SOAP UI DOCUMENT

Web services:

Web service is a standardized medium to propagate communication between the client and server applications on the World Wide Web. A web service is a software module which is designed to perform a certain set of tasks.

There are mainly two types of web services.

1. SOAP web services.
2. RESTful web services.

What is SOAP UI?

Soap UI is a tool for testing Web Services; these can be the SOAP Web Services as well RESTful Web Services or HTTP based services. Soap UI is an Open Source and completely free tool with a commercial companion Soap UI Pro- that has extra functionality for companies with mission critical Web Services.

There are three main types of projects in SOAP UI:

- SOAP projects
- REST projects
- Generic projects

The project type indicates the main type of the tested service. However, you can manually combine service types in any projects you create.

2.1. SOAP Projects:

SOAP projects can be created from a WSDL file or single service call. You can use these projects to test every aspect of your SOAP services, verify that the services support commonly used standards such as WS-Security, WS-Addressing, and MTOM, create functional and load tests, and much more.

2.2. REST Projects:

REST projects can be created from WADL file or, directly, URI and its parameters. You can use these projects to test RESTful services, create various requests and check the information you receive, give a try to a multitude of methods and operation..

2.3. Generic Projects:

Generic projects are all-purpose project for services with varied interfaces and methods. They may combine tests you create for REST and SOAP services, as well as data-driven tests, mockups, and all the functionality SOAP UI provides.

What is groovy Scripting?

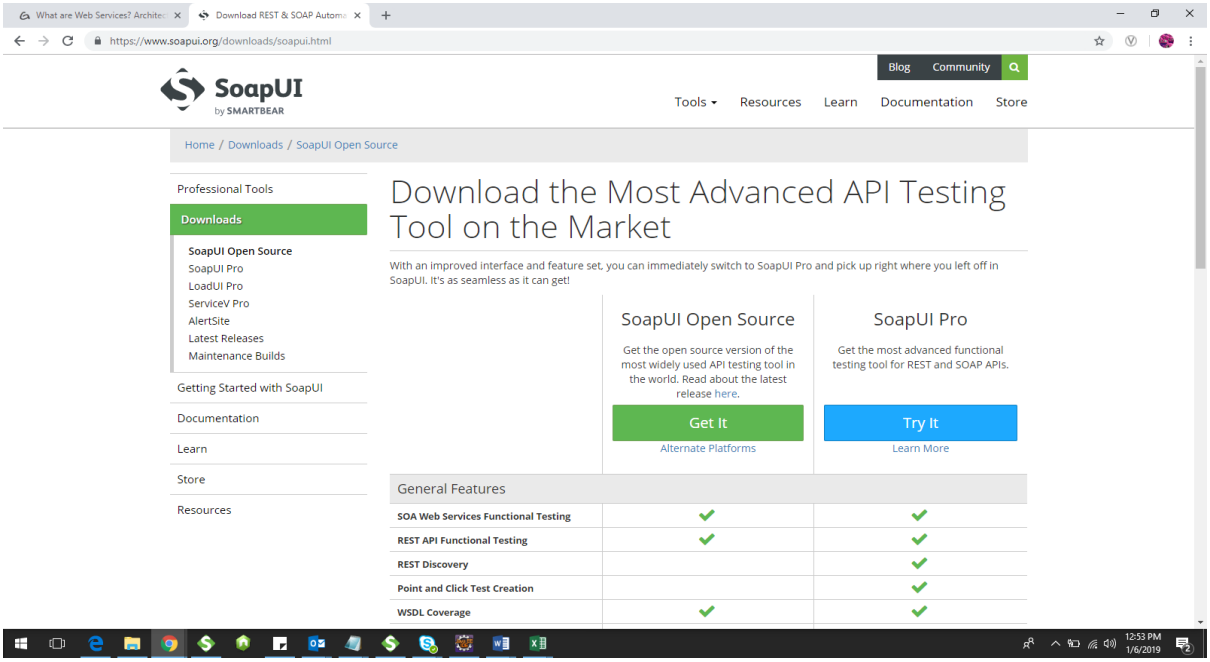
Apache Groovy is an Object-oriented programming language used for Java platform. This dynamic language has many features which are similar to Python, Ruby, Smalltalk, and Pero. It can be used as a scripting language for the Java platform.

Groovy source code gets compiled in Java Bytecode so it can run on any platform if JRE is installed on that operating system. Groovy also performs a lot of tasks behind the scene that makes it more agile and dynamic.

SOAP UI provides extensive options for scripting, using either Groovy or JavaScript (since SOAP UI 3.0) as its scripting language.

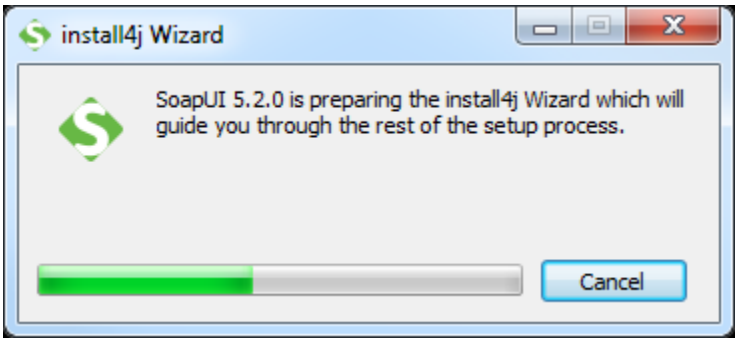
Steps for Installation of SOAP UI:

1. The link for downloading the SOAP UI open source or the pro version is attached below.
   Link:https://www.soapui.org/downloads/soapui.html

