

Public Service: <http://www.webservicex.net/periodictable.asmx?WSDL>

Public API: <http://maps.googleapis.com/maps/api/geocode/xml?address=1600+Amphitheatre+Parkway,+Mountain+View,+CA&sensor=false>

Final Driver Script:

Note: jxl jar file should be placed in “C:\Program Files (x86)\SmartBear\SoapUI-5.3.0\bin\ext” folder

import jxl.\*

import jxl.write.\*

//calling step should be declared

def step = 3

def stepname = testRunner.testCase.getTestStepAt(step).getName()

def statuscolnum = 6

/\* def props = new Properties()

def propFile = new File("C:\\Users\\616181\\Downloads\\SoapUI\\Properties\\" + stepname + ".properties")

//load the properties files into properties object

props.load(propFile.newDataInputStream())

//loop thru the properties and set them at test case level

props.each {

// context.testCase.testSuite.setPropertyValue(it.key, it.value.toString())

testRunner.testCase.getTestStepByName("Properties").setPropertyValue(it.key, it.value.toString())

sleep(500)

} \*/

Workbook wb = Workbook.getWorkbook(new File("C:\\Users\\616181\\Downloads\\SoapUI\\TestData\\" + stepname + ".xls"))

Sheet sh = wb.getSheet(0)

row = sh.getRows()

for (i=1;i<row;i++)

{

// testRunner.testCase.getTestStepByName("Properties").setPropertyValue(testRunner.testCase.getTestStepByName("Properties").getPropertyAt(0).getName() , sh.getCell(0,i).getContents())

// testRunner.testCase.getTestStepByName("Properties").setPropertyValue(testRunner.testCase.getTestStepByName("Properties").getPropertyAt(1).getName() , sh.getCell(1,i).getContents())

// testRunner.testCase.getTestStepByName("Properties").setPropertyValue(testRunner.testCase.getTestStepByName("Properties").getPropertyAt(2).getName() , sh.getCell(2,i).getContents())

// testRunner.testCase.getTestStepByName("Properties").setPropertyValue(testRunner.testCase.getTestStepByName("Properties").getPropertyAt(3).getName() , sh.getCell(3,i).getContents())

// testRunner.testCase.getTestStepByName("Properties").setPropertyValue(testRunner.testCase.getTestStepByName("Properties").getPropertyAt(4).getName() , sh.getCell(4,i).getContents())

if (sh.getCell(0,i).getContents() == "RUN")

{

def j = 1

testRunner.testCase.getTestStepByName("Properties").setPropertyValue(sh.getCell(j,0).getContents() , sh.getCell(1,i).getContents())

j++

testRunner.testCase.getTestStepByName("Properties").setPropertyValue(sh.getCell(j,0).getContents() , sh.getCell(2,i).getContents())

j++

testRunner.testCase.getTestStepByName("Properties").setPropertyValue(sh.getCell(j,0).getContents() , sh.getCell(3,i).getContents())

j++

testRunner.testCase.getTestStepByName("Properties").setPropertyValue(sh.getCell(j,0).getContents() , sh.getCell(4,i).getContents())

j++

testRunner.testCase.getTestStepByName("Properties").setPropertyValue(sh.getCell(j,0).getContents() , sh.getCell(5,i).getContents())

def status = null

def now = new Date()

def newdate=now.format("YYYY\_MM\_dd\_hh\_mm\_ss\_ms")

def newdatefol=now.format("YYYY\_MM\_dd")

// Create a File object representing the folder 'A/B'

def resfolder = new File("C:\\Users\\616181\\Downloads\\SoapUI\\ResponseXML\\" + stepname + "\_" + newdatefol)

def resfoldername = testRunner.testCase.getTestStepAt(step).getName() + "\_" + newdatefol

// If it doesn't exist

if( !resfolder.exists() ) {

// Create all folders up-to and including B

resfolder.mkdirs()

}

def inputFile = new File("C:\\Users\\616181\\Downloads\\SoapUI\\ResponseXML\\" + resfoldername + "\\" + stepname + "\_" + "Iteration\_" + i + "\_" + newdate + ".xml")

inputFile.write(testRunner.runTestStepByName(testRunner.testCase.getTestStepAt(step).getName()).getResponseContentAsXml())

sleep(500)

// Create a File object representing the folder 'A/B'

def assfolder = new File("C:\\Users\\616181\\Downloads\\SoapUI\\Assertion\\" + stepname + "\_" + newdatefol)

def assfoldername = testRunner.testCase.getTestStepAt(step).getName() + "\_" + newdatefol

// If it doesn't exist

if( !assfolder.exists() ) {

// Create all folders up-to and including B

assfolder.mkdirs()

}

def assertion = new File("C:\\Users\\616181\\Downloads\\SoapUI\\Assertion\\" + assfoldername + "\\" + stepname + "\_" + "Iteration\_" + i + "\_" + newdate + ".txt")

for (e in testRunner.testCase.getTestStepAt(step).getAssertionList())

{

assertion.append( e.getName() + " = " + e.getStatus()+ ";")

if (e.getStatus().toString() != "VALID")

{

status = "FAIL"

sleep(500)

}

}

if (status == "FAIL")

