

Create REST project.

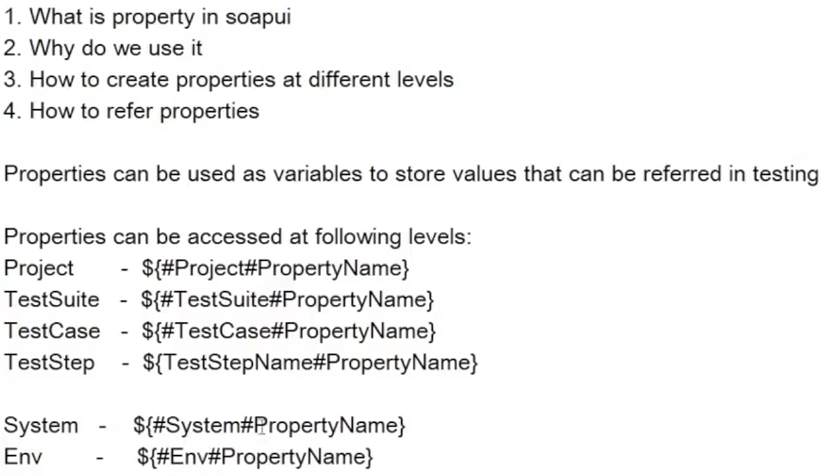
Search on sample country rest api in browser.

Click on **REST Countries**.

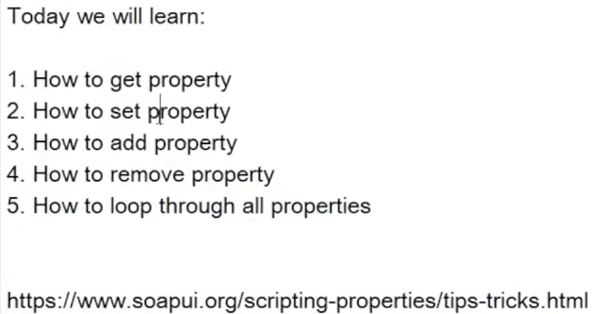
1. Scroll down page to **ALL.**
2. Copy <https://restcountries.eu/rest/v2/all>.
3. Click REST button on toolbar.



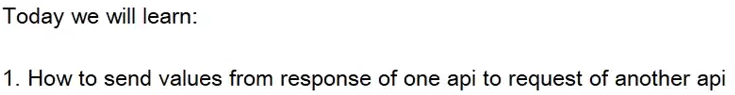












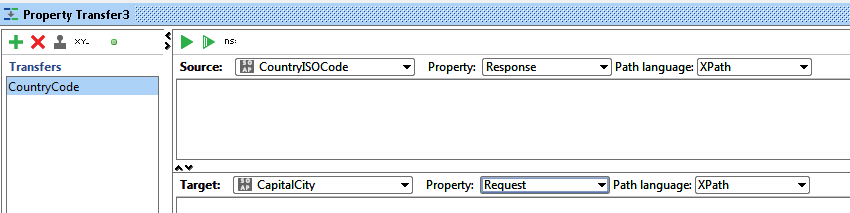
2. Create a testcase TestSuite2/“PropertyTransferDemo.

3. Clone CountryISOCode and CapitalCity Api.

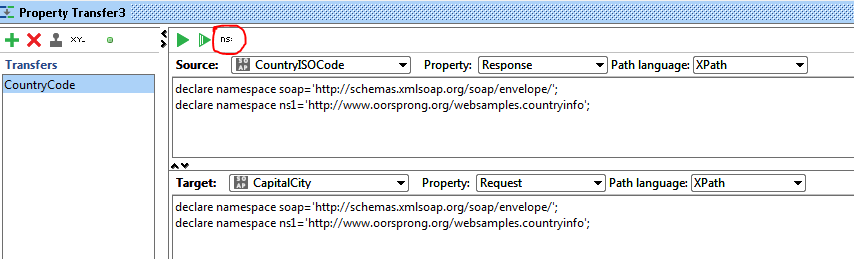
4. Right click PropertyTransferDemo/Add Step/Property Transfer and OK.

