

SOAP and REST-API Testing

A. Calculator Web Services Example

1. Create New Project and give the details of WSDL file

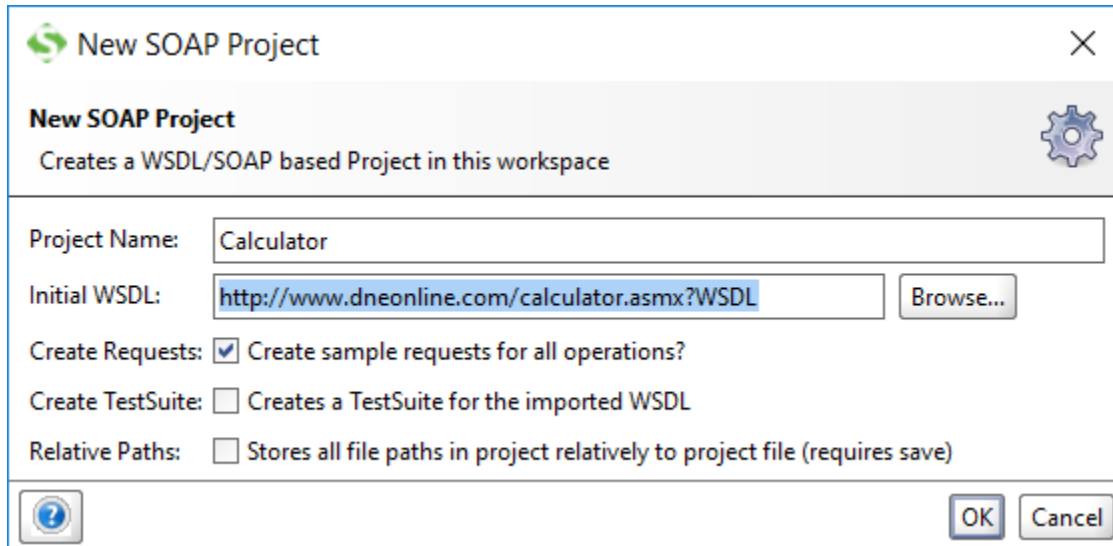


Figure 1- New Project Creation

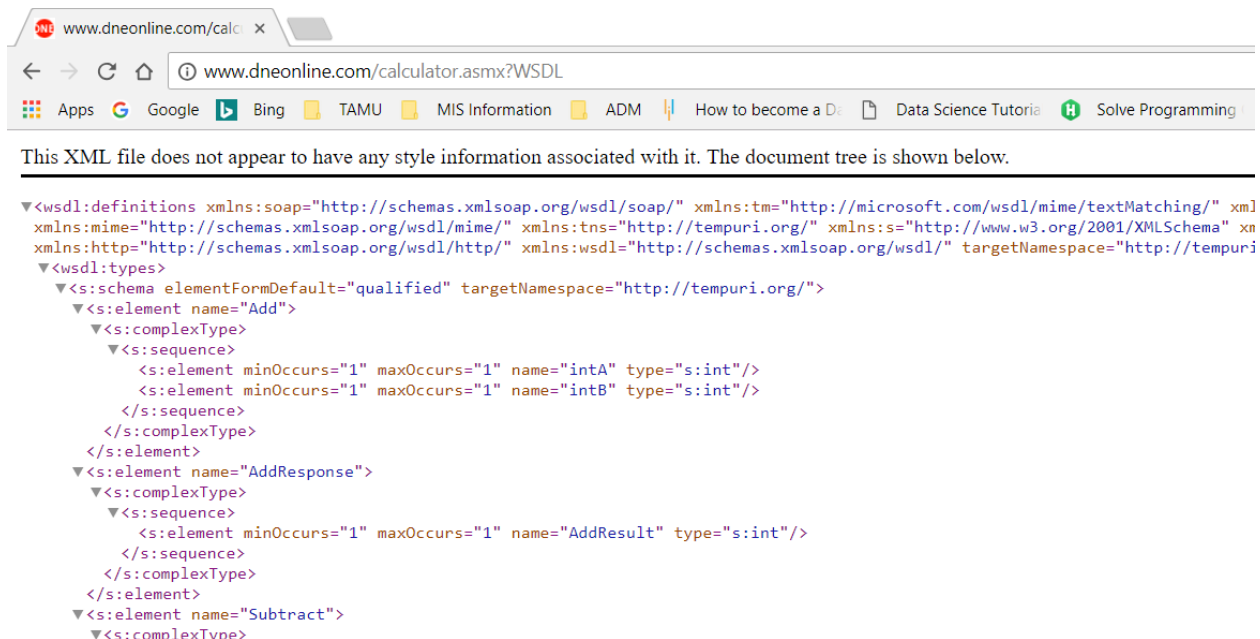


Figure 2- WSDL File URL

2. Click OK and all look for all the requests generated by this web service. Interface Properties for the web service are populated.

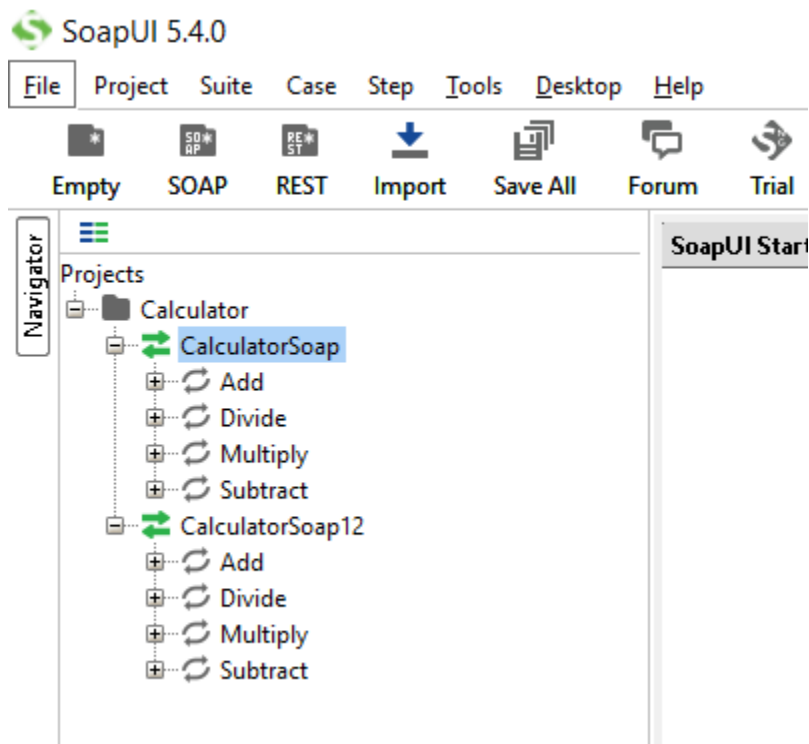


Figure 3 - Generated Requests in the Web Service

Two folders for two different versions are created.

CalculatorSoap12 is the latest version.

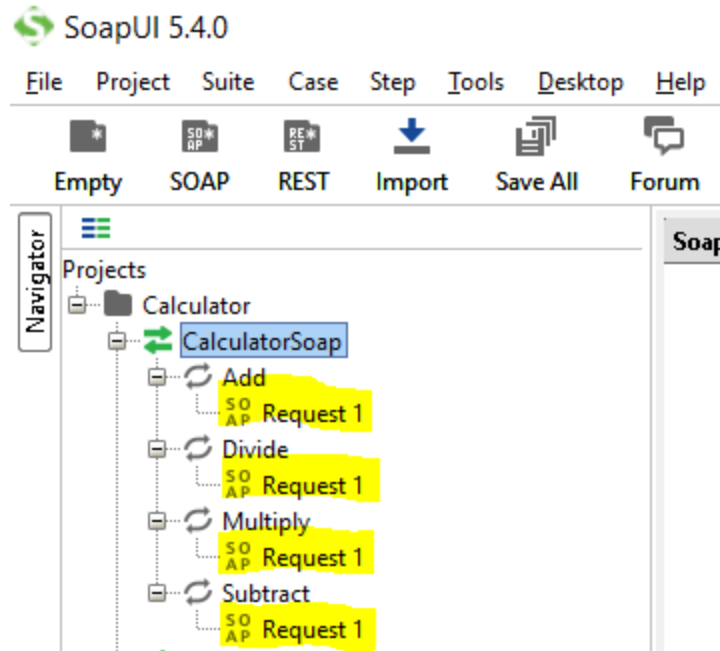


Figure 4 - Expanded Requests in the Web Service

Interface Properties	
Property	Value
Name	CalculatorSoap
Description	
Definition URL	http://www.dneonline.c...
Binding	{http://tempuri.org/}Cal...
SOAP Version	SOAP 1.1
Cached	true
Style	Document
WS-A version	NONE
WS-A anonymous	optional

Figure 5 - Interface Properties

3. Double Click on Request1 under CALCULATORSOAP12, Addition

<http://www.dneonline.com/calculator.asmx> is the end point.