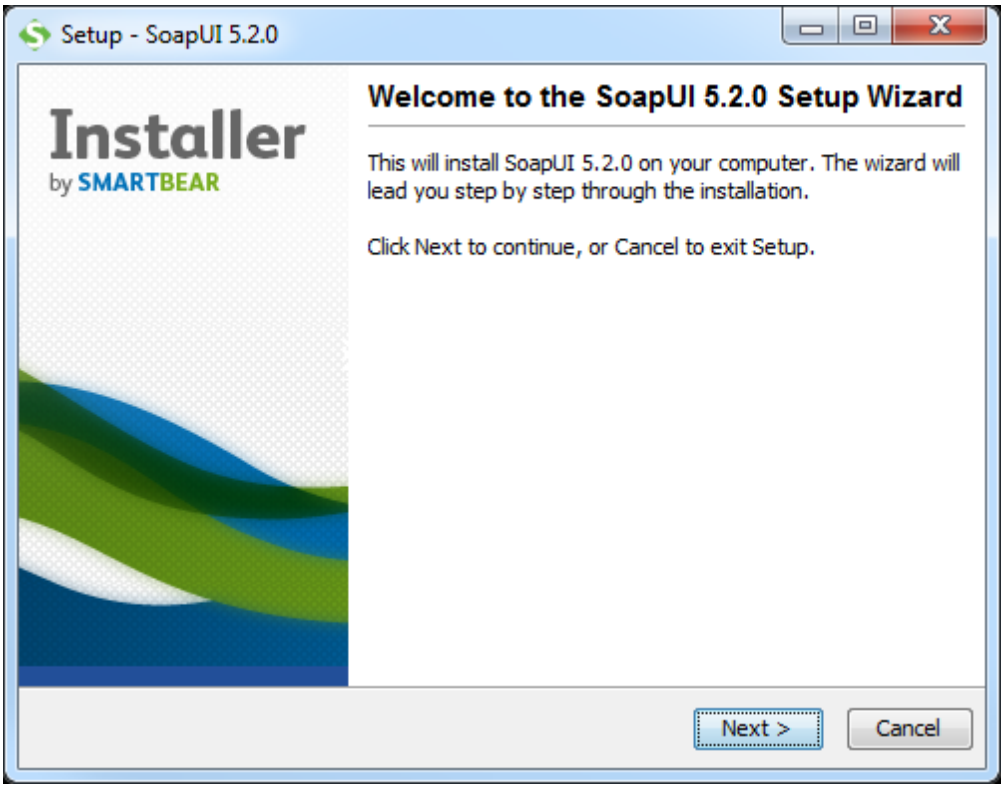


2. Installation of SOAP UI :

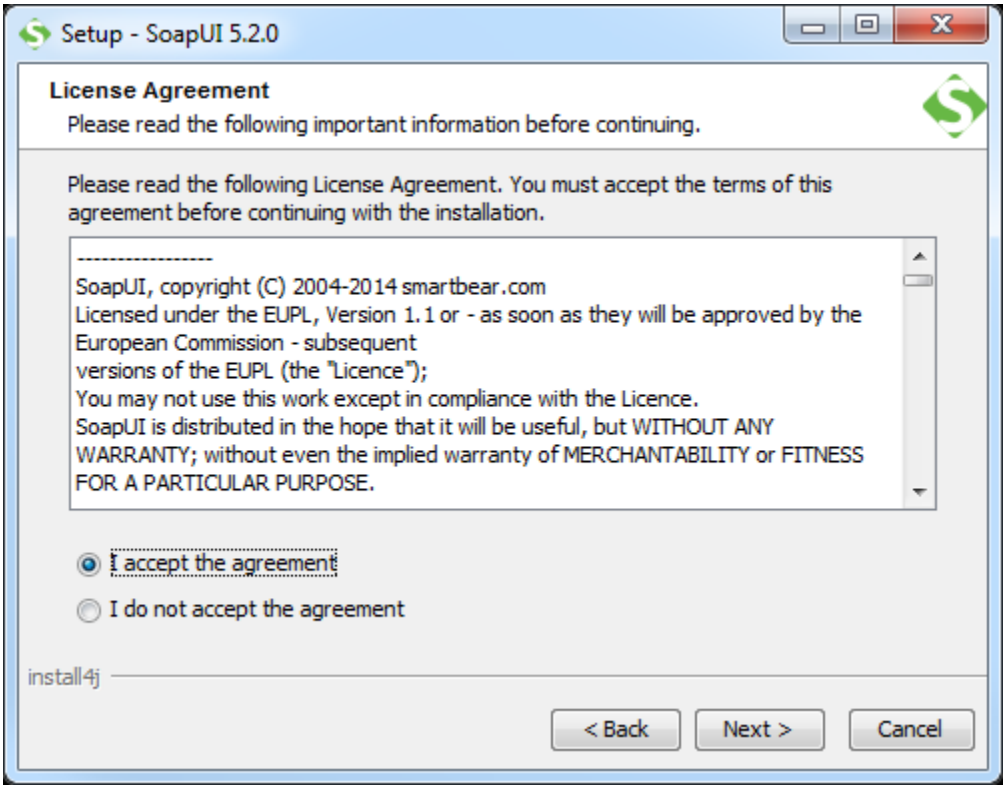
Once you've downloaded the installer, Start it by just double-clicking on it. The installer will start immediately.



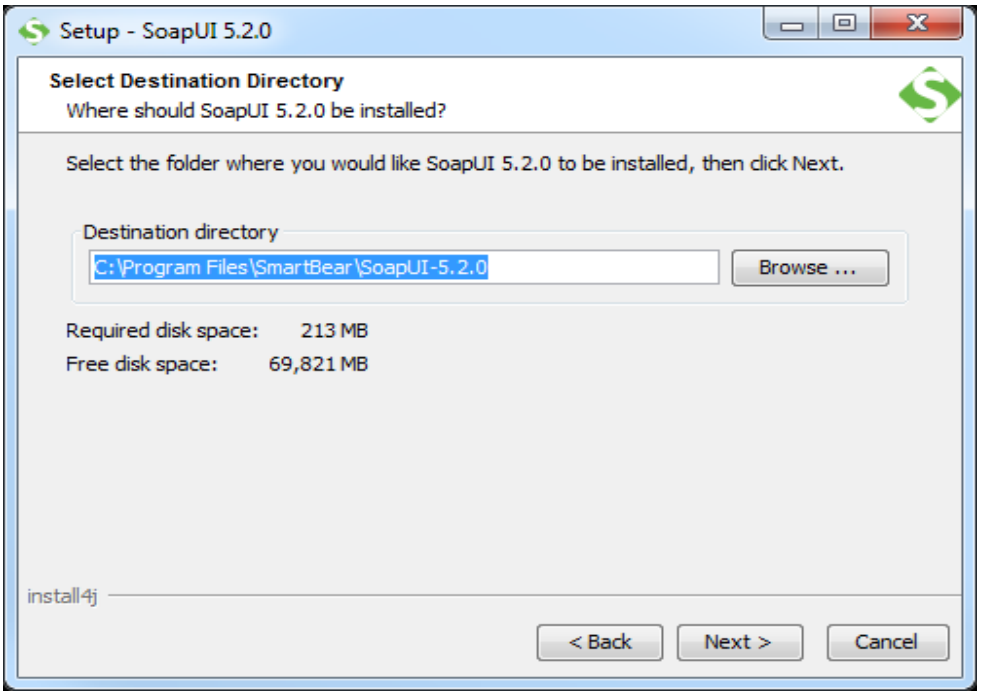
You'll soon see the starting screen.



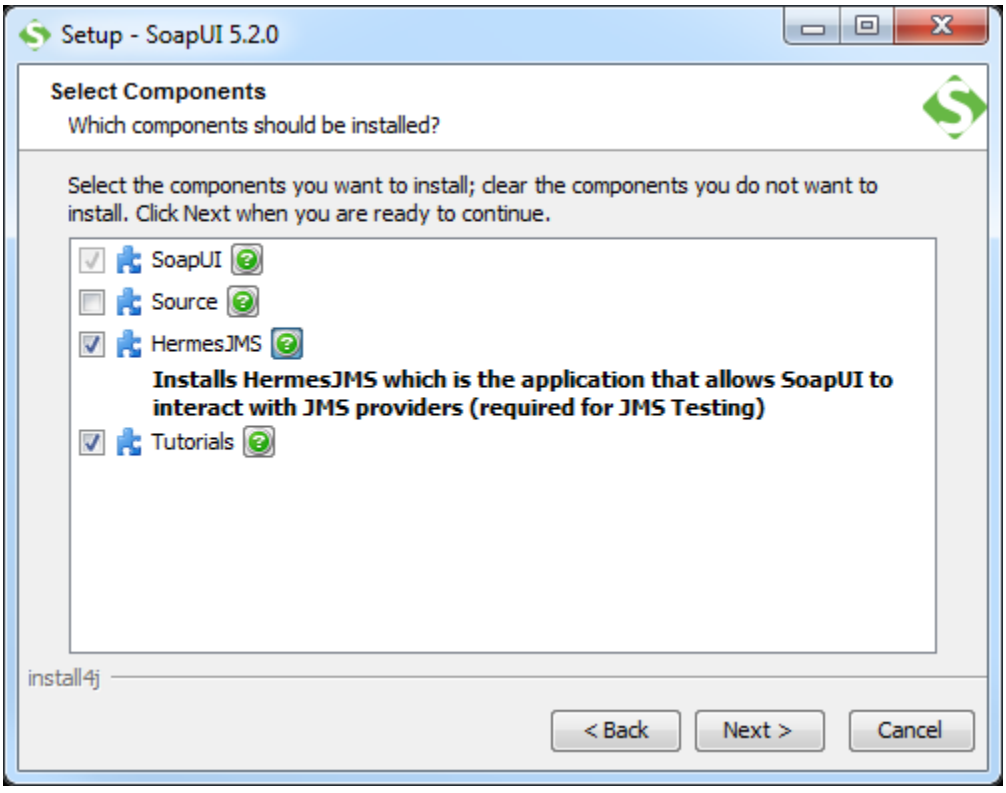
3. Click Next to continue. You'll be asked to accept our license agreement.



