**Cross-platform tool**

Built entirely over Java platform

uses Swing for GUI

**Hierarchy**

Project - Test Suite - Test Case - Test Step

**Project Navigator**

Request Editor

Response Editor

URI - Uniform Resource Identifier

**Properties can be used as variables to store values that can be referred in testing**

Properties can be accessed at following levels:

Project - ${[#Project](https://www.youtube.com/results?search_query=%23Project)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

TestSuite - ${[#TestSuite](https://www.youtube.com/results?search_query=%23TestSuite)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

TestCase - ${[#TestCase](https://www.youtube.com/results?search_query=%23TestCase)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

TestStep - ${TestStepName[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

System - ${[#System](https://www.youtube.com/results?search_query=%23System)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

Env - ${[#Env](https://www.youtube.com/results?search_query=%23Env)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

Global - ${[#Global](https://www.youtube.com/results?search_query=%23Global)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

**//Get and Set Property**

**//Project:**

testRunner.testCase.testSuite.project.getPropertyValue("Name") testRunner.testCase.testSuite.project.setPropertyValue("Name","I am in Project")

**//TestSuite**

testRunner.testCase.testSuite.getPropertyValue("Name")

testRunner.testCase.testSuite.setPropertyValue("Name","I am in TestSuite")

**//TestCase**

testRunner.testCase.getPropertyValue("Name")

testRunner.testCase.setPropertyValue("Name","I am in TestCase")

**//TestStep**

testRunner.testCase.getTestStepByName("Properties").getPropertyValue("Name") testRunner.testCase.getTestStepByName("Properties").setPropertyValue("Name","I am in Test Step")

**//Global**

com.eviware.soapui.SoapUI.globalProperties.getPropertyValue( "Name") com.eviware.soapui.SoapUI.globalProperties.setPropertyValue( "Name","I am in Global Prop" )

**// Add Property**

testRunner.testCase.testSuite.project.addProperty("DOB")

**// Remove property**

testRunner.testCase.testSuite.project.removeProperty("Name");

**//TestSuite**

testRunner.testCase.testSuite.removeProperty("Name")

**// Loop through properties**

testRunner.testCase.properties.each

{

key,value - log.info (testRunner.testCase.getPropertyValue(key))

//log.info (key+" - "+value)

}

**Code Snippets**

=============================

**This is sample code for groovy scripts to see how commands work**

import java.io.\*;

log.info (" Hello World ... ");

int a = 10 ; int b = 20 ;

int c = a+b ;

log.info (" Result is : "+c) ;

// this is a single line comment

/\* \* This is \* multiline comment \*/

//log

//context

//testRunner

log.info testRunner.metaClass.methods\*.name.unique().sort()

**This is code get hello printed**

class Hello

{

def log;

def context;

def testRunner;

def Hello(log, context, testRunner)

{

this.log = log

this.context = context

this.testRunner = testRunner

}

def sayHello(String name)

{

log.info (" Hello "+name);

}

}

context.setProperty("Hello", new Hello(log,context,testRunner))

=========================================

**This is to call code from another(callHello) groovy script**

def testStep = testRunner.testCase.getTestStepByName("Hello")

testStep.run(testRunner, context)

context.Hello.sayHello("Raghav")

=========================================

**This is to call groovy script from another test case**

project = testRunner.testCase.testSuite.project ;

tcase = project.testSuites["CountryInfoServiceSoapBindingTestsuite-3"].testCases["TestCase 1"] ;

hello = tcase.getTestStepByName("Hello");

hello.run(testRunner, context)

context.Hello.sayHello("Anusha")

=========================================

project = testRunner.testCase.testSuite.project ;

tcase = project.testSuites["TestSuite"].testCases["TestCaseName"] ;

hello = tcase.getTestStepByName("Hello");

log.info testRunner.metaClass.methods\*.name.unique().sort()

context.setProperty("Hello", new Hello(log,context,testRunner))

def hello = testRunner.testCase.getTestStepByName("Hello");

hello.run(testRunner, context)

def ref = context.Hello;

ref.printHello("Raghav");

=========================================

**//Groovy - run request from same TestCase**

def status = testRunner.runTestStepByName("TestStepName")

def result = status.getStatus().toString();

log.info (" ---- "+result)

**//Groovy - run request from another TestCase or TestSuite**

project = testRunner.testCase.testSuite.project ;

tcase = project.testSuites["TestSuiteName"].testCases["TestCaseName"] ;

tstep = tcase.getTestStepByName("TestStepName");

def status = tstep.run(testRunner, context)

def result = status.getStatus().toString();

log.info (" ---- "+result)

**Useful TIPS**

**How to get list of all TCs in a test suite**

Get TestCaseName - testRunner.testCase.name

**Notes: groovy code to run test case**

=====================

def tCase = testRunner.testCase.testSuite.testCases["TestCaseName"]

runner = tCase.run(new com.eviware.soapui.support.types.StringToObjectMap(), false);

**groovy code to loop test cases in a test suite**

**==================================**

def testCases = context.testCase.testSuite.getTestCaseList()

testCases.each

{

log.info(it.name)

}

**command line run we can**

**==============**

cd C:\Program Files (x86)\SmartBear\SoapUI-5.4.0\bin\

testrunner.bat -sCountryInfoServiceSoapBindingTestSuite-3 -c"TestCase 3" C:\Users\anushag\Documents\SoapUI-Xml's\CountryInfoService-soapui-project.xml

**To run testSuite using GroovyScript**

**==================================**

def suite = context.testCase.testSuite.project.testSuites["TestSuiteName"]

suite.run(null,false)

// null can be replaced with : new com.eviware.soapui.support.types.StringToObjectMap()

log.info (" === "+suite.getName().toString()+" - Executed successfully")