{

Workbook existingWorkbook = Workbook.getWorkbook(new File("C:\\Users\\616181\\Downloads\\SoapUI\\TestData\\" + stepname + ".xls"))

WritableWorkbook workbookCopy = Workbook.createWorkbook(new File("C:\\Users\\616181\\Downloads\\SoapUI\\TestData\\" + stepname + ".xls"), existingWorkbook)

WritableSheet sheetToEdit = workbookCopy.getSheet("TestData")

WritableCellFormat newFormat = new WritableCellFormat()

newFormat.setBackground(Colour.RED)

Label label = new Label (statuscolnum, i, status, newFormat)

//Add the created Cells to the sheet

sheetToEdit.addCell(label)

//Write and close the workbook

workbookCopy.write()

workbookCopy.close()

}

else

{

status = "PASS"

Workbook existingWorkbook = Workbook.getWorkbook(new File("C:\\Users\\616181\\Downloads\\SoapUI\\TestData\\" + stepname + ".xls"))

WritableWorkbook workbookCopy = Workbook.createWorkbook(new File("C:\\Users\\616181\\Downloads\\SoapUI\\TestData\\" + stepname + ".xls"), existingWorkbook)

WritableSheet sheetToEdit = workbookCopy.getSheet("TestData")

WritableCellFormat newFormat = new WritableCellFormat()

newFormat.setBackground(Colour.GREEN)

Label label = new Label (statuscolnum, i, status,newFormat)

//Add the created Cells to the sheet

sheetToEdit.addCell(label)

//Write and close the workbook

workbookCopy.write()

workbookCopy.close()

}

}

}

Final Driver Script with separate property step for each request:

import jxl.\*

import jxl.write.\*

//calling step should be declared

def step = 1

def PropStepNo = 2

def stepname = testRunner.testCase.getTestStepAt(step).getName()

def statuscolnum = 4

/\* def props = new Properties()

def propFile = new File("C:\\Users\\616181\\Downloads\\SoapUI\\Properties\\" + stepname + ".properties")

//load the properties files into properties object

props.load(propFile.newDataInputStream())

//loop thru the properties and set them at test case level

props.each {

// context.testCase.testSuite.setPropertyValue(it.key, it.value.toString())

testRunner.testCase.getTestStepByName("Properties").setPropertyValue(it.key, it.value.toString())

sleep(500)

} \*/

Workbook wb = Workbook.getWorkbook(new File("C:\\Users\\616181\\Downloads\\SoapUI\\TestData\\" + stepname + ".xls"))

Sheet sh = wb.getSheet(0)

row = sh.getRows()

for (i=1;i<row;i++)

{

// testRunner.testCase.getTestStepByName("Properties").setPropertyValue(testRunner.testCase.getTestStepByName("Properties").getPropertyAt(0).getName() , sh.getCell(0,i).getContents())

// testRunner.testCase.getTestStepByName("Properties").setPropertyValue(testRunner.testCase.getTestStepByName("Properties").getPropertyAt(1).getName() , sh.getCell(1,i).getContents())

// testRunner.testCase.getTestStepByName("Properties").setPropertyValue(testRunner.testCase.getTestStepByName("Properties").getPropertyAt(2).getName() , sh.getCell(2,i).getContents())

// testRunner.testCase.getTestStepByName("Properties").setPropertyValue(testRunner.testCase.getTestStepByName("Properties").getPropertyAt(3).getName() , sh.getCell(3,i).getContents())

// testRunner.testCase.getTestStepByName("Properties").setPropertyValue(testRunner.testCase.getTestStepByName("Properties").getPropertyAt(4).getName() , sh.getCell(4,i).getContents())

if (sh.getCell(0,i).getContents() == "RUN")

{

def j = 1

testRunner.testCase.getTestStepAt(PropStepNo).setPropertyValue(sh.getCell(j,0).getContents() , sh.getCell(1,i).getContents())

j++

testRunner.testCase.getTestStepAt(PropStepNo).setPropertyValue(sh.getCell(j,0).getContents() , sh.getCell(2,i).getContents())

def status = null

def now = new Date()

def newdate=now.format("YYYY\_MM\_dd\_hh\_mm\_ss\_ms")

def newdatefol=now.format("YYYY\_MM\_dd")

// Create a File object representing the folder 'A/B'

def resfolder = new File("C:\\Users\\616181\\Downloads\\SoapUI\\ResponseXML\\" + stepname + "\_" + newdatefol)

def resfoldername = testRunner.testCase.getTestStepAt(step).getName() + "\_" + newdatefol

// If it doesn't exist

if( !resfolder.exists() ) {

// Create all folders up-to and including B

resfolder.mkdirs()

}

def inputFile = new File("C:\\Users\\616181\\Downloads\\SoapUI\\ResponseXML\\" + resfoldername + "\\" + stepname + "\_" + "Iteration\_" + i + "\_" + newdate + ".xml")

inputFile.write(testRunner.runTestStepByName(testRunner.testCase.getTestStepAt(step).getName()).getResponseContentAsXml())

sleep(500)

// Create a File object representing the folder 'A/B'

def assfolder = new File("C:\\Users\\616181\\Downloads\\SoapUI\\Assertion\\" + stepname + "\_" + newdatefol)

def assfoldername = testRunner.testCase.getTestStepAt(step).getName() + "\_" + newdatefol