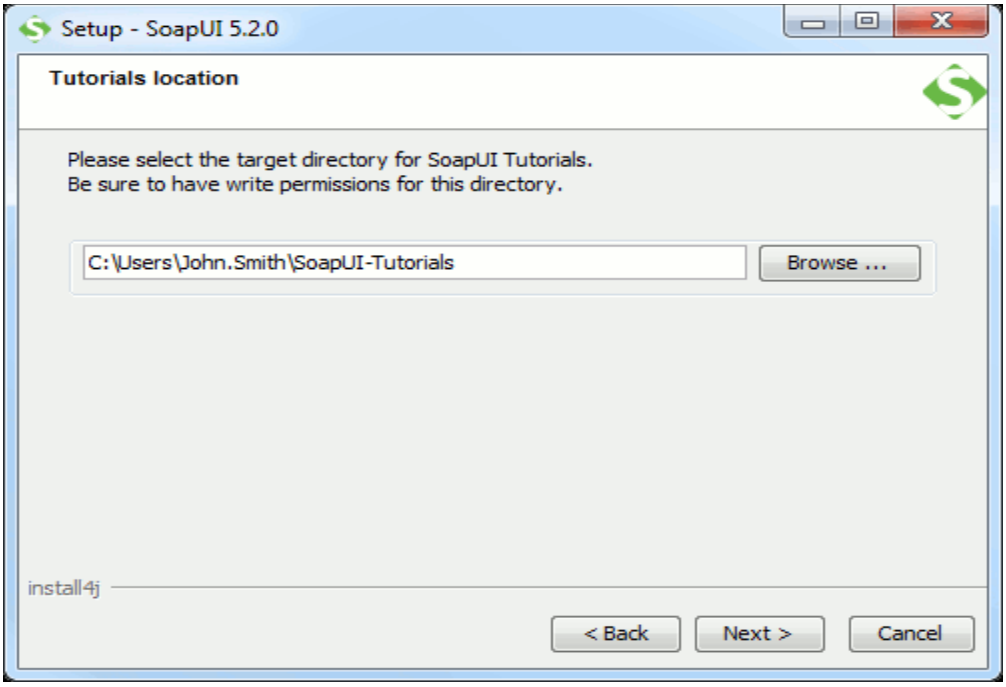
4. Select the destination folder, which by default is set to C:\Program Files\SmartBear\SoapUI-5.2.0, (path can be changed).



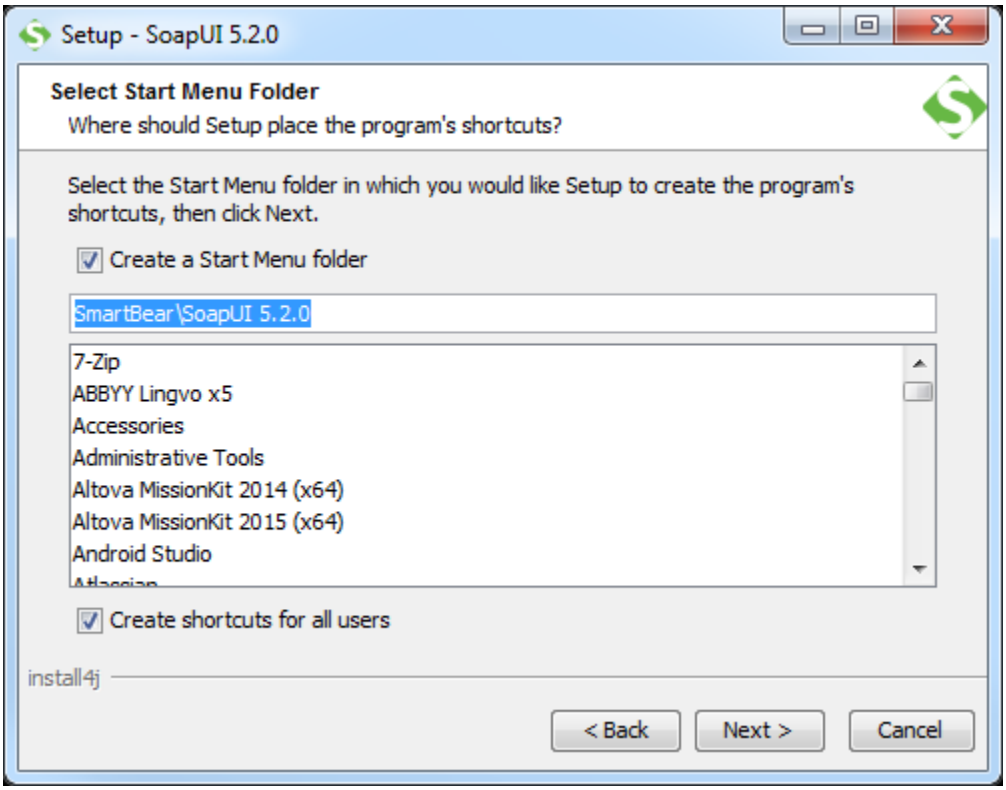
5. Include all the additional components like tutorials required as per your requirements.



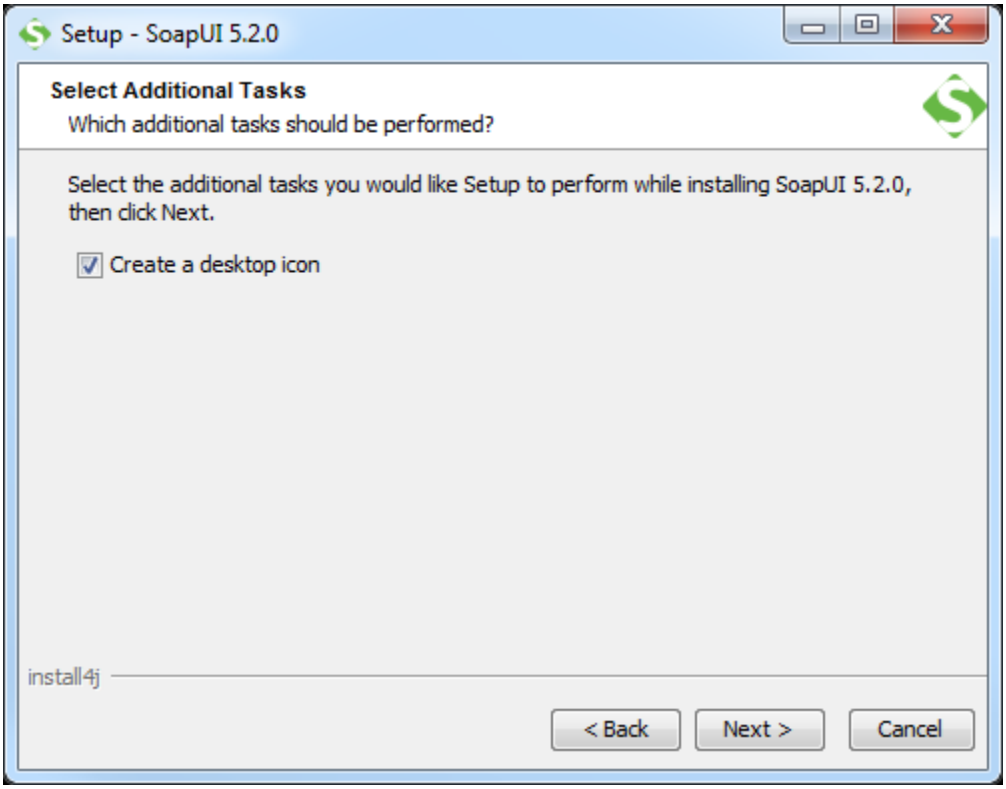
6. Specify the path to the directory where the tutorials will be installed.



