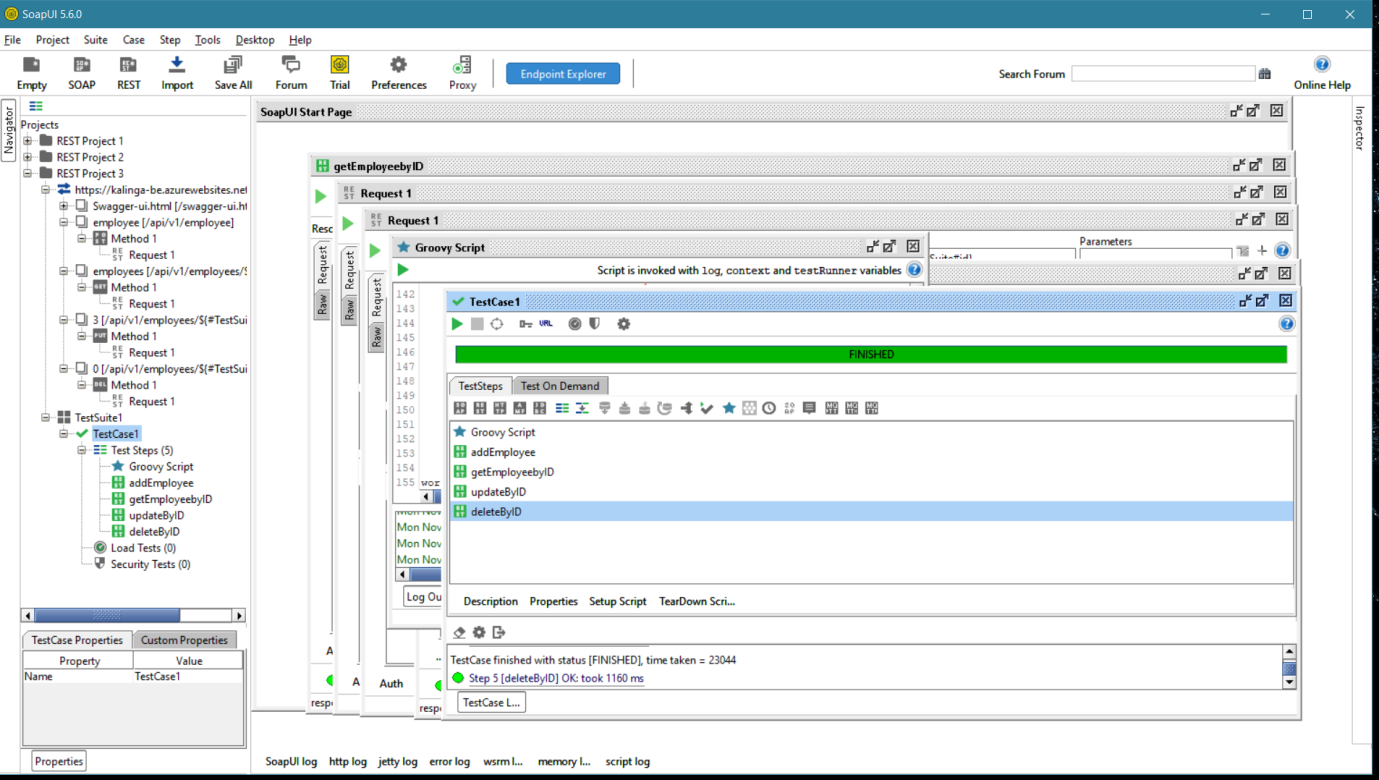