// If it doesn't exist

if( !assfolder.exists() ) {

// Create all folders up-to and including B

assfolder.mkdirs()

}

def assertion = new File("C:\\Users\\616181\\Downloads\\SoapUI\\Assertion\\" + assfoldername + "\\" + stepname + "\_" + "Iteration\_" + i + "\_" + newdate + ".txt")

for (e in testRunner.testCase.getTestStepAt(step).getAssertionList())

{

assertion.append( e.getName() + " = " + e.getStatus()+ ";")

if (e.getStatus().toString() != "VALID")

{

status = "FAIL"

sleep(500)

}

}

if (status == "FAIL")

{

Workbook existingWorkbook = Workbook.getWorkbook(new File("C:\\Users\\616181\\Downloads\\SoapUI\\TestData\\" + stepname + ".xls"))

WritableWorkbook workbookCopy = Workbook.createWorkbook(new File("C:\\Users\\616181\\Downloads\\SoapUI\\TestData\\" + stepname + ".xls"), existingWorkbook)

WritableSheet sheetToEdit = workbookCopy.getSheet("TestData")

WritableCellFormat newFormat = new WritableCellFormat()

newFormat.setBackground(Colour.RED)

Label label = new Label (statuscolnum, i, status, newFormat)

//Add the created Cells to the sheet

sheetToEdit.addCell(label)

//Write and close the workbook

workbookCopy.write()

workbookCopy.close()

}

else

{

status = "PASS"

Workbook existingWorkbook = Workbook.getWorkbook(new File("C:\\Users\\616181\\Downloads\\SoapUI\\TestData\\" + stepname + ".xls"))

WritableWorkbook workbookCopy = Workbook.createWorkbook(new File("C:\\Users\\616181\\Downloads\\SoapUI\\TestData\\" + stepname + ".xls"), existingWorkbook)

WritableSheet sheetToEdit = workbookCopy.getSheet("TestData")

WritableCellFormat newFormat = new WritableCellFormat()

newFormat.setBackground(Colour.GREEN)

Label label = new Label (statuscolnum, i, status,newFormat)

//Add the created Cells to the sheet

sheetToEdit.addCell(label)

//Write and close the workbook

workbookCopy.write()

workbookCopy.close()

}

}

}

Get assertion list code:

def alist = testRunner.testCase.getTestStepByName("GetAtomicNumber").getAssertionList()

for (e in alist)

{

log.info e.getName() //gives the value of the content to search for

log.info e.getClass()

log.info e.getLabel()

log.info e.getStatus()

// log.info e.getSLA()

log.info e.toString()

}

Groovy Script Add Headers:

//log.info testRunner.testCase.getTestStepByName("Properties").getPropertyValue("MaxItemClips\_1")

import com.eviware.soapui.support.types.StringToStringMap

def headers = new StringToStringMap()

headers.put("name","value");

headers.put("name1","value1")

headers.put("name2","value2")

testRunner.testCase.testSteps["GetMaxItemClips"].getHttpRequest().setRequestHeaders(headers)

Groovy Script Add Header1:

import groovy.json.JsonSlurper;

import com.eviware.soapui.support.types.StringToStringMap;

//get the response from the test step that makes the call to get the authentication token

def authResponse = testRunner.testCase.getTestStepByName("GetToken").getPropertyValue("response");

//parse the json response

def authResponseJSON = new JsonSlurper().parseText(authResponse);

//access the access\_token property in the respose

def oauthToken = authResponseJSON.access\_token;

//create a new empty map for headers

def headers = new StringToStringMap();

//create a key value pair in the headers mao

headers.put("OAuth-Token", oauthToken);

//obtain the list of test steps

def steps = context.testCase.getTestStepList();

//iterate thru each of the test steps and set the oauth token to each of the steps

steps.each{

// This block will loop through the test steps and set headers for each of them

if (it.name != 'GetToken' && it.name != 'TransferOauthToken')

it.testRequest.setRequestHeaders(headers);

}

Groovy Script make folder:

// Create a File object representing the folder 'A/B'

def folder = new File("C:\\Users\\616181\\Downloads\\SoapUI\\ResponseXML\\baskar" )

// If it doesn't exist

if( !folder.exists() ) {

// Create all folders up-to and including B

folder.mkdirs()

}

// Then, write to file.txt inside B

new File( folder, "googleapi\_Iteration\_0\_2017\_03\_20\_04\_51\_46\_5146.xml" ).withWriterAppend { w ->

w << "Some text\n"

}

XPath Expression:

declare namespace a='http://schemas.datacontract.org/2004/07/Pearson.Epen.DataContracts';

declare namespace ns1='http://epen.pearson.com/Services/2011/11';

//ns1:GetMaxItemClipsResponse[1]/ns1:GetMaxItemClipsResult[1]/a:Item[1]/a:MaxItemClips[1]

XPath Expression Count:

declare namespace ns1='http://epen.pearson.com/Services/2011/11';

declare namespace i='http://www.w3.org/2001/XMLSchema-instance';

declare namespace a='http://schemas.datacontract.org/2004/07/Pearson.Epen.DataContracts';

declare namespace s='http://schemas.xmlsoap.org/soap/envelope/';

count(//a:MaxItemClips)