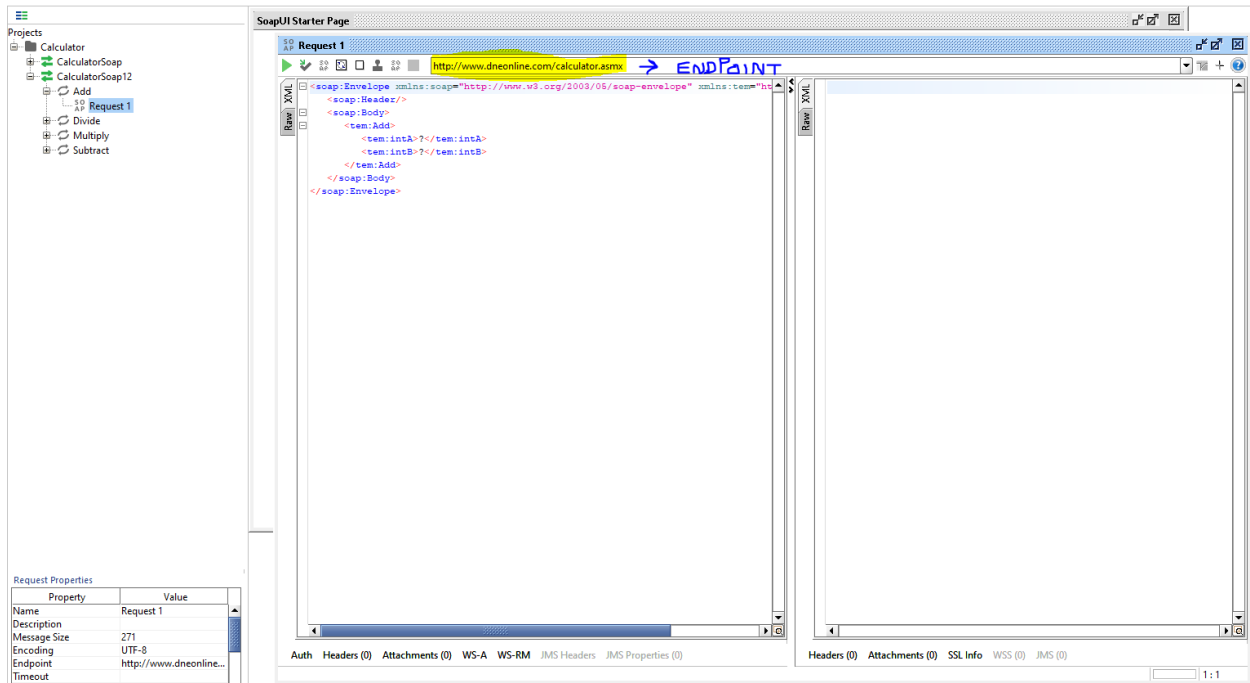


Figure 6 - Addition Request with Endpoint

'?' symbol indicates the parameters to be passed.

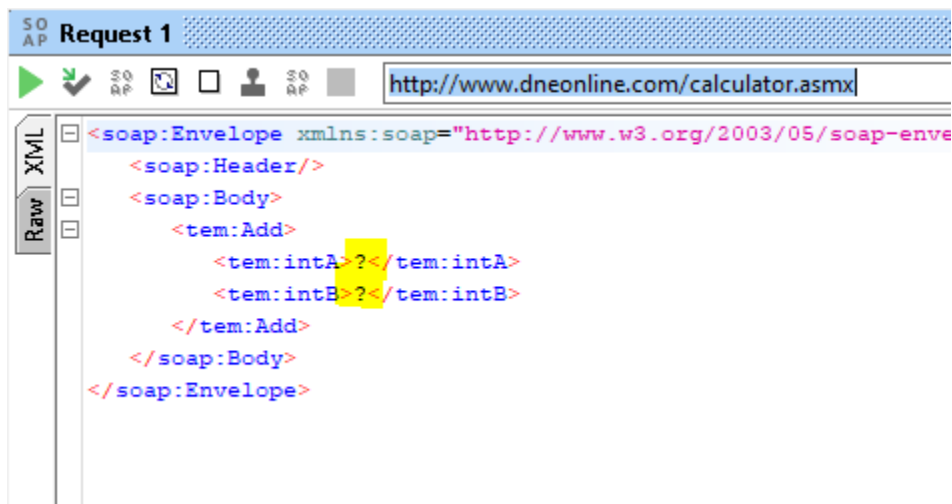


Figure 7 - Parameters to be passed indicated by '?' symbol

4. Enter numbers in place of parameters and click submit button (Green Triangle) at top

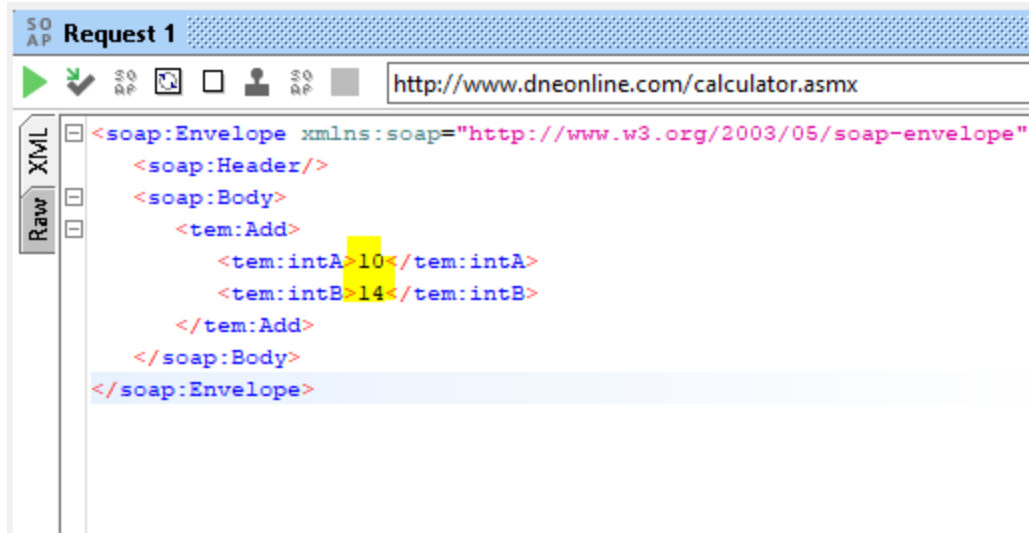


Figure 8 - Insert Parameters in Request

5. Evaluate the Response



Figure 9 - Response in XML form

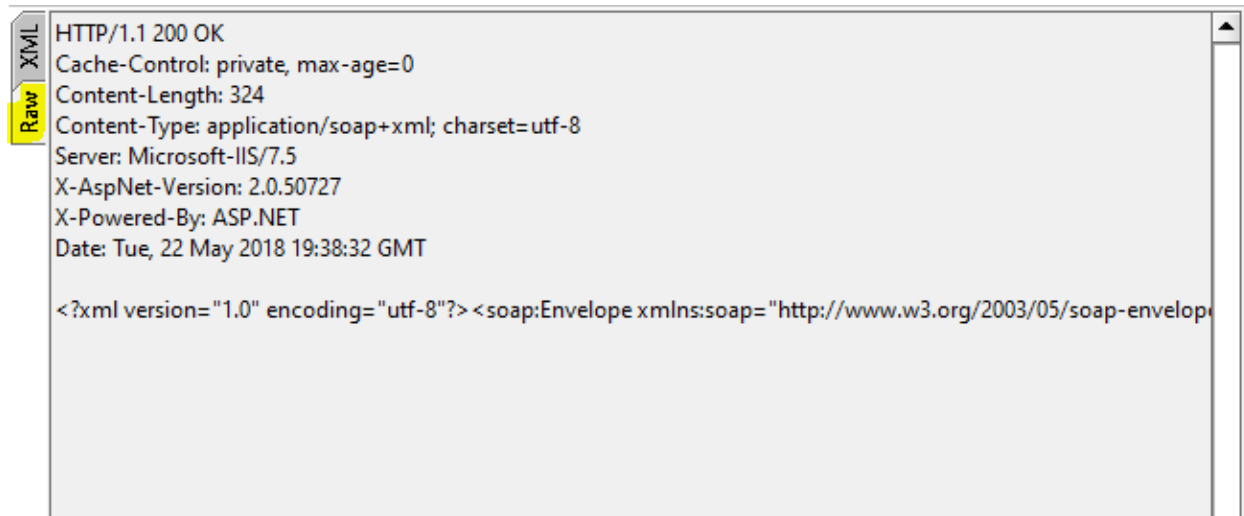


Figure 10 - Response in Raw Form

B. Creating Test Suites and test cases

1. Create Test Suite

Right Click on Project and create new Test Suite (Ctrl + T)