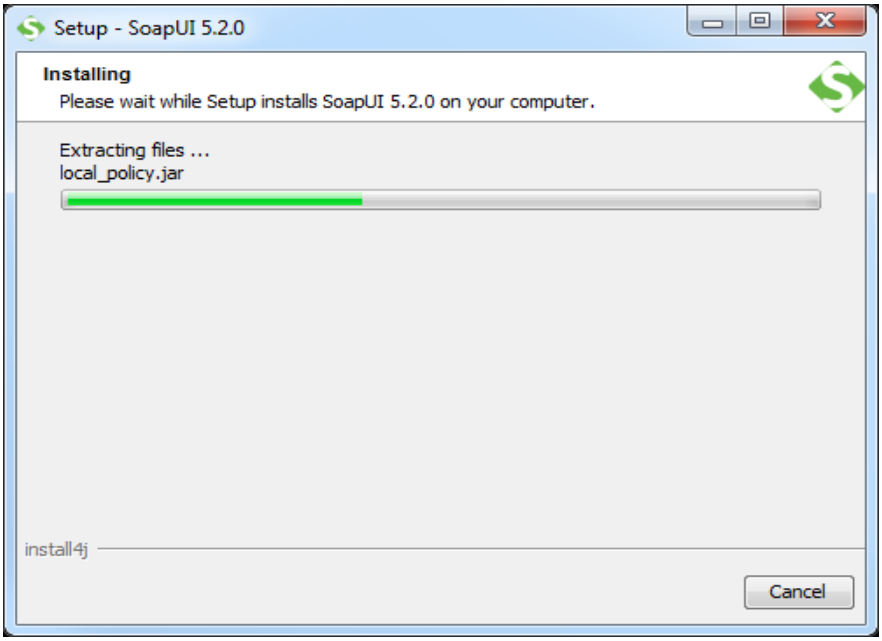
7. Select the Start Menu folder where you want the SOAP UI shortcut to be added



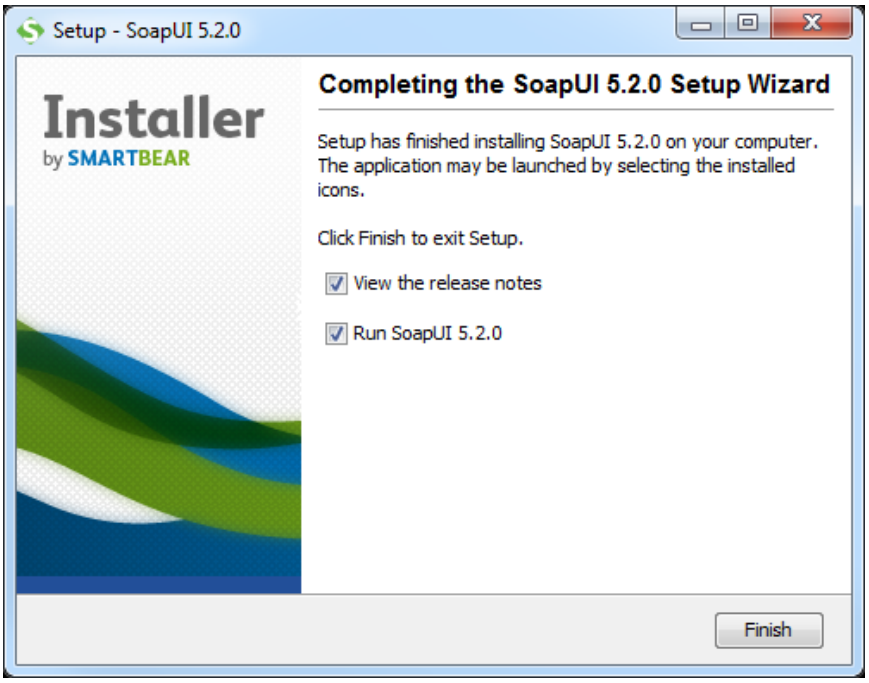
8. Choose whether you would like desktop icon to be added.



9. Click the Next button, the installation starts.



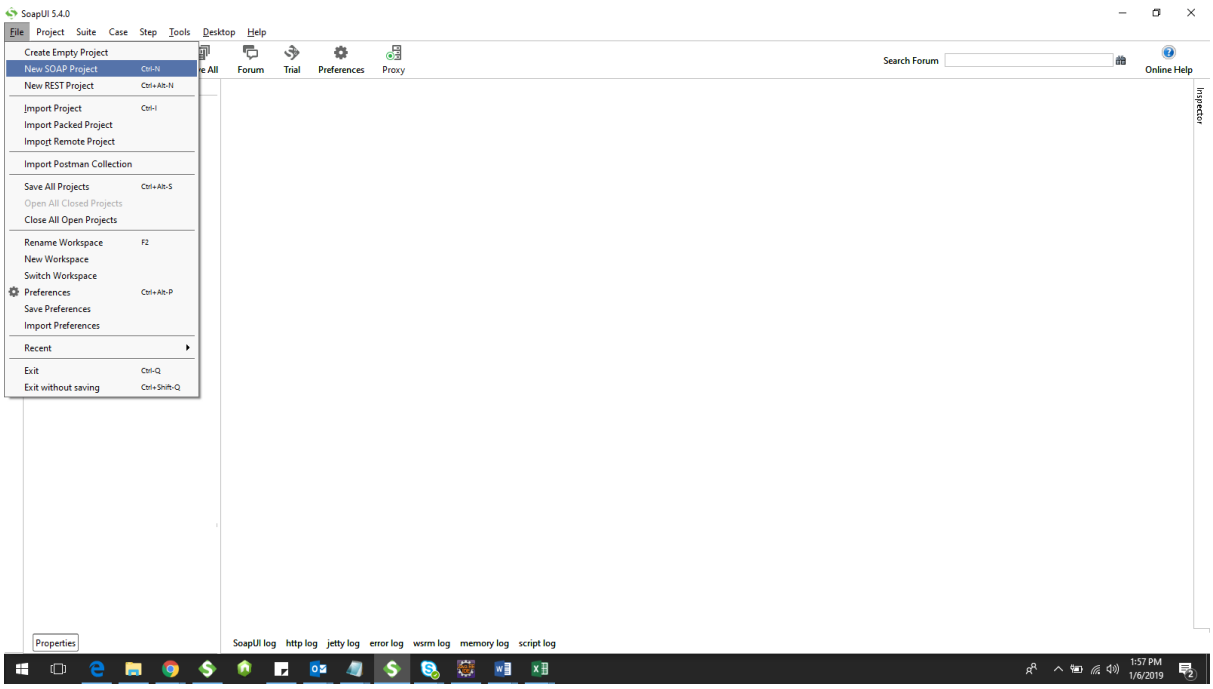
After couple of minutes at most, the installation should be finished and you should see the next screen.



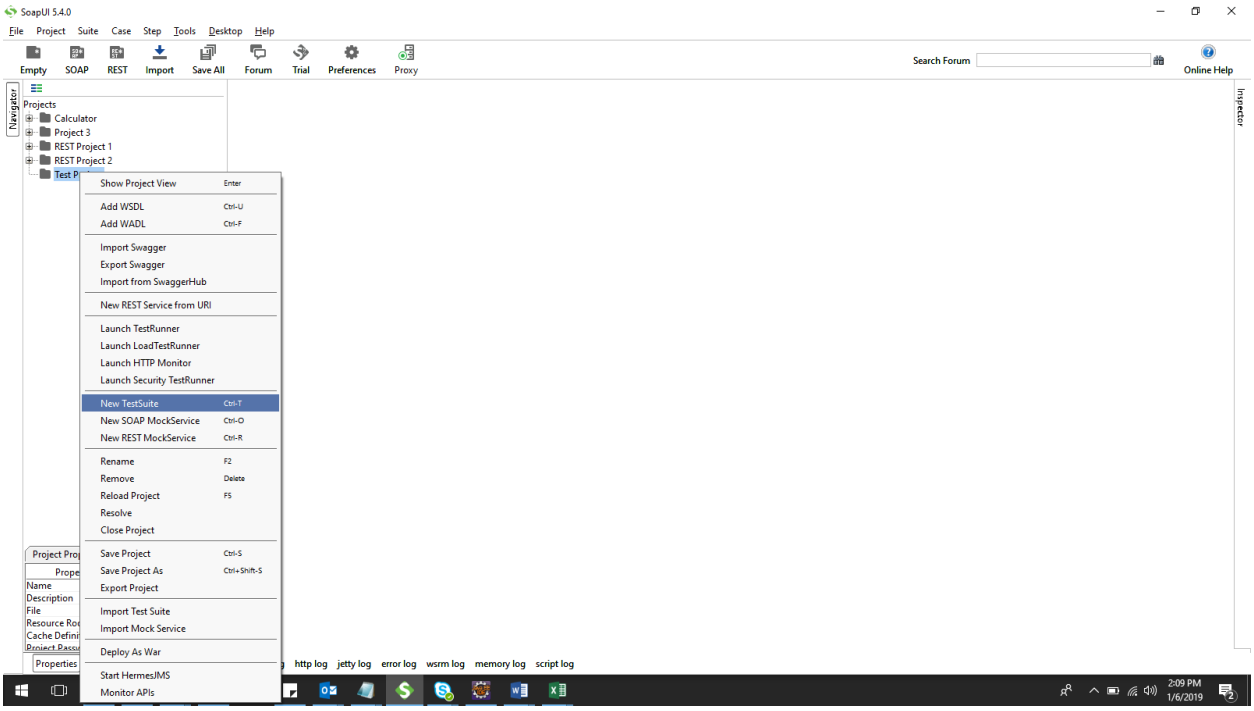
Now we are ready to use SOAP UI.