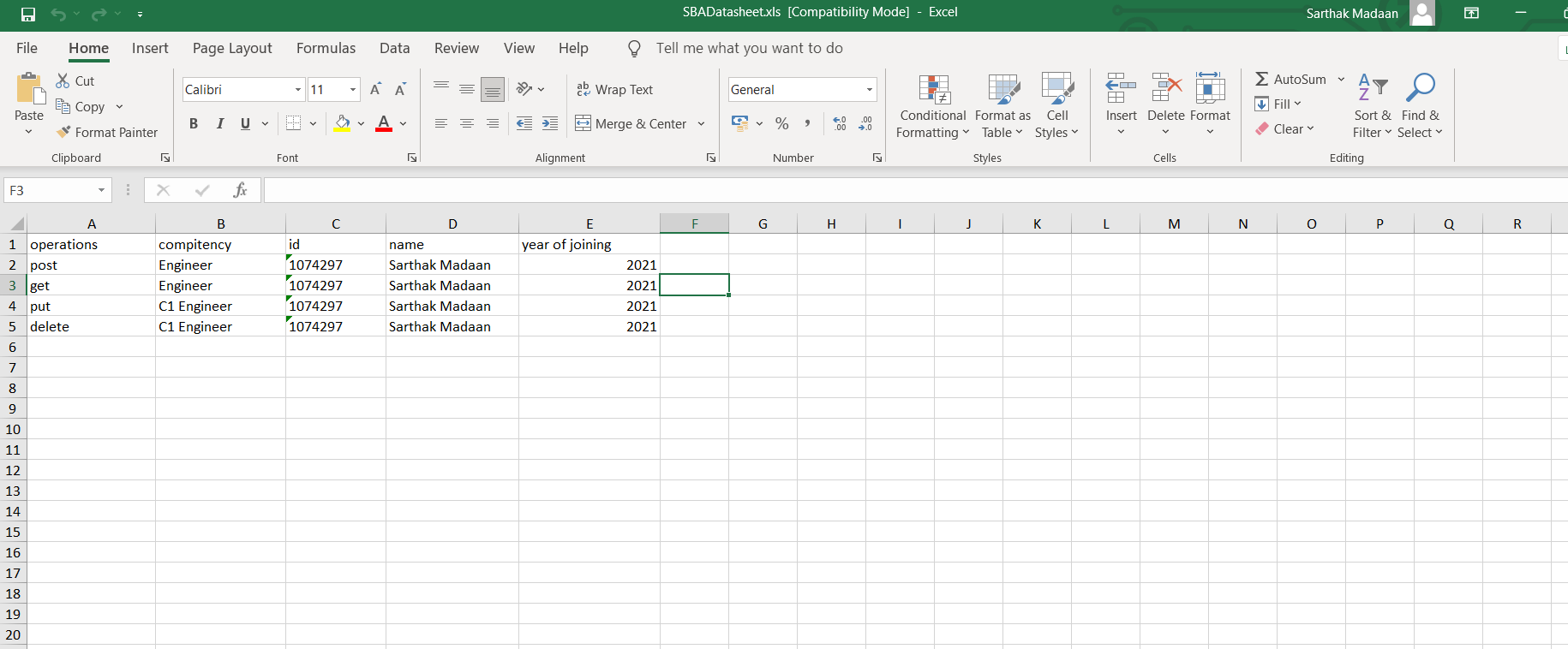
Comprehensive assessment