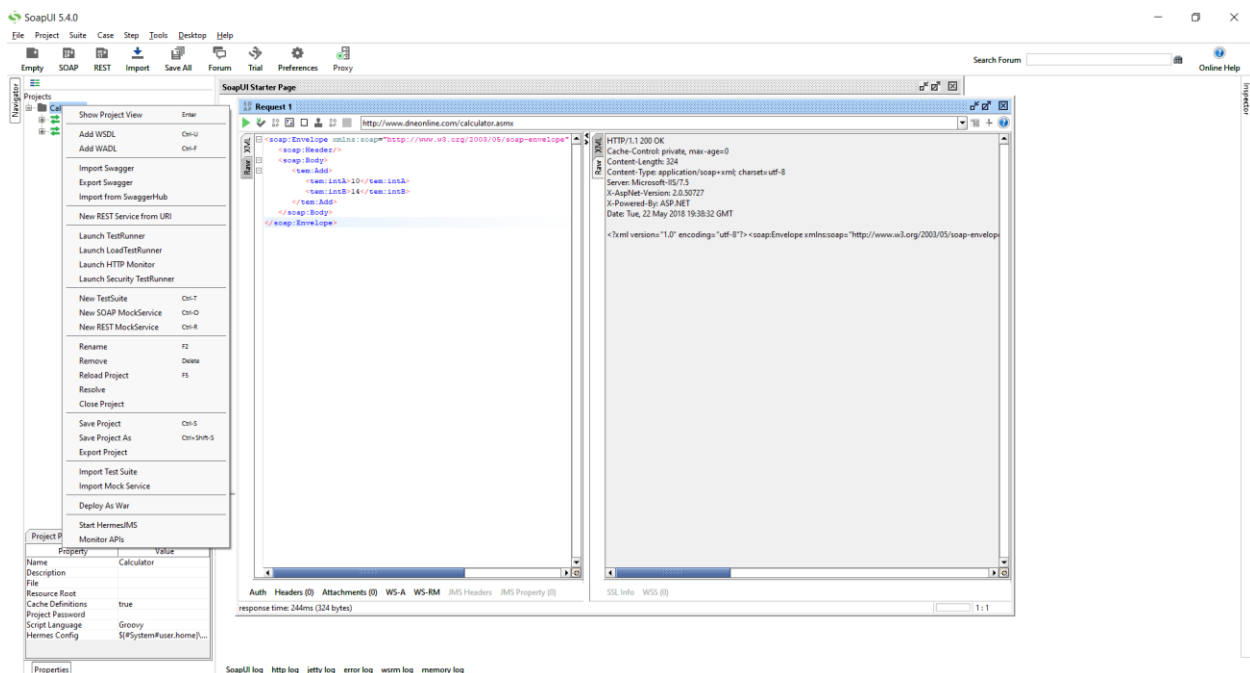


Figure 11 - Create New Test Suite

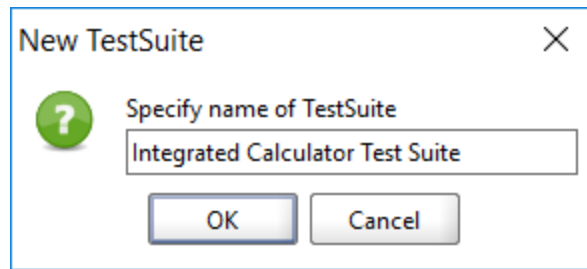


Figure 12 - Name the test suite

New Folder named Integrated Calculator Test Suite us created.