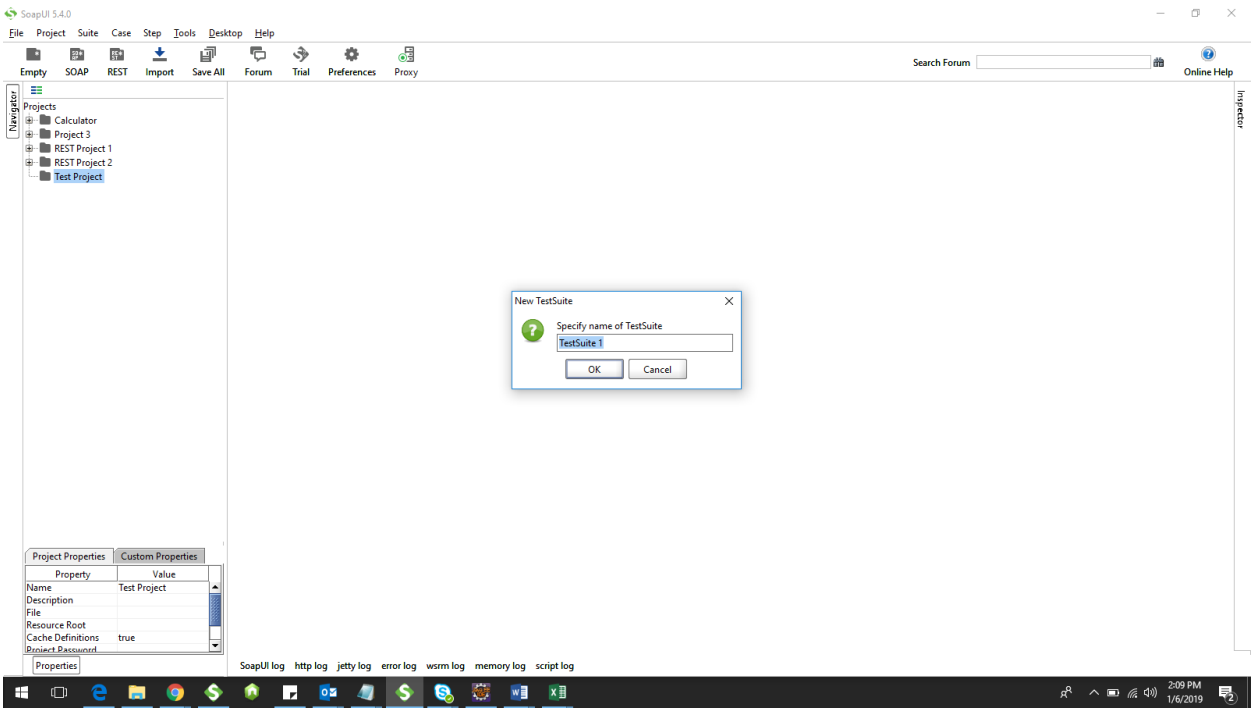
STEPS TO CREATE A NEW SOAP PROJECT:

1) Click the File→new soap project.