Test Successful



Excel sheet



Groovy script:

import groovy.json.JsonSlurper

import jxl.\*

import jxl.write.\*

Workbook work = Workbook.getWorkbook(new File("C:\Users\sarthak madaan\Desktop\ms5"));

Sheet sht = work.getSheet(0)

rc = sht.getRows();

cc = sht.getColumns();

log.info"rows count="+rc

log.info"column count="+cc

for(int i=1;i<rc;i++)

{

String option=sht.getCell(0,i).getContents()

switch(option)

{

case "post":

log.info "post"

def data=testRunner.testCase.testSuite

def testStep = testRunner.testCase.testSteps["addEmployee"]

for(int ii=testStep.assertionCount-1; ii>=0; ii--) {

testStep.removeAssertion(testStep.getAssertionAt(ii));

}

def Assertion = testRunner.testCase.testSteps["addEmployee"].addAssertion("Contains")

for(int j=1;j<cc;j++)

{

if(j==1)

{

data.setPropertyValue("compitency",sht.getCell(j,i).getContents())

Assertion.setToken(sht.getCell(j,i).getContents())

}

else if(j==2)

{

data.setPropertyValue("id",sht.getCell(j,i).getContents())

Assertion.setToken(sht.getCell(j,i).getContents())

}

else if(j==3)

{

data.setPropertyValue("name",sht.getCell(j,i).getContents())

Assertion.setToken(sht.getCell(j,i).getContents())

}

else

{

data.setPropertyValue("yearJoined",sht.getCell(j,i).getContents())

Assertion.setToken(sht.getCell(j,i).getContents())

}

}

def tStep=data.testCases["TestCase1"].testSteps["addEmployee"]

tStep.run(testRunner,context)

def json=tStep.testRequest.response.responseContent

def tJson=new JsonSlurper()

def object=tJson.parseText(json)

assert object.toString().contains(sht.getCell(1,i).getContents())==true

log.info (object)

Thread.sleep(5000)

break

case "get":

log.info "get"

def data=testRunner.testCase.testSuite

def testStep = testRunner.testCase.testSteps["getEmployeebyID"]

for(int ii=testStep.assertionCount-1; ii>=0; ii--) {

testStep.removeAssertion(testStep.getAssertionAt(ii));

}

def Assertion = testRunner.testCase.testSteps["getEmployeebyID"].addAssertion("Contains")

for(int j=1;j<cc;j++)

{

if(j==1)

{

Assertion.setToken(sht.getCell(j,i).getContents())

}

else if(j==2)

{

Assertion.setToken(sht.getCell(j,i).getContents())

}

else if(j==2)

{

Assertion.setToken(sht.getCell(j,i).getContents())

}

else

{

Assertion.setToken(sht.getCell(j,i).getContents())

}

}

def tStep=data.testCases["TestCase1"].testSteps["getEmployeebyID"]

tStep.run(testRunner,context)

def json=tStep.testRequest.response.responseContent

def tJson=new JsonSlurper()

def object=tJson.parseText(json)

assert object.toString().contains(sht.getCell(1,i).getContents())==true

log.info (object)

Thread.sleep(3000)

break

case "put":

log.info "put"

def data=testRunner.testCase.testSuite

def testStep = testRunner.testCase.testSteps["updateByID"]

for(int ii=testStep.assertionCount-1; ii>=0; ii--) {

testStep.removeAssertion(testStep.getAssertionAt(ii));

}

def Assertion = testRunner.testCase.testSteps["updateByID"].addAssertion("Contains")

for(int j=1;j<cc;j++)

{

if(j==1)

{

data.setPropertyValue("compitency",sht.getCell(j,i).getContents())

Assertion.setToken(sht.getCell(j,i).getContents())

}

else if(j==2)

{

data.setPropertyValue("id",sht.getCell(j,i).getContents())

Assertion.setToken(sht.getCell(j,i).getContents())

}

else if(j==3)

{

data.setPropertyValue("name",sht.getCell(j,i).getContents())

Assertion.setToken(sht.getCell(j,i).getContents())

}

else

{

data.setPropertyValue("yearJoined",sht.getCell(j,i).getContents())

Assertion.setToken(sht.getCell(j,i).getContents())

}

}

def tStep=data.testCases["TestCase1"].testSteps["updateByID"]

tStep.run(testRunner,context)

def json=tStep.testRequest.response.responseContent

def tJson=new JsonSlurper()

def object=tJson.parseText(json)

assert object.toString().contains(sht.getCell(2,i).getContents())==true

log.info (object)

Thread.sleep(3000)

break

case "delete":

log.info "delete"

def data=testRunner.testCase.testSuite

def testStep = testRunner.testCase.testSteps["deleteByID"]

for(int ii=testStep.assertionCount-1; ii>=0; ii--) {

testStep.removeAssertion(testStep.getAssertionAt(ii));

}

def Assertion = testRunner.testCase.testSteps["deleteByID"].addAssertion("Contains")

Assertion.setToken("DELETED")

def tStep=data.testCases["TestCase1"].testSteps["deleteByID"]

tStep.run(testRunner,context)

def json=tStep.testRequest.response.responseContent

def tJson=new JsonSlurper()

def object=tJson.parseText(json)

assert object.toString().contains("DELETED")==true

log.info (object)

break

}

}

work.close();