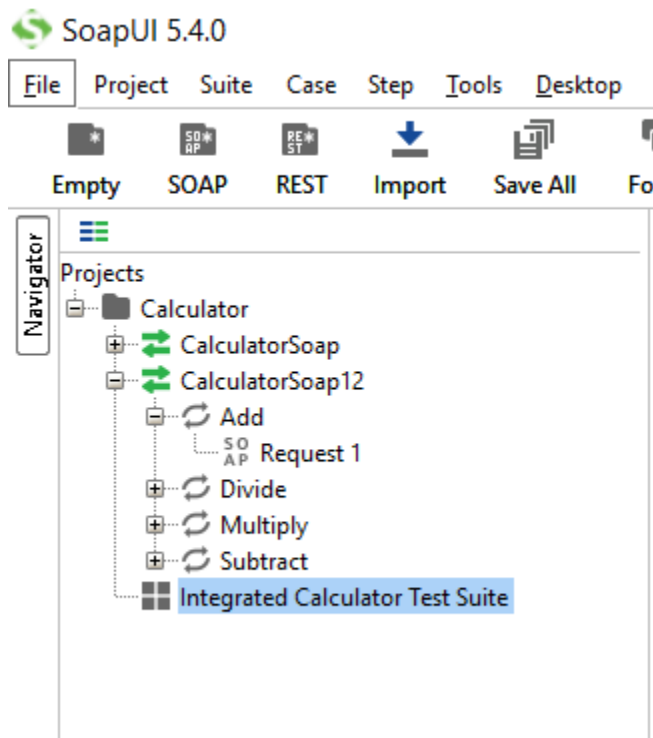


Figure 13 - New folder for Test Suite is created

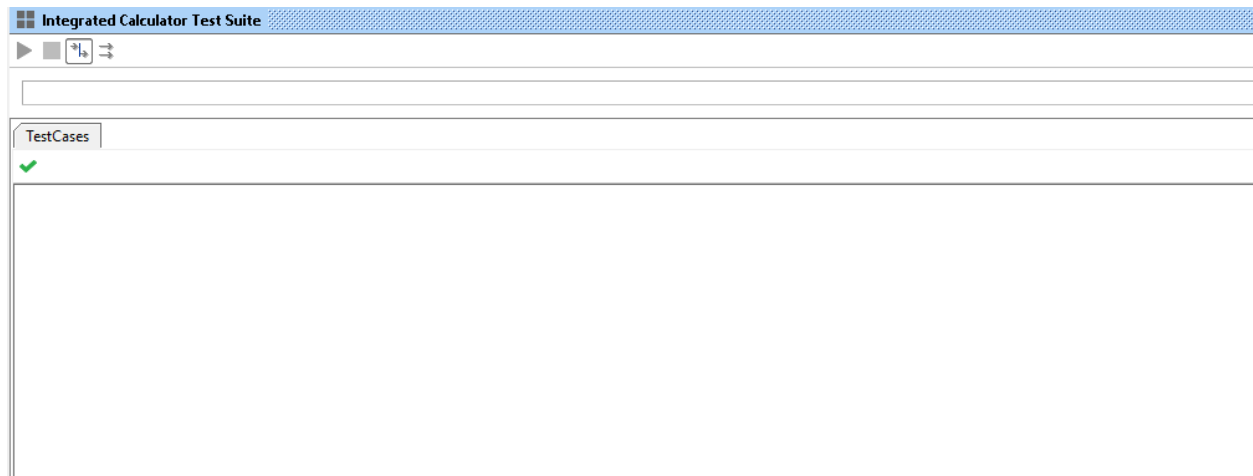


Figure 14 - Test Suite Window

2. Create Test Cases in the Test Suite

Right Click on the Test Suite folder and select **New Test Case**

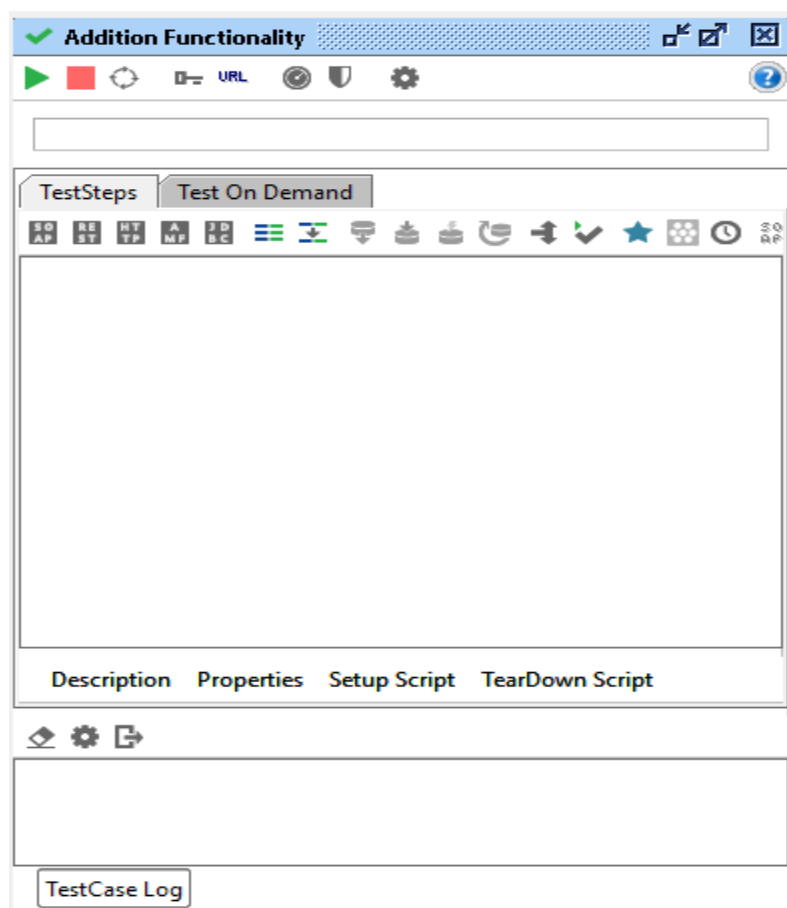


Figure 15 - Test Case to test addition functionality

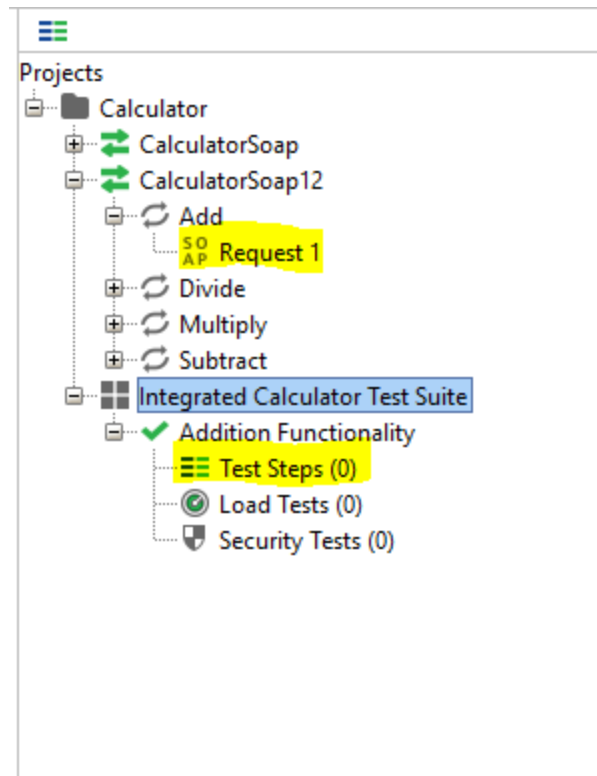


Figure 16 - Expanded Project Tree

3. Add Step with SOAP Request

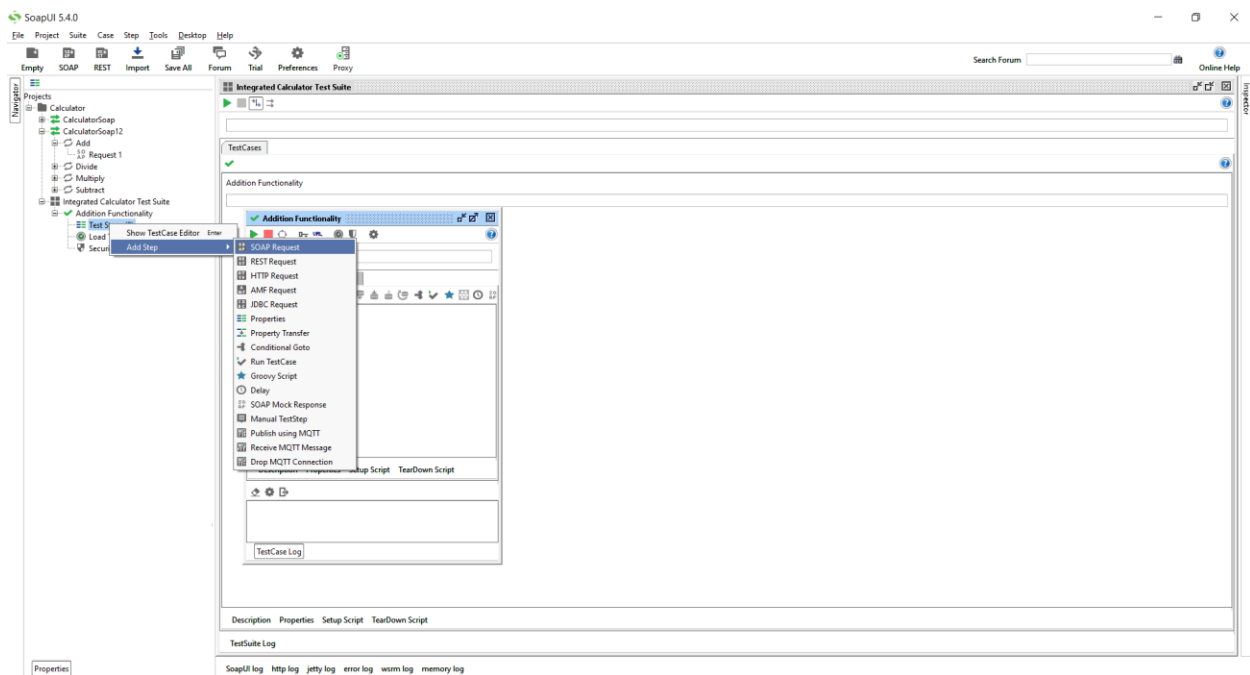


Figure 17 - Add SOAP Request as a Test Step to the test case

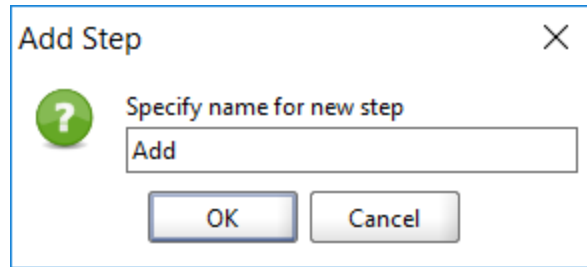


Figure 18 - Add Step

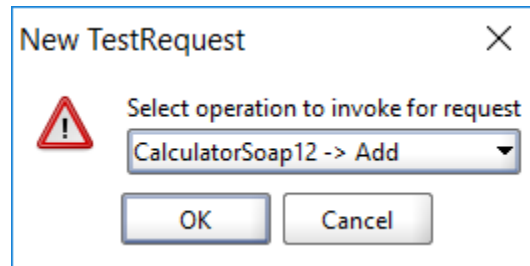


Figure 19 - Select appropriate operation

4. Specify the details for the step by checking appropriate options

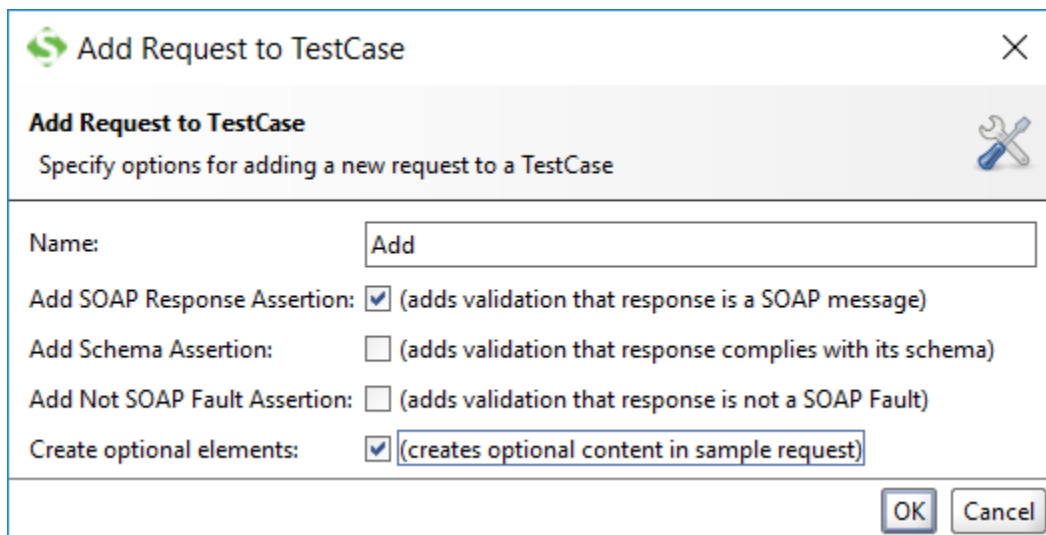


Figure 20 - Details for the test step

Navigator

Projects

- Calculator
 - CalculatorSoap
 - CalculatorSoap12
 - Add
 - Request 1
 - Divide
 - Multiply
 - Subtract
 - Integrated Calculator Test Suite
 - Addition Functionality
 - Test Steps (1)
 - Add
 - Load Tests (0)
 - Security Tests (0)