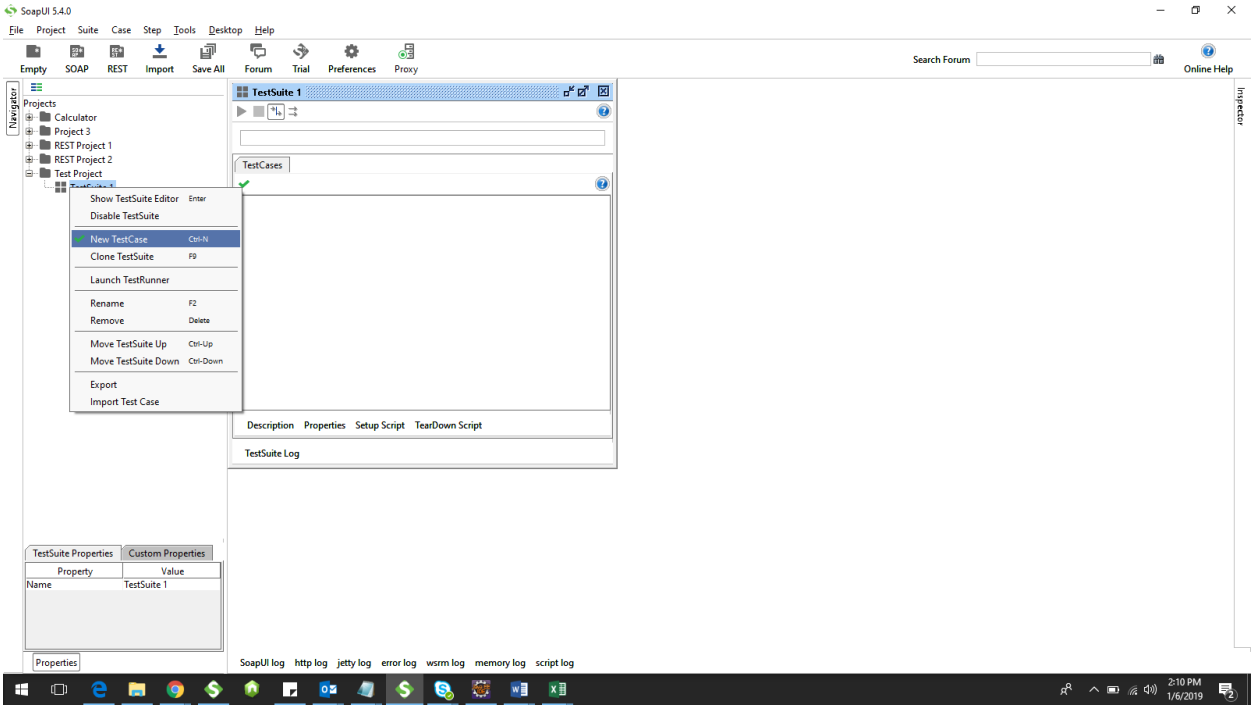


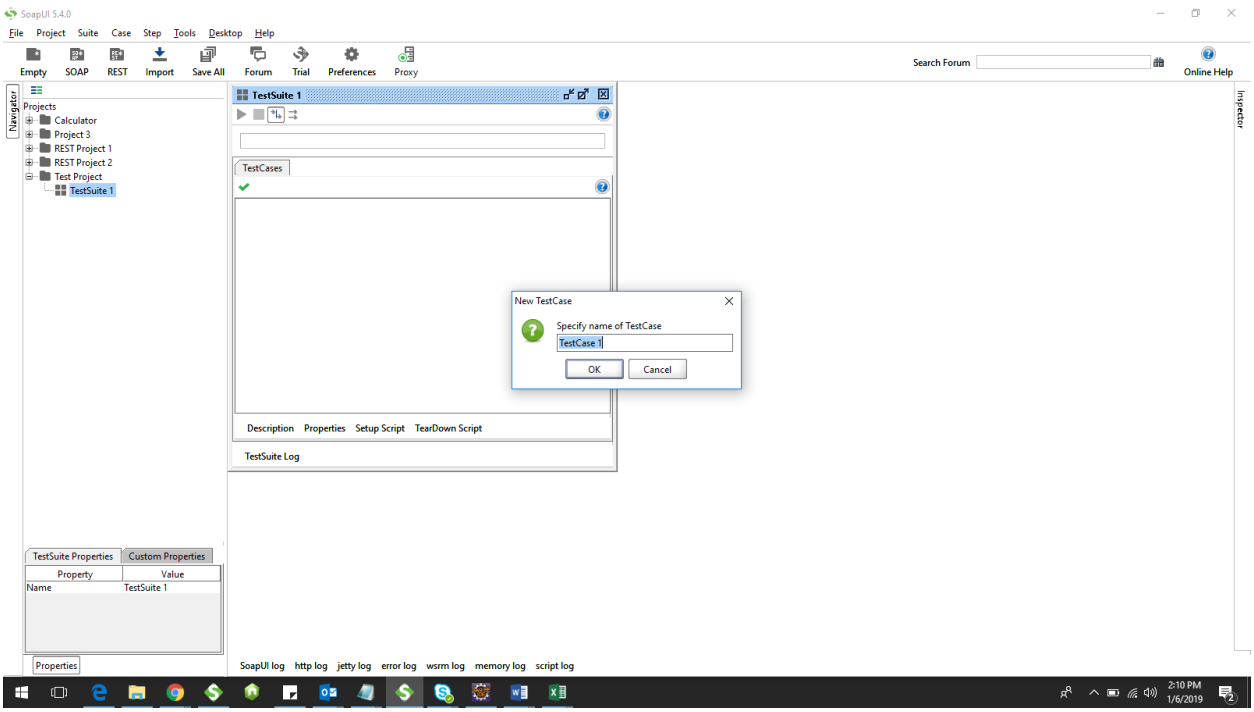
2) Right click on the project and select the new Test Suite. Then click OK (Test suite name can be changed).



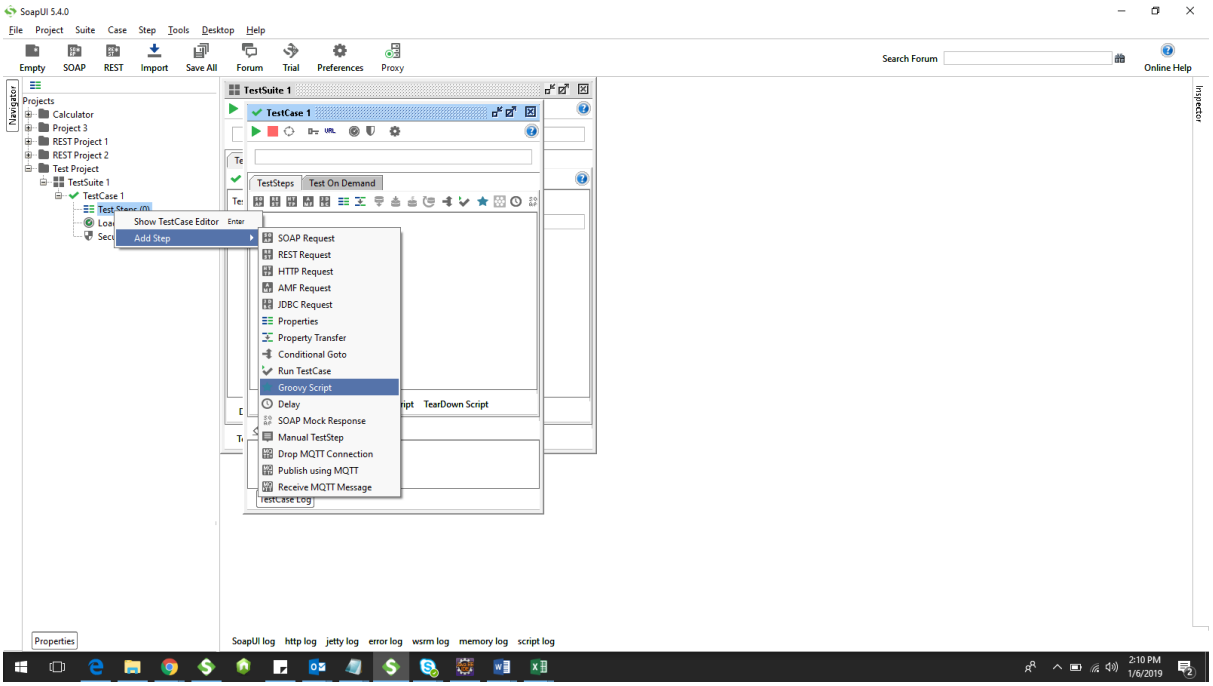


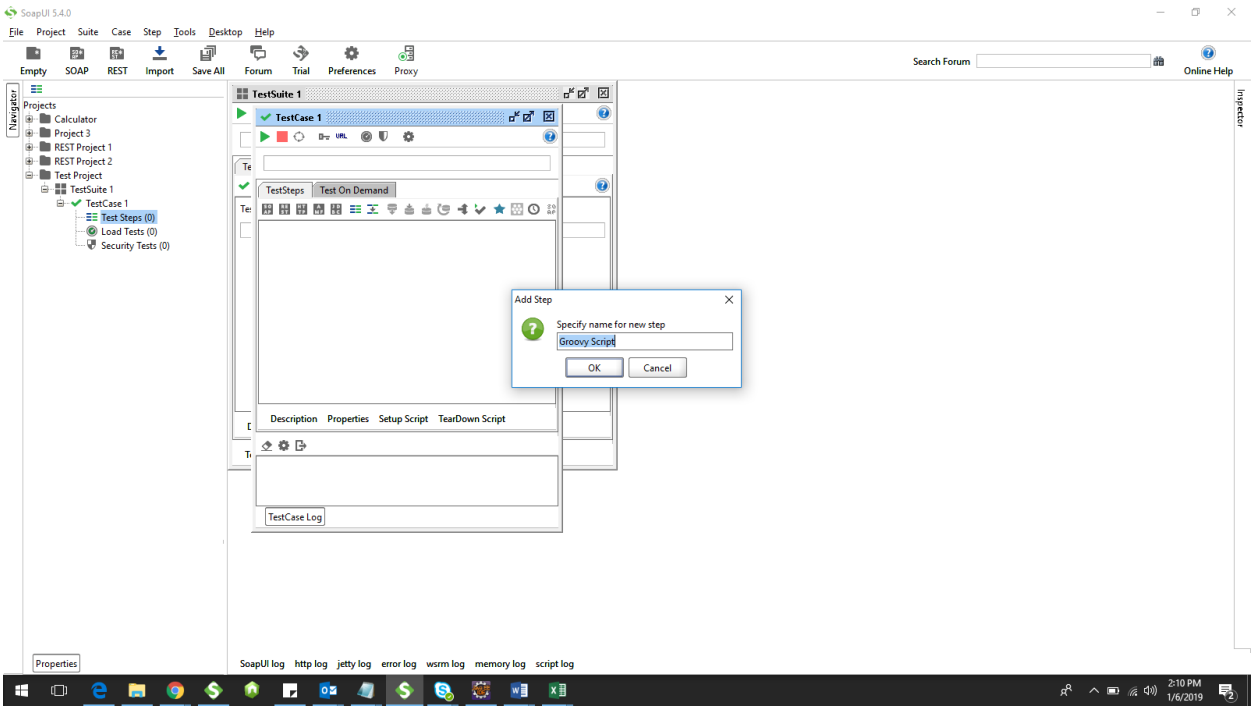
3) Right click on the test suite and select new Test case. Then click OK (Test case name can be changed).