**To Using Command Line**

**==================================**

cd C:\Program Files (x86)\SmartBear\SoapUI-5.4.0\bin\

testrunner.bat -sCountryInfoServiceSoapBindingTestSuite-3 -c"TestCase 3" C:\Users\anushag\Documents\SoapUI-Xml's\CountryInfoService-soapui-project.xml

**To run a Project from Groovy**

**==================================**

def projectName=testRunner.getTestCase().getTestSuite().getProject().getWorkspace().getProjectByName("REST Project 1")

projectName.run(null,true)

//projectName.run(new com.eviware.soapui.support.types.StringToObjectMap(), false)

Go to project , double click on it -> open a tab at the bottom click on **Setup** and **Tear down**

**Setup** - runs before running the element in context

**Teardown** - runs after running the element in context

groovy code to set and get setup and teardown scripts

-------------------------------------------------------------------------- testRunner.testCase.testSuite.project.getTestSuiteByName('TestSuite1').getTestCaseByName('TestCase1').setSetupScript('log.info "setup"') testRunner.testCase.testSuite.project.getTestSuiteByName('TestSuite1').getTestCaseByName('TestCase1').setTearDownScript('log.info "teardown"')

log.info (" -- "+testRunner.testCase.getSetupScript());

log.info (" -- "+testRunner.testCase.getTearDownScript());

groovy code can be given at project load script to load setup and teardown scritps from file

--------------------------------------------------------------------------------------------------------------------------

def su = new File("setup.txt").text

def td = new File("teardown.txt").text

project.getTestSuiteList().each

{

testSuite - testSuite.getTestCaseList().each

{

testCase - testCase.setSetupScript(su)

testCase.setTearDownScript(td)

}

}

add different assertions in SoapUI Contains

XPath Match

XQuery Match

Compliance JSON

Path assertions

SLA - SERVICE LEVEL AGREEMENT

Script assertion works on the last reponse received in soapui works with messageExchange object (messageExchange object stores all the details of the last request and response)

**Script Assertion samples =====================**

**//check response time**

assert messageExchange.timeTaken 4000

**//check for Endpoint**

log.info messageExchange.getEndpoint()

**//check for TimeTaken**

log.info messageExchange.getTimeTaken()

**//check for header**

log.info (messageExchange.responseHeaders["Content-Length"])

assert messageExchange.responseHeaders["Content-Length"] != null

**//check attachments**

assert messageExchange.responseAttachments.length == 0

log.info (messageExchange.responseAttachments.length)

**//validate response nodes**

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

def requsetHolder = groovyUtils.getXmlHolder( messageExchange.requestContent )

def responseHolder = groovyUtils.getXmlHolder( messageExchange.responseContent )

def refNum = responseHolder.getNodeValue("//m:CountryCurrencyResult/m:sName")

assert responseHolder.getNodeValue("//m:CountryCurrencyResult/m:sName") == "Rupees"

**//to get response**

def resp = messageExchange.responseContentAsXml.toString()

For JSON response -------------------------------

**//get json response**

import groovy.json.JsonSlurper

def responseMessage = messageExchange.response.responseContent

def json = new JsonSlurper().parseText(responseMessage)

**//assert node values**

log.info json.name

assert json.capital.toString() != null

assert json.name.toString() == "[Taiwan]"

testStepName = messageExchange.modelItem.testStep.name//to get the Test Step Name log.info testStepName

xmlHold = messageExchange.responseContentAsXml.toString() //to store the response as Xml string

1. How to get Jenkins

2. How to add SoapUI commands in Jenkins

3. How to run SoapUI tests from Jenkins

Step 1 : Open SoapUI and goto the test case

Step 2 : Right click and select Launch Test Runner

Step 3 : Run from Test Runner and copy the commands

cd C:\Program Files\SmartBear\SoapUI-5.4.0\bin\ testrunner.bat -sTestSuite1 -cAssertions "C:\Users\Raghav Pal\OneDrive\Projects\SoapUIProjects\REST-Project-1-soapui-project.xml"

Step 4 : Get Jenkins on your system

Step 5 : Run jenkins - java -jar jenkins.war --httpPort=xxxx

Step 6 : Create a new job and add details

Step 7 : Run the job and validate results

References: Jenkins download - [https://jenkins.io/download/](https://www.youtube.com/redirect?q=https%3A%2F%2Fjenkins.io%2Fdownload%2F&v=JLQ0ayUr4CE&event=video_description&redir_token=diA4sQlk3hT5rKIASgN9Zy17Xrt8MTUzODczNjc4M0AxNTM4NjUwMzgz)

cd C:\Program Files\SmartBear\SoapUI-5.4.0\bin\ testrunner.bat -sTestSuite1 -cAssertions

cmd.exe /C testrunner.bat -sTestSuite1 -cAssertions C:\Users\anushag\Documents\SoapUI-Xml's\REST-Project-1-soapui-project.xml

SoapUI 5.4.0 TestCase Runner

1. How to setup API Monitors

2. How to Monitor APIs in SoapUI

What is API Monitoring:

Checking of your APIs at regular interval for

**Availability**

**Correctness**

**Performance**

Step 1 : Open SoapUI - Right click on Project and Select - Monitor APIs

Step 2 : Connect to AlertSite (SignUp | SignIn)

Step 3 : Create Monitors Step 4 : Add APIs to monitor and view in AlertSite

References : [https://smartbear.com/lp/alertsite/fr...](https://www.youtube.com/redirect?redir_token=gXqVE0Z1UvjUhCYN6t4MAqSyath8MTUzODgxODAyNUAxNTM4NzMxNjI1&q=https%3A%2F%2Fsmartbear.com%2Flp%2Falertsite%2Ffree-api-monitoring%2F%E2%80%8B&event=video_description&v=0fHmwmRSEa4)