Integrated Calculator Test Suite

TestCases

Addition Functional

URL: <http://www.dneonline.com/calculator.asmx>

Raw XML

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:tem="http://tempuri.org/">
  <soap:Header/>
  <soap:Body>
    <tem:Add>
      <tem:intA>?</tem:intA>
      <tem:intB>?</tem:intB>
    </tem:Add>
  </soap:Body>
</soap:Envelope>
```

TestSteps

Add

Description

TestRequest Properties

Property	Value
Name	Add
Description	

Custom Properties

Test Case L



Figure 21 - Execution after adding parameters

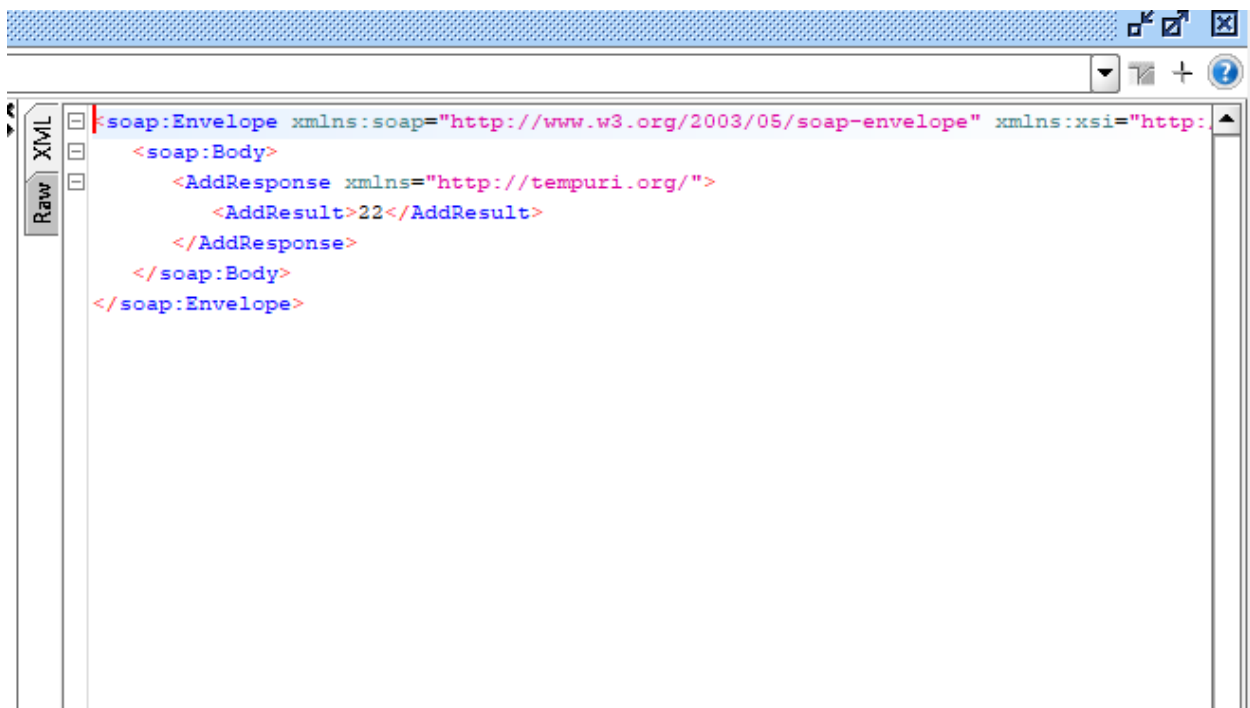
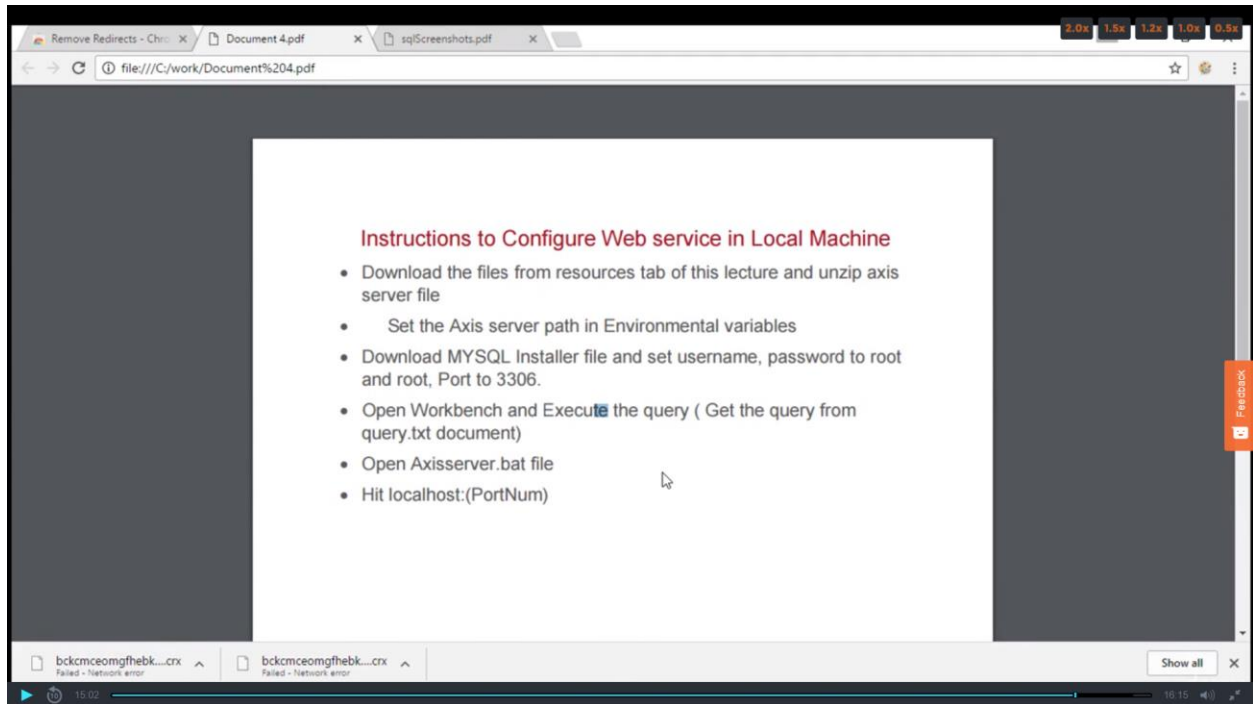