4) Right click on the test step→Add Step→Groovy Script. Then click OK (Groovy Script name can be changed).





5) Write the code in groovy scripts and the code can be executed.

GROOVY SCRIPT TO READ FROM AN EXCEL SHEET:

```
import jxl.*
import jxl.write.*
Workbook work  = Workbook.getWorkbook(new File("D:\\Book1.xls"));
Sheet sh  = work.getSheet(0)
rc = sh.getRows();
cc = sh.getColumns();
log.info"rows count="+rc
log.info"column count="+cc
for(int i=0;i<rc;i++)
{//log.info"i="+i
for(j=0;j<cc;j++)
{

        Cell var = sh.getCell(j,i);
        log.info var.getContents()
}
}
work.close();
```

GROOVY SCRIPT TO WRITE INTO AN EXCEL SHEET:

```
import jxl.*
import jxl.write.*
WritableWorkbook workbook = Workbook.createWorkbook(new File("D:\\Book2.xls"))
WritableSheet sheet1 = workbook.createSheet("Worksheet2", 1)
int r=4
int c=2
for(i=0;i<r;i++)
{
        for(j=0;j<c;j++)
        {
                Label label = new Label(j,i,"this is cell"+j+" "+i+" ");
                sheet1.addCell(label);
        }
}
workbook.write()
workbook.close()
```

ASSIGNMENT

Web Service URL to be Tested:

1. Calculator Web service: http://www.dneonline.com/calculator.asmx?WSDL

Import the WSDL file into SOAP UI and implement the following:

1. Create a test suite with all the Services present in the Web Service.

2. Hit the Add Service with two input parameters and extract the result using Groovy scripting. 3. Use the result from Add Service and pass it in Subtract Service as Parameter 1 and use one of the input parameters used in add service as Parameter 2.

4. Extract the result from Subtract Service and pass it in Multiply Service as Parameter 1 and a new value as Parameter 2.

5. Repeat Step 4 for Division Service as well.

Note:

1. No Input Parameters should be hardcoded, instead use Properties in SOAP UI.

2. Validate results of all Services using Assertions in SOAP UI.

3. Disable SOAP UI Proxy while execution.

Groovy Script Addition:

```
import jxl.*
import java.io.*
import jxl.write.*;

import com.eviware.soapui.model.testsuite.TestRunner.Status
Workbook wb = Workbook.getWorkbook(new File("C:\\Users\\M1049091\\Desktop\\GroovyAssignment.xls"))
Sheet sh= wb.getSheet(0)

WritableWorkbook workbook1 = Workbook.createWorkbook(new File("C:\\Users\\M1049091\\Desktop\\GroovyResult.xls"))

WritableSheet sheet1 = workbook1.createSheet("Sheet1",0)

rc= sh.getRows()
cc=sh.getColumns()
log.info 'row count' + rc
log.info 'column count' + cc
Label labelOperation = new Label(0, 0, "Operation");
Label labelFirstNumber = new Label(1, 0, "FirstNumber");
Label labelSecondNumber = new Label(2, 0, "SecondNumber");
Label labelResult = new Label(3, 0, "Result");
Label labelExpectedResult = new Label(4, 0, "Expected Result");
Label labelStatus = new Label(5, 0, "Response Status");
sheet1.addCell(labelOperation);
sheet1.addCell(labelFirstNumber);
sheet1.addCell(labelSecondNumber);
sheet1.addCell(labelResult);
sheet1.addCell(labelExpectedResult);
sheet1.addCell(labelStatus);


for (i=1; i<2;i++)
{
        for(j=0;j<cc;j++)
        {
                log.info 'j= '+ j
                Cell var=sh.getCell(j,i)
```

```
                log.info var.getContents()
                if(j==0)
                {Operation  = var.getContents()

                Label label = new Label(0, i, Operation);
                sheet1.addCell(label);
                }
                if(j==1)
                {  FirstNumber  = var.getContents()

                Label label = new Label(1, i, FirstNumber);
                sheet1.addCell(label);
                }
                else
                {
                  SecondNumber = var.getContents()

                Label label = new Label(2, i, SecondNumber);
                sheet1.addCell(label);
                }
        }

Integer value1= FirstNumber.toInteger()
Integer value2= SecondNumber.toInteger()
Integer sum= value1+value2
ExpectedResult= sum.toString()
            Label labelExpectedResultWrite = new Label(4, i, sum.toString());
sheet1.addCell(labelExpectedResultWrite);
def request= '''
<soapenv:Envelope            xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:Add>
      <tem:intA>'''+ FirstNumber+'''</tem:intA>
      <tem:intB>'''+ SecondNumber+'''</tem:intB>
    </tem:Add>
  </soapenv:Body>
</soapenv:Envelope>
'''
testRunner.testCase.testSteps["Add"].setPropertyValue( 'Request', request )
testRunner.runTestStepByName( "Add")
```

```
def groovyUtils=new com.eviware.soapui.support.GroovyUtils(context)
 def res = groovyUtils.getXmlHolder("Add#Response")
 def xmlData = res.getXmlObject()
 String str=xmlData.toString()
 def a=str.split("<AddResult>")[1].split("</AddResult>")
 log.info a[0]
 Label labelAddResult = new Label(3, 1, a[0]);
sheet1.addCell(labelAddResult);
def POSTmethod=testRunner
   .testCase
  .getTestStepByName("Add").getTestRequest()
assertions = POSTmethod.getAssertions()
stat = (assertions.status).toString()
   if(stat == "FAILED"){
      log.info "FAIL"

      Label label2 = new Label(5, i, "FAIL");
sheet1.addCell(label12);
   }
   else
    { log.info "PASS"
     Label label3 = new Label(5, i, "PASS");
sheet1.addCell(label3);



   }
                  workbook1.write()
         workbook1.close()
           }
```

Groovy Script Subtraction:

```
import jxl.*
import java.io.*
import jxl.write.*;

import com.eviware.soapui.model.testsuite.TestRunner.Status

Workbook wb= Workbook.getWorkbook(new
File("C:\\Users\\M1049091\\Desktop\\GroovyResult.xls"))
Sheet sh= wb.getSheet(0)
```

```
Cell var1= sh.getCell(3,1)
FirstNumber  = var1.getContents()
                log.info 'num1 '+ FirstNumber
                Label labelFirstNumber = new Label(1, 2, FirstNumber);
                Cell var2= sh.getCell(2,1)
                SecondNumber= var2.getContents()
                log.info 'num2 '+ SecondNumber
Label labelSecondNumber = new Label(2, 2, SecondNumber);
Label labelOperation = new Label(0, 2, 'Subtraction');


Workbook existingWorkbook = Workbook.getWorkbook(new
File("C:\\Users\\M1049091\\Desktop\\GroovyResult.xls"));
WritableWorkbook workbookCopy = Workbook.createWorkbook(new
File("C:\\Users\\M1049091\\Desktop\\GroovyResult.xls"), existingWorkbook);

WritableSheet sheet1 = workbookCopy.getSheet("Sheet1")
sheet1.addCell(labelOperation);
sheet1.addCell(labelFirstNumber);
sheet1.addCell(labelSecondNumber);

Integer value1= FirstNumber.toInteger()
Integer value2= SecondNumber.toInteger()
Integer difference= value1-value2
ExpectedResult= difference.toString()
        Label labelExpectedResultWrite = new Label(4, 2, difference.toString());
sheet1.addCell(labelExpectedResultWrite);

def request= '''
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:Subtract>
      <tem:intA>'''+ FirstNumber+'''</tem:intA>
      <tem:intB>'''+ SecondNumber+'''</tem:intB>
    </tem:Subtract>
  </soapenv:Body>
</soapenv:Envelope>
'''
testRunner.testCase.testSteps["Subtract"].setPropertyValue( 'Request', request )
testRunner.runTestStepByName( "Subtract")
```

```
def groovyUtils=new com.eviware.soapui.support.GroovyUtils(context)
 def res = groovyUtils.getXmlHolder("Subtract#Response")
 def xmlData = res.getXmlObject()
 String str=xmlData.toString()
 def a=str.split("<SubtractResult>")[1].split("</SubtractResult>")
 log.info a[0]
 Label labelAddResult = new Label(3, 2, a[0]);
sheet1.addCell(labelAddResult);

def POSTmethod=testRunner
   .testCase
  .getTestStepByName("Subtract").getTestRequest()
assertions = POSTmethod.getAssertions()
stat = (assertions.status).toString()
   if(stat == "FAILED"){
      log.info "FAIL"

      Label label2 = new Label(5, 2, "FAIL");
sheet1.addCell(label12);
   }
   else
    { log.info "PASS"
     Label label3 = new Label(5, 2, "PASS");
sheet1.addCell(label3);
    }
workbookCopy.write()
workbookCopy.close()
```