5. Click + and enter Country Code and OK.

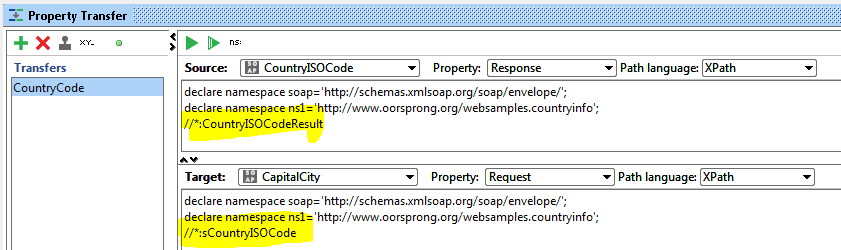
6. Enter Source and Target.



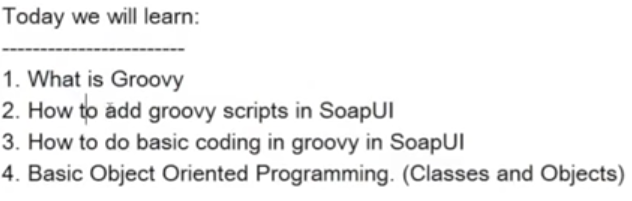
7. Click on symbol in circle.



8. Add the following lines.

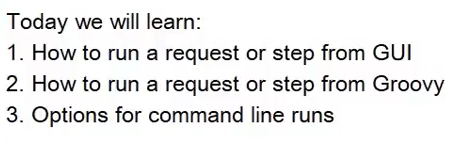






See TestSuite3.





See TestCase3



Today we will learn:

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

1. How to run a Test Case from GUI

2. How to run a Test Case from Groovy

3. How to run a Test Case from CommandLine

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 case in the test suite.

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

testCases.each{

log.info(it.name) }

command line run

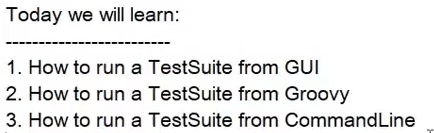
==============

cd C:\Program Files\SmartBear\SoapUI-5.5.0\bin\

testrunner.bat -sTestSuite3 -cTestCase2 "I:\SoapUI Tutorial\Beginner\CountryInfoService-soapui-project.xml"

SoapUI 5.5.0 TestCase Runner





Groovy script

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

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")

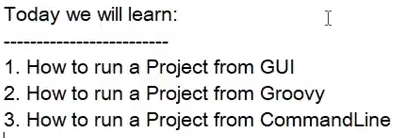
command line run

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

cd C:\Program Files\SmartBear\SoapUI-5.5.0\bin\

testrunner.bat -sTestSuite2 "I:\SoapUI Tutorial\Beginner\CountryInfoService-soapui-project.xml" SoapUI 5.5.0 TestCase Runner





Groovy Scripts

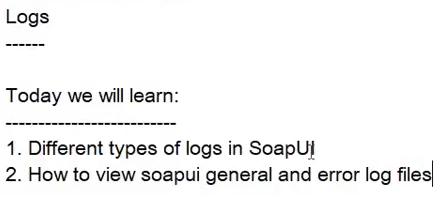
def projectName=testRunner.getTestCase().getTestSuite().getProject().getWorkspace().getProjectByName("REST Project 1") projectName.run(null,true)

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

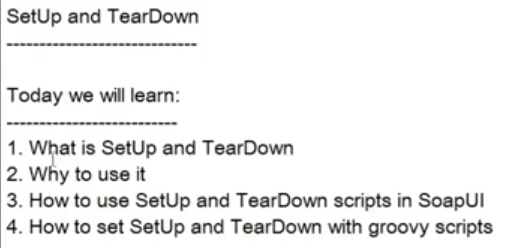
Command line run

cd C:\Program Files\SmartBear\SoapUI-5.5.0\bin\

testrunner.bat -j "I:\SoapUI Tutorial\Beginner\CountryInfoService-soapui-project.xml" SoapUI 5.5.0 TestCase Runner 