Figure 22 - Response after submitting for addition functionality

Validations can be added to set assertions using the + symbol.

C. Local Web Service Installation

1. Checklist



2. Download the latest version of axis server

<http://www.apache.org/dyn/closer.lua/axis/axis2/java/core/1.7.7/axis2-1.7.7-bin.zip>

Configure the system environment variable

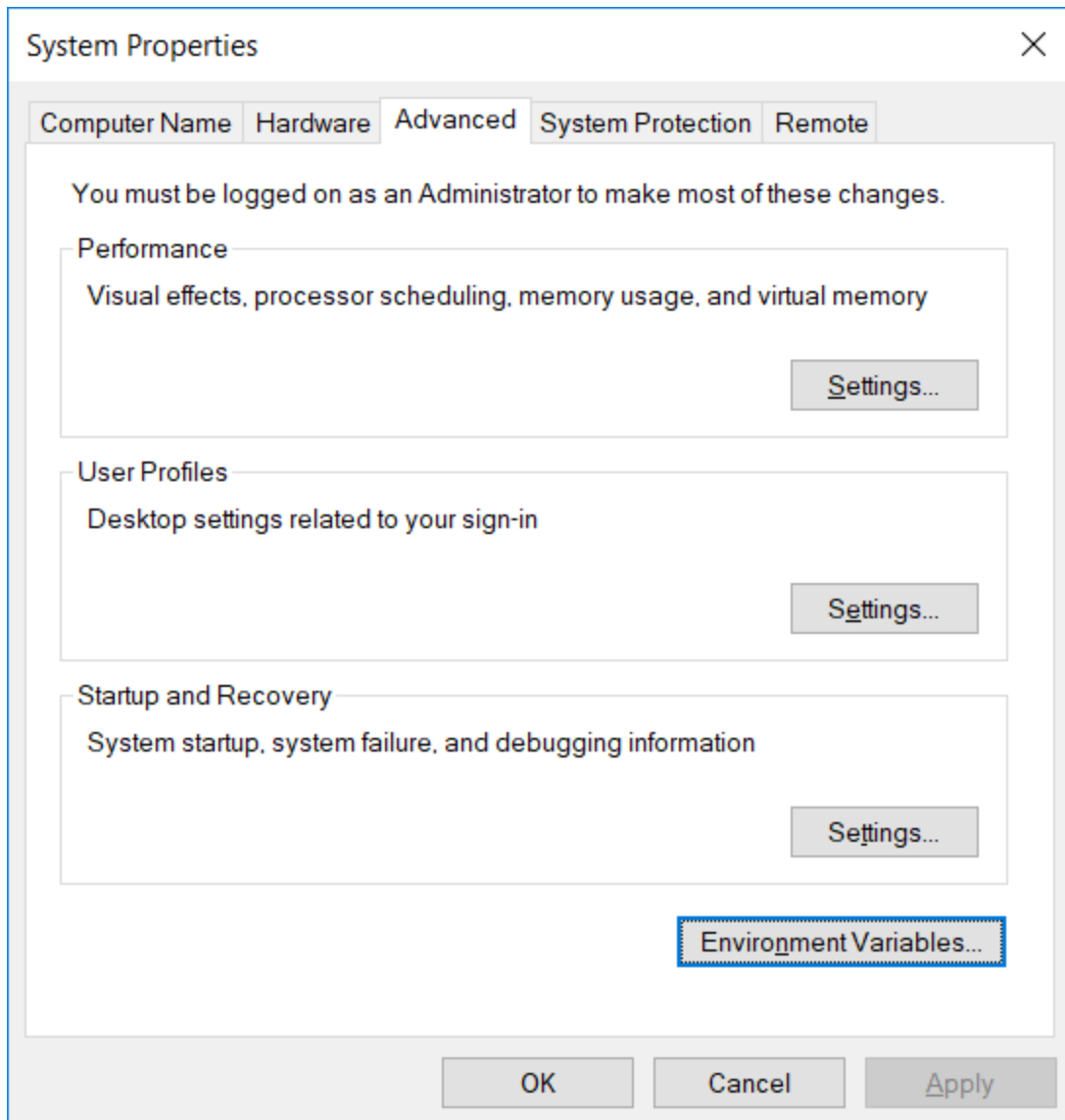


Figure 23 - System properties

3. Create a new system variable named AXIS_HOME

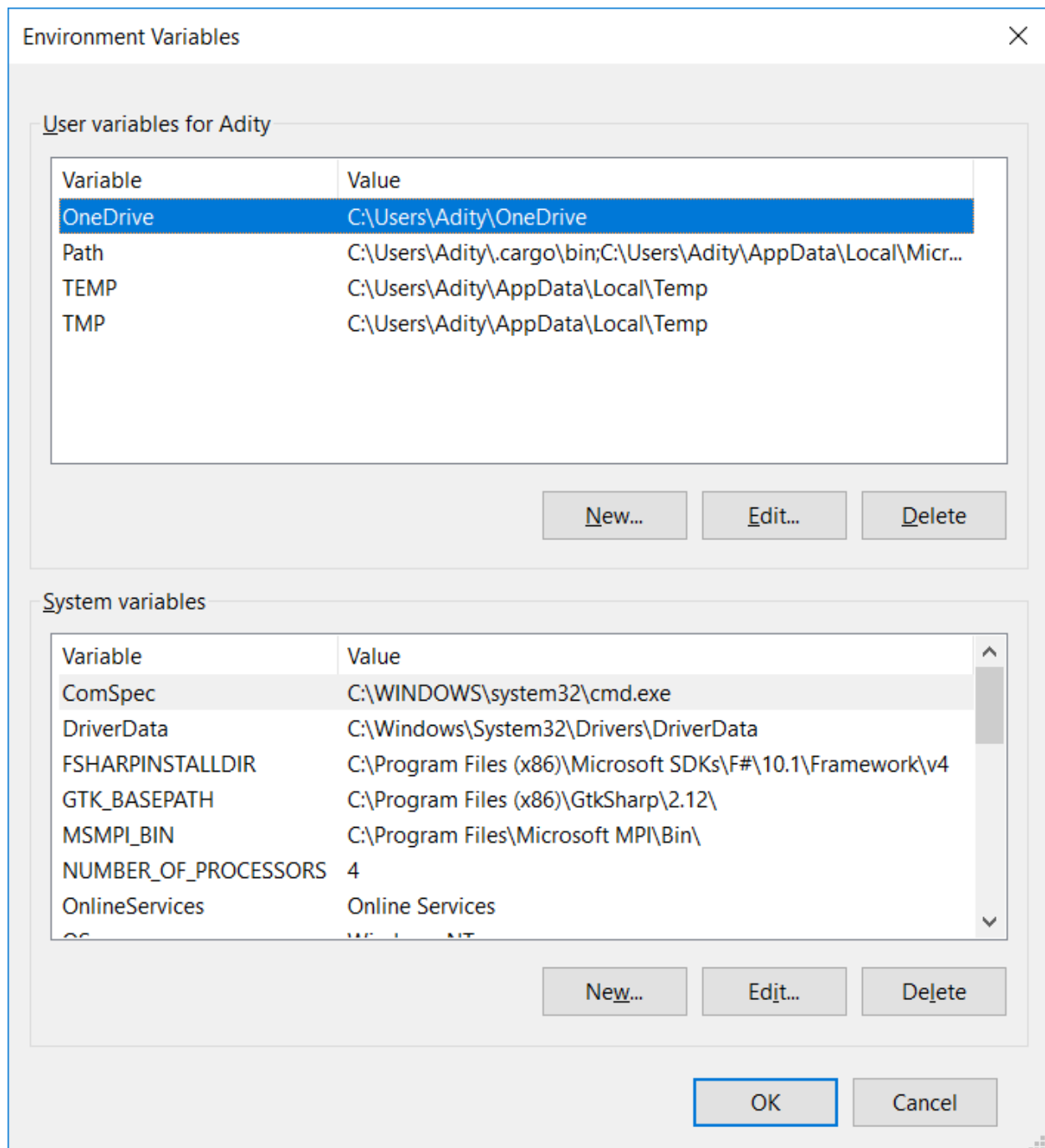


Figure 24 - System variables before adding new variable