Groovy Script Multiplication:

```
import jxl.*
import java.io.*
import jxl.write.*;

import com.eviware.soapui.model.testsuite.TestRunner.Status

Workbook wb= Workbook.getWorkbook(new
File("C:\\Users\\M1049091\\Desktop\\GroovyAssignment.xls"))
Sheet sh= wb.getSheet(0)
cc=sh.getColumns()
          for(j=0;j<cc;j++)
```

```
{
        log.info 'j= '+ j
        Cell var=sh.getCell(j,3)
        log.info var.getContents()
        if(j==0)
        {Operation  = var.getContents()



        }

        else if(j==2)
        {
          SecondNumber = var.getContents()



        }
}
Label labelOperation = new Label(0, 4, Operation);
Label labelSecondNumber = new Label(2, 4, SecondNumber);

Workbook wb1= Workbook.getWorkbook(new File("C:\\Users\\M1049091\\Desktop\\GroovyResult.xls"))
Sheet sh1= wb1.getSheet(0)

Cell var1= sh1.getCell(3,3)
FirstNumber  = var1.getContents()
        log.info 'num1 '+ FirstNumber
        Label labelFirstNumber = new Label(1, 4, FirstNumber);

        Workbook existingWorkbook = Workbook.getWorkbook(new File("C:\\Users\\M1049091\\Desktop\\GroovyResult.xls"));
WritableWorkbook workbookCopy = Workbook.createWorkbook(new File("C:\\Users\\M1049091\\Desktop\\GroovyResult.xls"), existingWorkbook);

WritableSheet sheet1 = workbookCopy.getSheet("Sheet1")
sheet1.addCell(labelOperation);
sheet1.addCell(labelFirstNumber);
sheet1.addCell(labelSecondNumber);

Integer value1= FirstNumber.toInteger()
```

```
Integer value2= SecondNumber.toInteger()
Integer product= value1*value2
ExpectedResult= product.toString()
            Label labelExpectedResultWrite = new Label(4, 4, product.toString());
sheet1.addCell(labelExpectedResultWrite);

def request= '''
<soapenv:Envelope            xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:Multiply>
      <tem:intA>'''+FirstNumber+'''</tem:intA>
      <tem:intB>'''+SecondNumber+'''</tem:intB>
    </tem:Multiply>
  </soapenv:Body>
</soapenv:Envelope>
'''
testRunner.testCase.testSteps["Multiply"].setPropertyValue( 'Request', request )
testRunner.runTestStepByName( "Multiply")
def groovyUtils=new com.eviware.soapui.support.GroovyUtils(context)
 def res = groovyUtils.getXmlHolder("Multiply#Response")
 def xmlData = res.getXmlObject()
 String str=xmlData.toString()
 def a=str.split("<MultiplyResult>")[1].split("</MultiplyResult>")
 log.info a[0]
 Label labelAddResult = new Label(3, 4, a[0]);
sheet1.addCell(labelAddResult);

def POSTmethod=testRunner
  .testCase
 .getTestStepByName("Multiply").getTestRequest()
assertions = POSTmethod.getAssertions()
stat = (assertions.status).toString()
  if(stat == "FAILED"){
    log.info "FAIL"

    Label label2 = new Label(5, 4, "FAIL");
sheet1.addCell(label12);
  }
  else
   { log.info "PASS"
```

```
    Label label3 = new Label(5, 4, "PASS");
sheet1.addCell(label3);
   }

        workbookCopy.write()
workbookCopy.close()
```

Groovy Script for Division:

```
import jxl.*
import java.io.*
import jxl.write.*;

import com.eviware.soapui.model.testsuite.TestRunner.Status

Workbook wb= Workbook.getWorkbook(new
File("C:\\Users\\M1049091\\Desktop\\GroovyAssignment.xls"))
Sheet sh= wb.getSheet(0)
cc=sh.getColumns()
        for(j=0;j<cc;j++)
        {
                log.info 'j= '+ j
                Cell var=sh.getCell(j,2)
                log.info var.getContents()
                if(j==0)
                {Operation  = var.getContents()



                }

                else if(j==2)
                {
                  SecondNumber = var.getContents()



                }
        }
        Label labelOperation = new Label(0, 3, Operation);
        Label labelSecondNumber = new Label(2, 3, SecondNumber);
```

```
        Workbook wb1= Workbook.getWorkbook(new File("C:\\Users\\M1049091\\Desktop\\GroovyResult.xls"))
Sheet sh1= wb1.getSheet(0)

Cell var1= sh1.getCell(3,2)
FirstNumber  = var1.getContents()
                log.info 'num1 '+ FirstNumber
                Label labelFirstNumber = new Label(1, 3, FirstNumber);

                Workbook existingWorkbook = Workbook.getWorkbook(new File("C:\\Users\\M1049091\\Desktop\\GroovyResult.xls"));
WritableWorkbook workbookCopy = Workbook.createWorkbook(new File("C:\\Users\\M1049091\\Desktop\\GroovyResult.xls"), existingWorkbook);

WritableSheet sheet1 = workbookCopy.getSheet("Sheet1")
sheet1.addCell(labelOperation);
sheet1.addCell(labelFirstNumber);
sheet1.addCell(labelSecondNumber);

Integer value1= FirstNumber.toInteger()
Integer value2= SecondNumber.toInteger()
Integer quotient= value1/value2
ExpectedResult= quotient.toString()
            Label labelExpectedResultWrite = new Label(4, 3, quotient.toString());
sheet1.addCell(labelExpectedResultWrite);


def request= '''
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:Divide>
      <tem:intA>'''+FirstNumber+'''</tem:intA>
      <tem:intB>'''+SecondNumber+'''</tem:intB>
    </tem:Divide>
  </soapenv:Body>
</soapenv:Envelope>
'''
testRunner.testCase.testSteps["Divide"].setPropertyValue( 'Request', request )
testRunner.runTestStepByName( "Divide")
```

```
def groovyUtils=new com.eviware.soapui.support.GroovyUtils(context)
 def res = groovyUtils.getXmlHolder("Divide#Response")
 def xmlData = res.getXmlObject()
 String str=xmlData.toString()
 def a=str.split("<DivideResult>")[1].split("</DivideResult>")
 log.info a[0]
 Label labelAddResult = new Label(3, 3, a[0]);
sheet1.addCell(labelAddResult);

def POSTmethod=testRunner
   .testCase
  .getTestStepByName("Divide").getTestRequest()
assertions = POSTmethod.getAssertions()
stat = (assertions.status).toString()
   if(stat == "FAILED"){
      log.info "FAIL"

      Label label2 = new Label(5, 3, "FAIL");
sheet1.addCell(label12);
   }
   else
    { log.info "PASS"
     Label label3 = new Label(5, 3, "PASS");
sheet1.addCell(label3);
    }

          workbookCopy.write()
workbookCopy.close()
```

REFERENCE LINKS:

- https://www.soapui.org/getting-started.html
- https://youtu.be/XolvQaeRglk