XQuery example:

declare namespace ns1='http://epen.pearson.com/Services/2011/11';

declare namespace i='http://www.w3.org/2001/XMLSchema-instance';

declare namespace a='http://schemas.datacontract.org/2004/07/Pearson.Epen.DataContracts';

declare namespace s='http://schemas.xmlsoap.org/soap/envelope/';

<Result>

{

for $X in //a:InstanceUri

where $X/a:UriDescription = 'DestinationPath'

order by $X/a:UriTypeId

return <Name>{data($X/a:Uri)} {data($X/a:UriDescription)}</Name>

}</Result>

Script Assertion:

import com.eviware.soapui.support.XmlHolder

def holder = new XmlHolder( messageExchange.responseContentAsXml )

holder.namespaces["a"] = "http://schemas.datacontract.org/2004/07/Pearson.Epen.DataContracts"

holder.namespaces["ns1"] = "http://epen.pearson.com/Services/2011/11"

def cnt = holder.getNodeValue("count(//a:MaxItemClips)")

def cnt1 = Integer.parseInt(cnt)

for(i=1;i<cnt1;i++)

{

def node = holder.getNodeValue( "//ns1:GetMaxItemClipsResponse[1]/ns1:GetMaxItemClipsResult[1]/a:Item["+i+"]/a:MaxItemClips[1]/text()" )

//ns1:GetMaxItemClipsResponse[1]/ns1:GetMaxItemClipsResult[1]/a:Item[1]/a:MaxItemClips[1]

//log.info node

assert node != null

targetStep = messageExchange.modelItem.testStep.testCase.getTestStepByName('GetMaxItemClipsProperties')

def val = targetStep.getPropertyValue("MaxItemClips\_"+i)

log.info "Value from response : " + node

log.info "Value from excel : " + val

assert node == val

/\* if (node == val)

{

log.info "true"

}

else

{

log.info "false"

} \*/

}

Script Assertion response time:

assert messageExchange.timeTaken <= 775

Script Assertion get response header:

def val = messageExchange.getResponseHeaders()

val.each

{

k,v ->

log.info "name ="+k+", value = "+v

}

Script Assertion:

log.info("MessageExchange class = " + messageExchange.getClass().toString())

log.info("Endpoint = " + messageExchange.getEndpoint())

log.info("Request Content = " + messageExchange.getRequestContent())

log.info("Response Content Xml = " + messageExchange.getResponseContentAsXml())

log.info("Time Taken = " + messageExchange.getTimeTaken() + "ms")

//Using a map: get the map and assign it to variable respHeadersMap

respHeadersMap = messageExchange.getResponseHeaders()

//Retrieve the map's keys and assign them to a list, respHeadersKeys

respHeadersKeys = respHeadersMap.getKeys()

log.info("Response Headers info:")

/\*Use a for loop to step through each key in the map and

use the map's get function to get each key's corresponding

map data. The get function takes two arguments: the first

is the key for which you'd like the corresponding data, the

second is a default value if the key can not be found in the map.\*/

for(key in respHeadersKeys){

log.info(" " + key + ": " + respHeadersMap.get(key,"NONE"))

}

//To clarify, here's the Date header retrieved again:

log.info(" Date (again): " + respHeadersMap.get("Date","NONE"))

/\*And here's a call to retrieve from a key that doesn't exist;

the default value of "NONE" is used instead.\*/

log.info(" Bogus: " + respHeadersMap.get("Bogus","NONE"))

Script Assertion get raw response data:

//Get raw request/response

def myRequestStep = messageExchange.modelItem.testStep.testCase.getTestStepByName('GetPaperName')

def GetrequestData = new String(myRequestStep.testRequest.messageExchange.getRawRequestData())

def GetresponseData = new String(myRequestStep.testRequest.messageExchange.getRawResponseData())

//assert (GetresponseData.contains("20375 F01"))

//assert true

log.info GetrequestData

log.info GetresponseData

Script assertion (not working)

//Get the result for the first step, a test request step

//Assign it to variable tReqResult

tReqResult = testRunner.getResults()[0]

log.info("tReqResult Class = " + tReqResult.getClass().toString())

log.info("Endpoint = " + tReqResult.getEndpoint())

log.info("Request Content Xml = " + tReqResult.getRequestContentAsXml())

log.info("Response Content = " + tReqResult.getResponseContent())

//Assign the value returned by getTimestamp() to variable timeStampAsLong

timeStampAsLong = tReqResult.getTimestamp()

//Use long value stored in timeStampAsLong to create a new Date object, assigned

//to variable timeStampAsTime; this can be used to get a "friendly" date format.

timeStampAsTime = new Date(timeStampAsLong)

log.info("Time Stamp = " + timeStampAsTime)

TestRunner command:

set path="C:\Program Files\SmartBear\ReadyAPI-1.9.0\bin"

testrunner.bat -s"BasicHttpBinding\_IConfigService\_TestSuite" -c"BasicHttpBinding\_IConfigService\_TestCase" -r -j -fC:\Users\616181\Downloads\SoapUI\Reports\BasicHttpBinding\_IConfigService\_TestSuite\_2017-03-20-14-30-06 D:\ConfigService-soapui-project.xml

Ant build.xml

Note: Ant path setting

Environment Variables:

Variable name: ANT\_HOME

Variable value: C:\Users\616181\Downloads\SoapUI\Libraries\apache-ant-1.10.1

Variable name: JAVA\_HOME

Variable value: C:\Program Files\Java\jdk1.8.0\_102

Variable name: path

Variable value: %JAVA\_HOME%/bin;%ANT\_HOME%/bin;

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<project basedir="." default="testreport" name="PeriodicTableExampleProject">

<tstamp>

<format property="TODAY\_MY" pattern="yyyyMMdd-HHmmss" locale="en,UK" />

</tstamp>

<mkdir dir="C:/Users/616181/Downloads/SoapUI/Reports/periodictableSoap\_TestSuite\_${TODAY\_MY}"/>

<!-- Target #1. Set property value depends on check result -->

<target name="check-dir">

<available property="no.date.dir" file="C:/Users/616181/Downloads/SoapUI/Reports/periodictableSoap\_TestSuite\_${TODAY\_MY}" type="dir"/>

</target>

<!-- target #2. Create dir 'asjava' if doesn't exist -->

<target name="create-date-dir" depends="check-dir" unless="no.date.dir">

<mkdir dir="no.date.dir"/>

</target>

<target name="SoapUI">

<exec dir="." executable="C:\Program Files (x86)\SmartBear\SoapUI-5.3.0\bin\testrunner.bat">

<arg line="-r -j -f 'C:\Users\616181\Downloads\SoapUI\Reports' -speriodictableSoap\_TestSuite 'C:\Users\616181\Downloads\SoapUI\Projects\Periodic-Table-Example-Project-Report-soapui-project.xml'"></arg>

</exec>

</target>

<target name="testreport" depends="SoapUI">

<junitreport todir="C:/Users/616181/Downloads/SoapUI/Reports/periodictableSoap\_TestSuite\_${TODAY\_MY}">

<fileset dir="C:\Users\616181\Downloads\SoapUI\Reports">

<include name="TEST-\*.xml"/>

</fileset>

<report todir="C:/Users/616181/Downloads/SoapUI/Reports/periodictableSoap\_TestSuite\_${TODAY\_MY}">

</report>

</junitreport>

</target>

</project>

Jenkins:

Launch jenkin : localhost:8080

Build.xml path : C:\Program Files (x86)\Jenkins\workspace\PeriodictableSoap\_TestSuite\build.xml

Job: invoke ant.

Git:

Step 1: initialize local repository (git)

616181@PC338374 MINGW64 ~/Downloads/SoapUI/Projects

$ git init

Initialized empty Git repository in C:/Users/616181/Downloads/SoapUI/Projects/.g it/

Step 2: link with remote repository (github)

616181@PC338374 MINGW64 ~/Downloads/SoapUI/Projects (master)

$ git remote add origin https://github.com/616181/EPEN.git

616181@PC338374 MINGW64 ~/Downloads/SoapUI/Projects (master)

$ git status

On branch master

Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

Periodic-Table-Example-Project-Report-soapui-project.xml

Periodic-Table-Example-Project-soapui-project.xml

nothing added to commit but untracked files present (use "git add" to track)

Step 3: add the files.

616181@PC338374 MINGW64 ~/Downloads/SoapUI/Projects (master)

$ git add Periodic-Table-Example-Project-soapui-project.xml

warning: LF will be replaced by CRLF in Periodic-Table-Example-Project-soapui-project.xml.

The file will have its original line endings in your working directory.

616181@PC338374 MINGW64 ~/Downloads/SoapUI/Projects (master)

$ git status

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: Periodic-Table-Example-Project-soapui-project.xml

Untracked files:

(use "git add <file>..." to include in what will be committed)

Periodic-Table-Example-Project-Report-soapui-project.xml

616181@PC338374 MINGW64 ~/Downloads/SoapUI/Projects (master)

$ git config --global user.name 616181

616181@PC338374 MINGW64 ~/Downloads/SoapUI/Projects (master)

$ git config --global user.email baskarsek@gmail.com

Step 4: commit the file changes

616181@PC338374 MINGW64 ~/Downloads/SoapUI/Projects (master)

$ git commit -m "first commit"

[master (root-commit) 5e2c874] first commit

1 file changed, 1321 insertions(+)

create mode 100644 Periodic-Table-Example-Project-soapui-project.xml

616181@PC338374 MINGW64 ~/Downloads/SoapUI/Projects (master)

$ git remote -v

origin https://github.com/616181/EPEN.git (fetch)

origin https://github.com/616181/EPEN.git (push)

Step 5: push the file changes

616181@PC338374 MINGW64 ~/Downloads/SoapUI/Projects (master)

$ git push origin master

Counting objects: 3, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 8.23 KiB | 0 bytes/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/616181/EPEN.git

\* [new branch] master -> master

Force push example:

616181@PC338374 MINGW64 ~/Downloads/SoapUI/Projects (master)

$ git push origin master --force

Counting objects: 8, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (8/8), done.

Writing objects: 100% (8/8), 10.60 KiB | 0 bytes/s, done.

Total 8 (delta 2), reused 0 (delta 0)

remote: Resolving deltas: 100% (2/2), done.