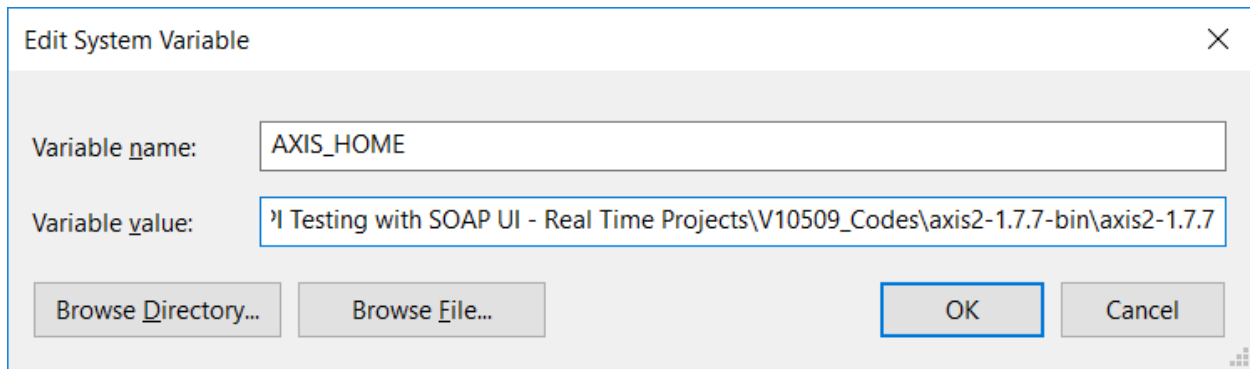


Figure 25 - Configuration of AXIS_HOME system variable

4. Go to bin directory in axis folder

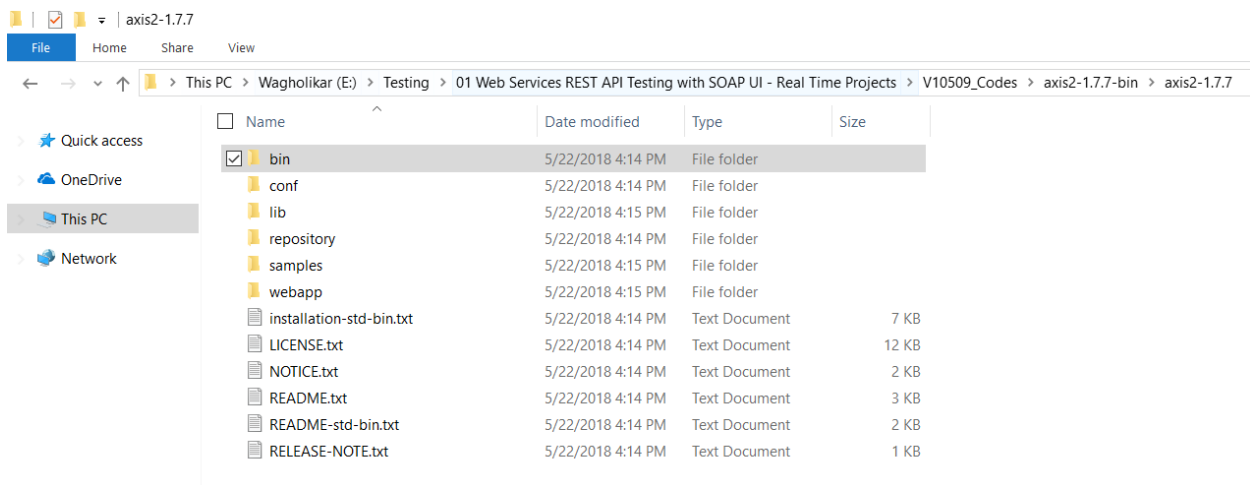


Figure 26 - bin folder in Axis

5. Copy the entire path

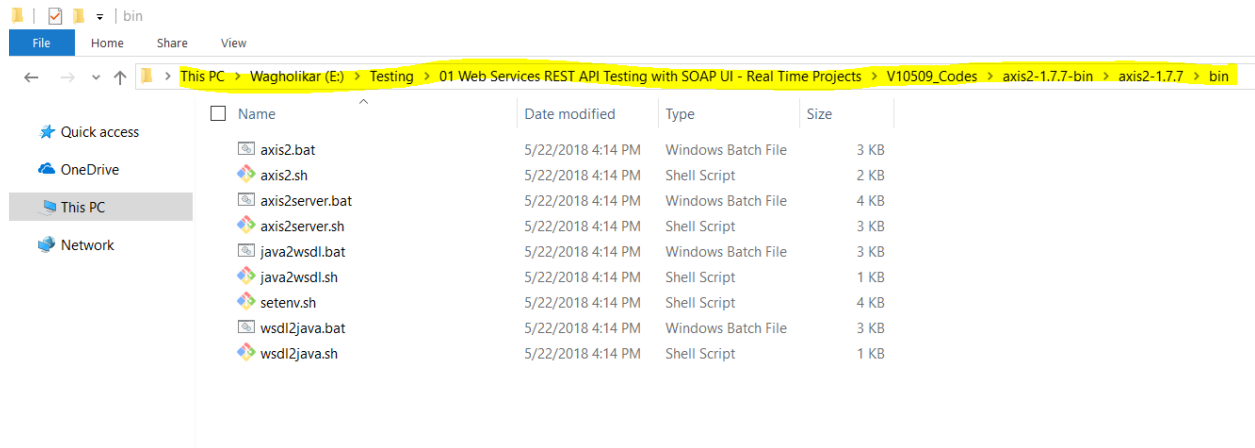


Figure 27 - Inside bin folder with path highlighted

E:\Testing\01 Web Services REST API Testing with SOAP UI - Real Time Projects\V10509_Codes\axis2-1.7.7-bin\axis2-1.7.7\bin

6. Add the path to the System path in System Variables Environment

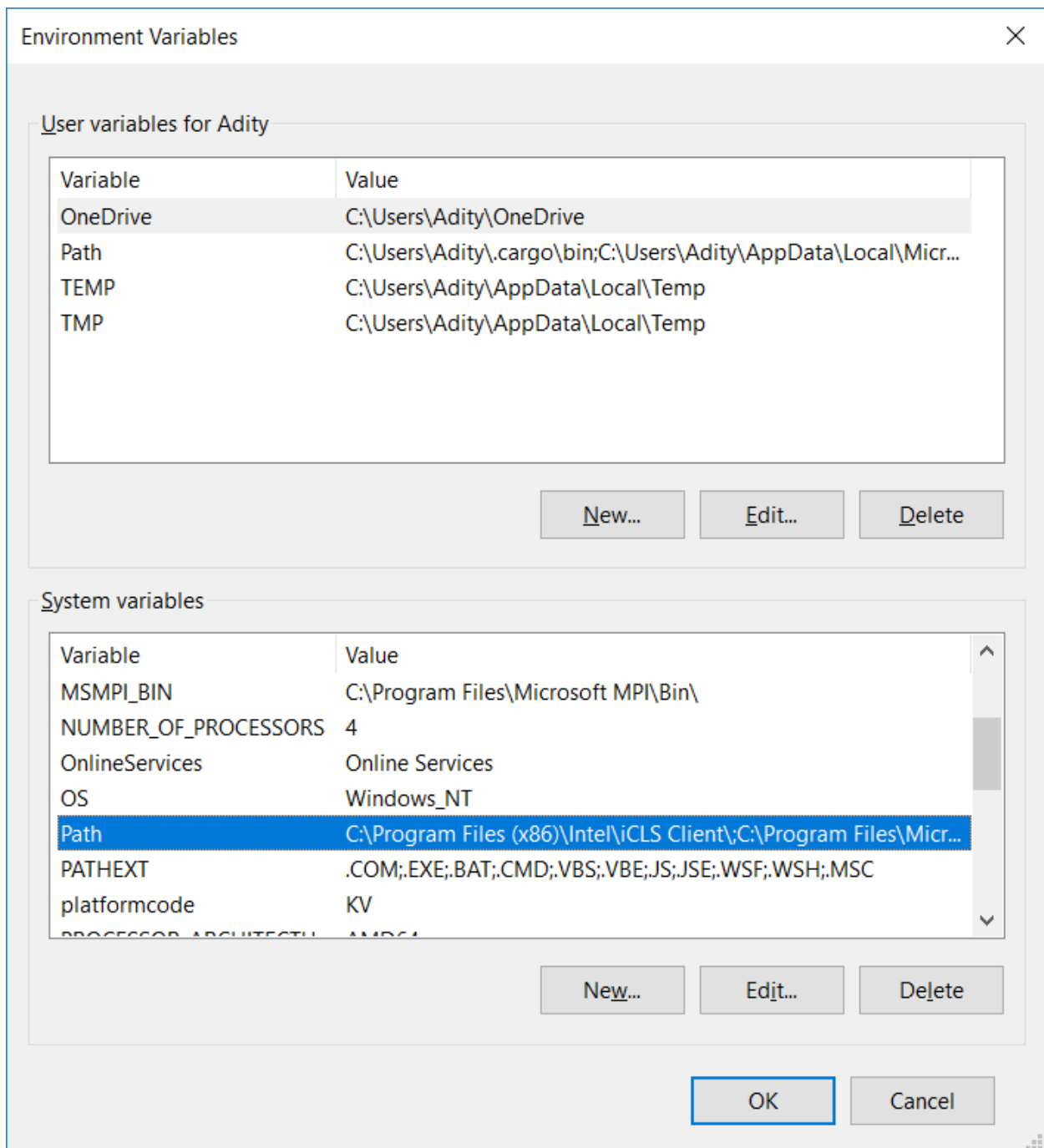


Figure 28 - Path variable to be edited