Setup - runs before running the element in context

Teardown - runs after running the element in context

Groovy code to set and et 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

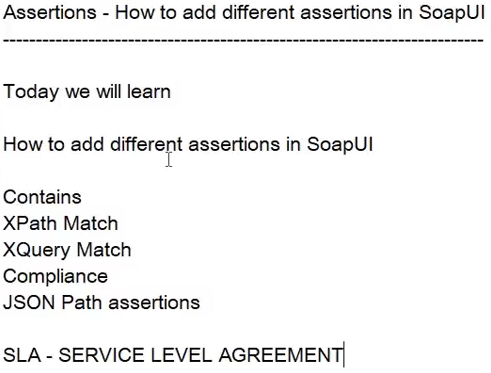
{ testCase - testCase.setSetupScript(su) testCase.setTearDownScript(td)

} }

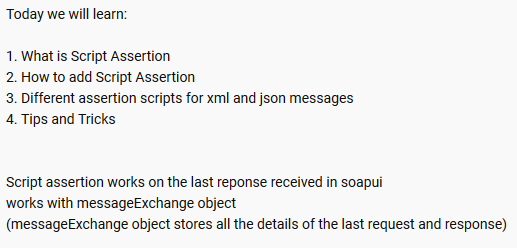














//check response time

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()

//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

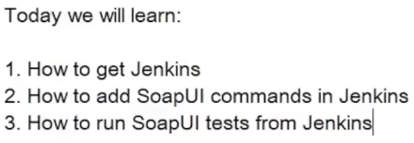
//to get the Test Step Name

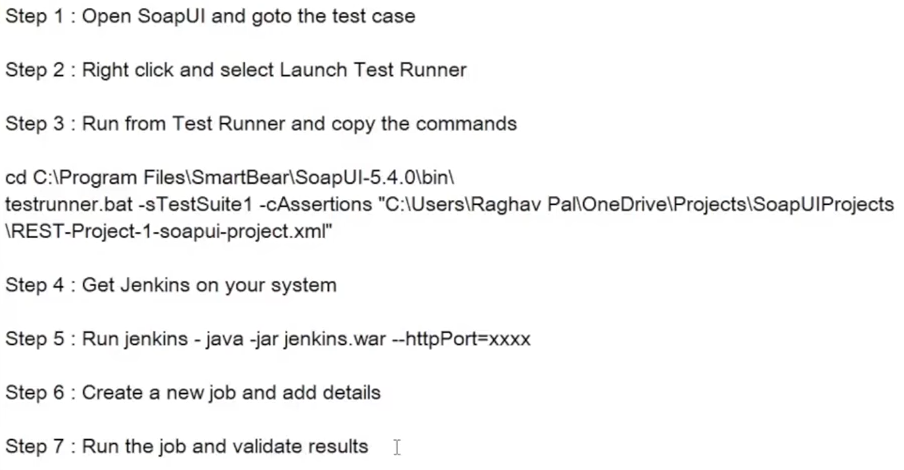
log.info testStepName

xmlHold = messageExchange.responseContentAsXml.toString()

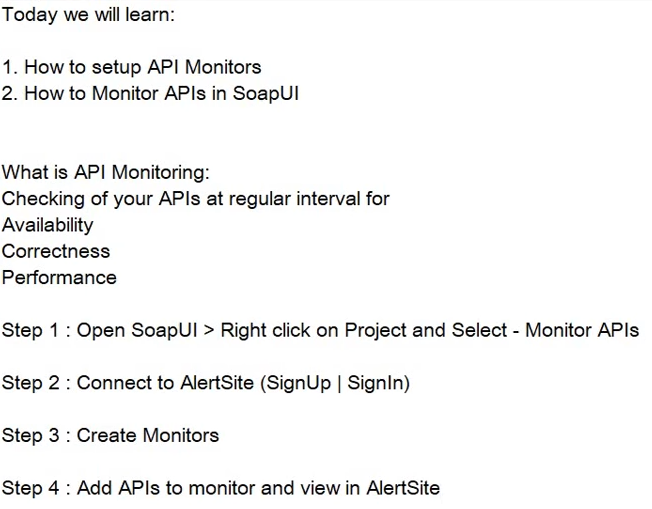
//to store the response as Xml string











Login to SMARTBEAR AlertSite to monitor APIs.