To https://github.com/616181/EPEN.git

5bd34d9..d0b7d5a master -> master

Other user steps Download into local:

Step1: download

C:\Users\616181\Downloads\SoapUI\BatFiles>git clone https://github.com/616181/EPEN.git

Cloning into 'EPEN'...

remote: Counting objects: 3, done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

Step 2: modify the files.

Step 3: git status

Step 4: $ git add Periodic-Table-Example-Project-soapui-project.xml

Step 5: $ git commit -m "first commit"

Step 6: $ git push origin master –force

Reference:

Git video:

1. WebService Testing using SoapUI: Tutorial-28-Part 1 : Code Management using GIT
2. WebService Testing using SoapUI: Tutorial-28-Part 2 : Code Management using GIT
3. WebService Testing using SoapUI: Tutorial-28-Part 3 : Code Management using GIT
4. WebService Testing using SoapUI: Tutorial-28-Part 4 : Code Management using GIT

class - method call video:

1. WebService Testing using SoapUI: Tutorial-27-Part-1 : Reusable Script Library
2. WebService Testing using SoapUI: Tutorial-27-Part-2 : Reusable Script Library
3. WebService Testing using SoapUI: Tutorial-27-Part-3 : Reusable Script Library

Ant Video:

1. WebService Testing using SoapUI: Tutorial-26 : SoapUI with ANT | HTML report in Soapui free

Script Assertion:

1. <http://testautomationnoob.blogspot.co.uk/2013/10/beginning-soapui-scripting-7-more-with.html>

Assertion Video:

1. SOAP UI Assertions Part 1 - [www.soapui-tutorial.com](http://www.soapui-tutorial.com)

Data Driven:

<http://testautomationnoob.blogspot.co.uk/2014/02/data-driven-testing-with-xls-example-2.html>

<https://www.youtube.com/watch?v=xdw4F2xo1io>

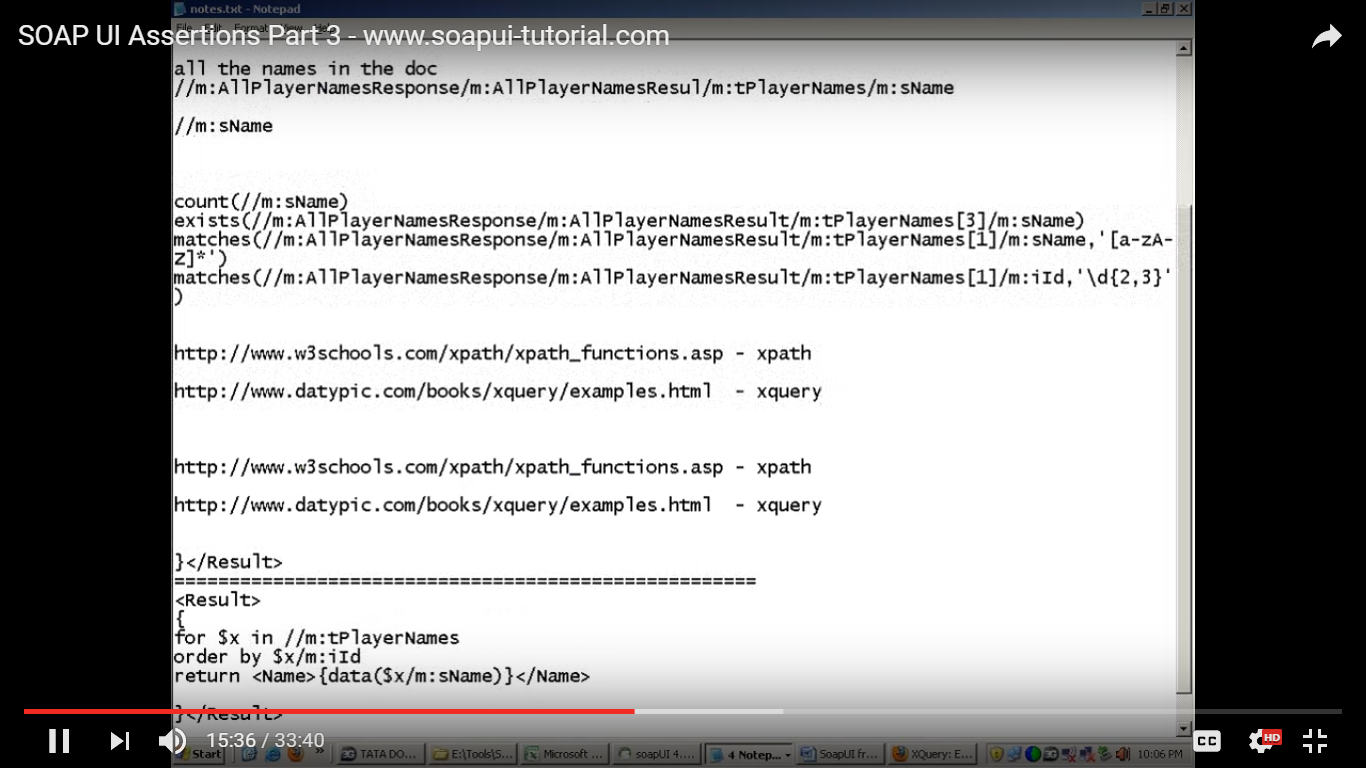
Public Web Services:

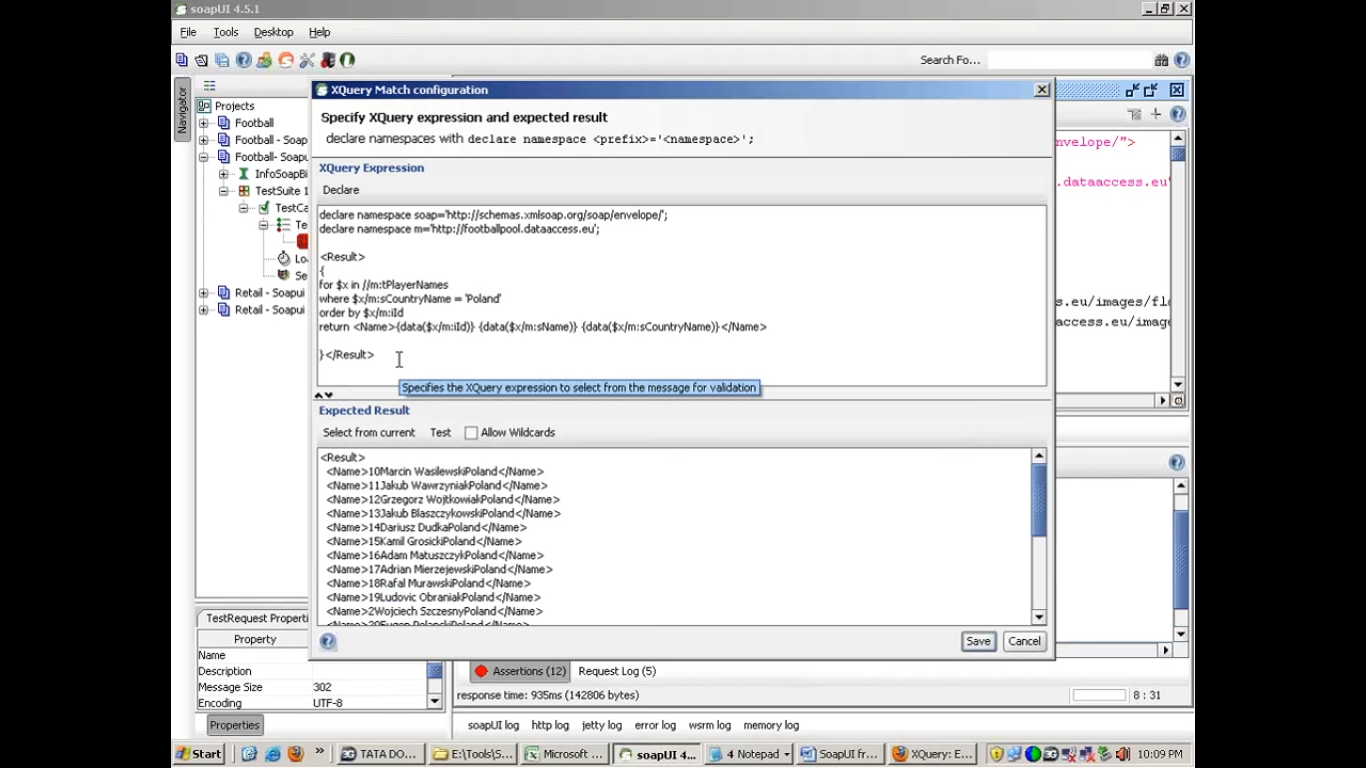
Webservicex.net

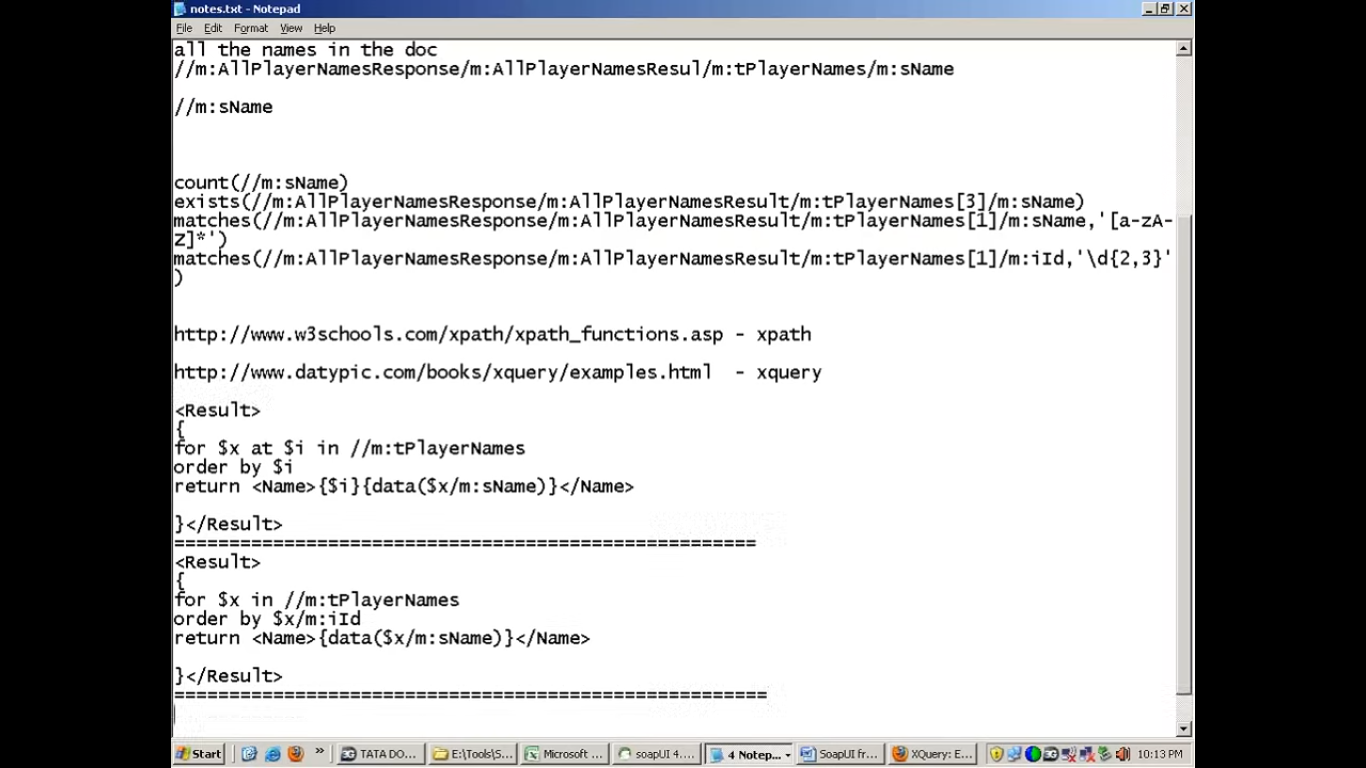
Rest API:

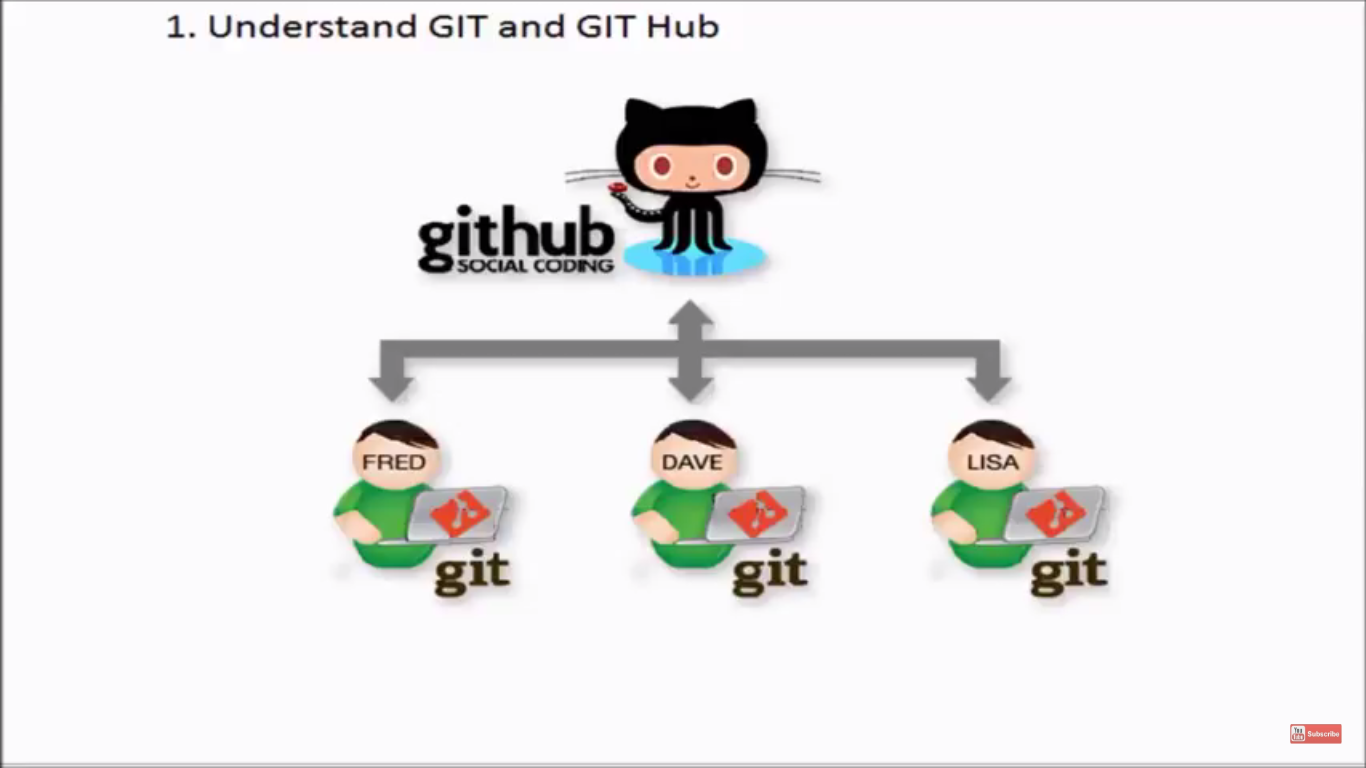
<https://www.youtube.com/watch?v=XV7WW0bDy9c>











https://git-scm.com/downloads