7. Click New

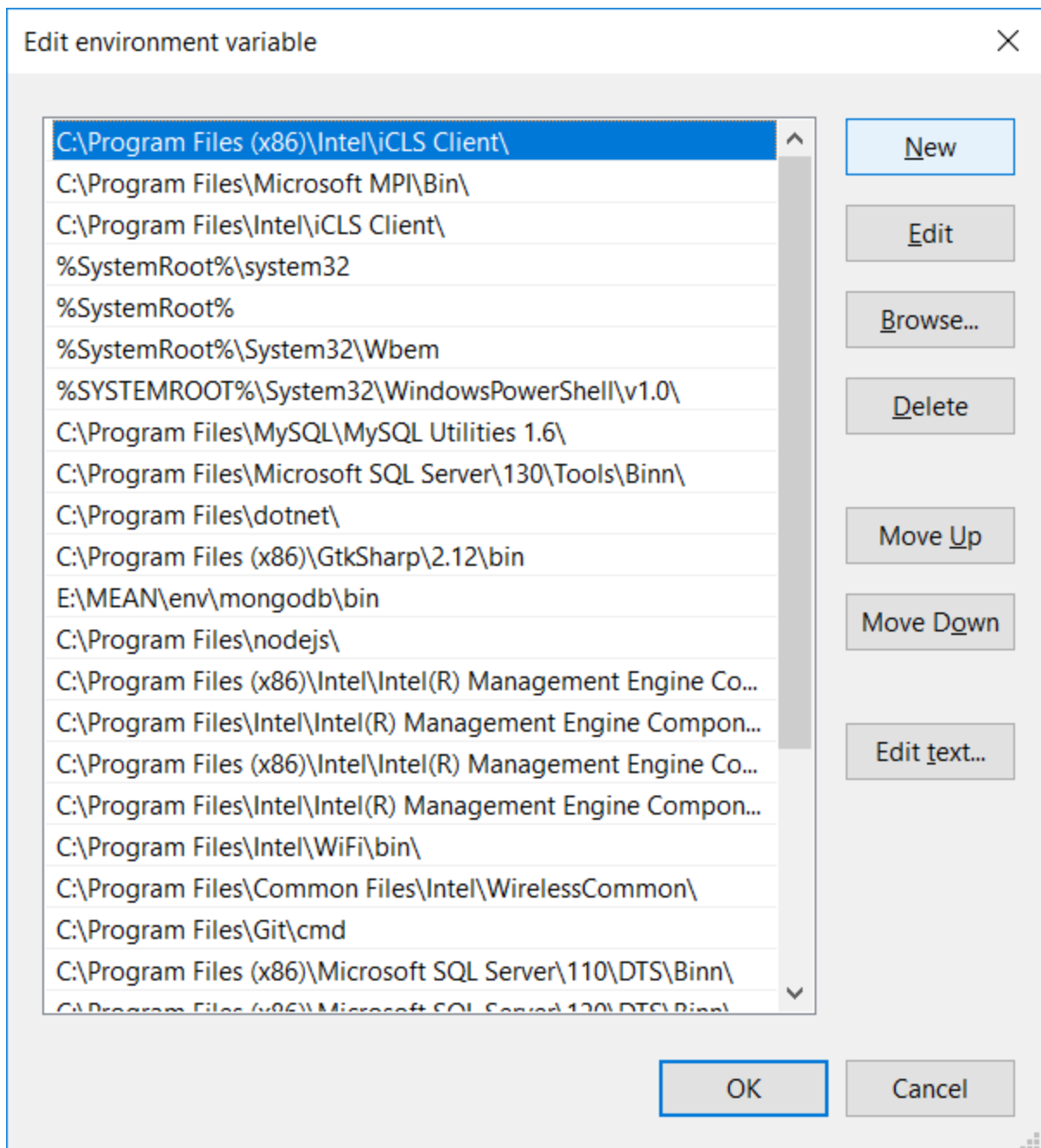


Figure 29 - Various Path in Path Variable

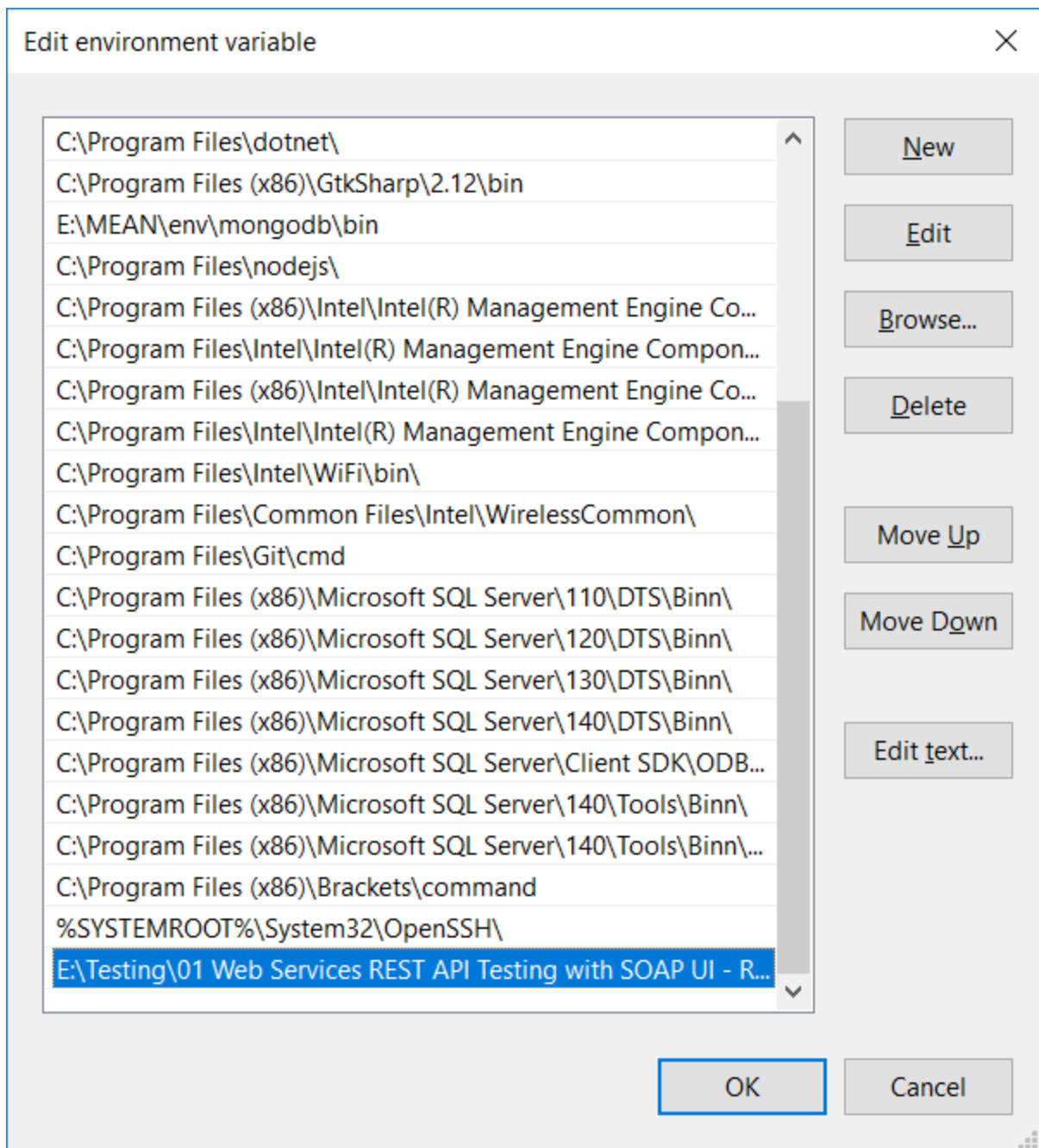


Figure 30 - New system path appended

8. Add JAVA_HOME to the System Variables path if it is missing

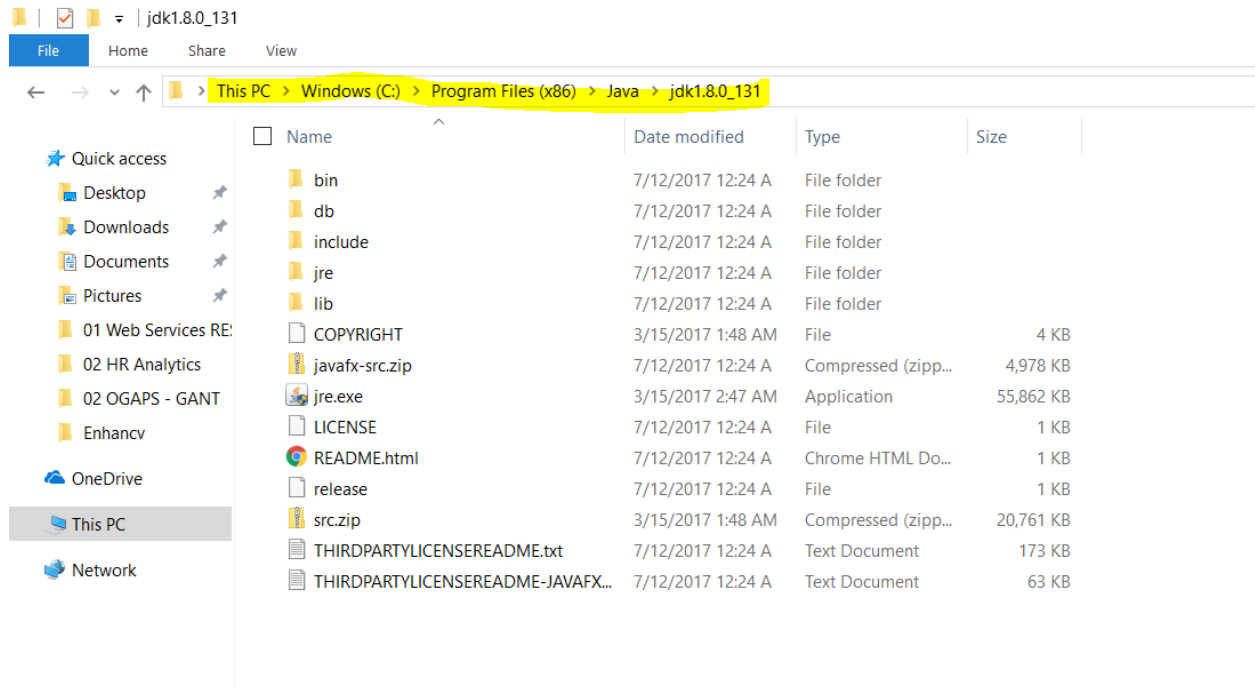


Figure 31 - Java_Home path

In my case the path is -

'C:\Program Files (x86)\Java\jdk1.8.0_131'

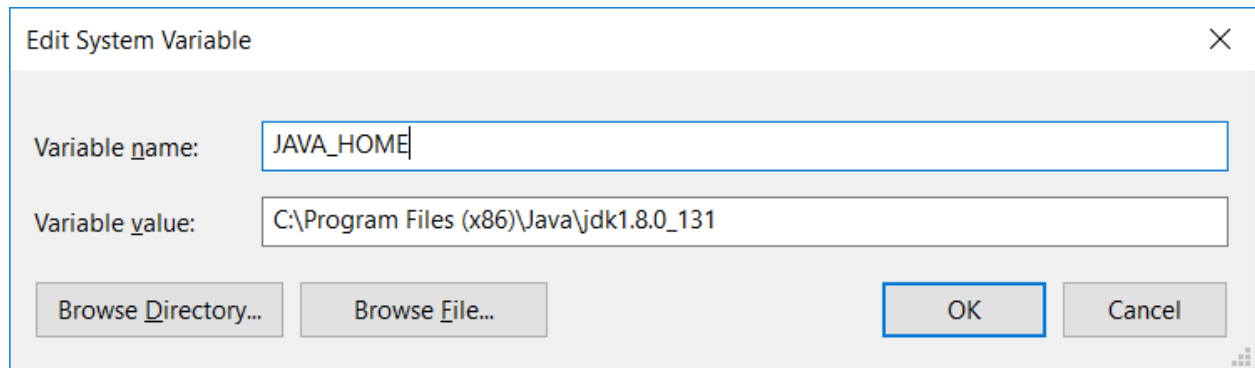


Figure 32 - Create JAVA_HOME system variable

9. Run the query in SQL Workbench

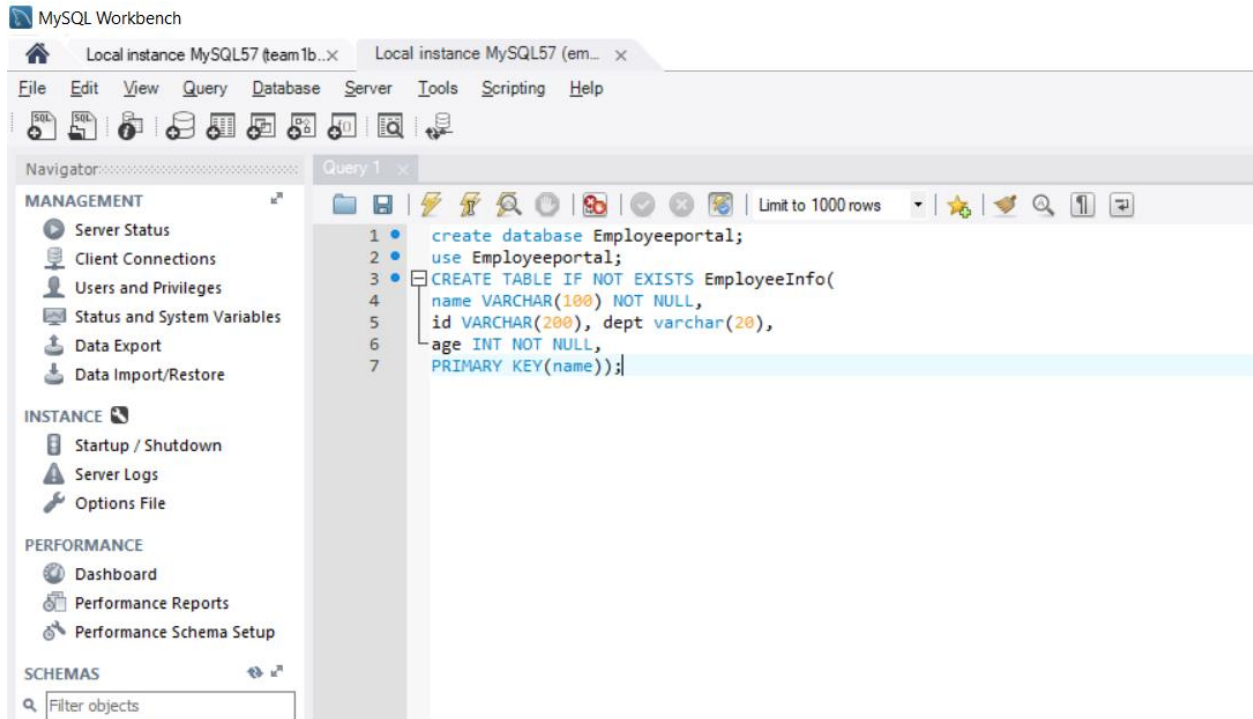


Figure 33 - Query RUN in MYSQL

10. Run the batch file

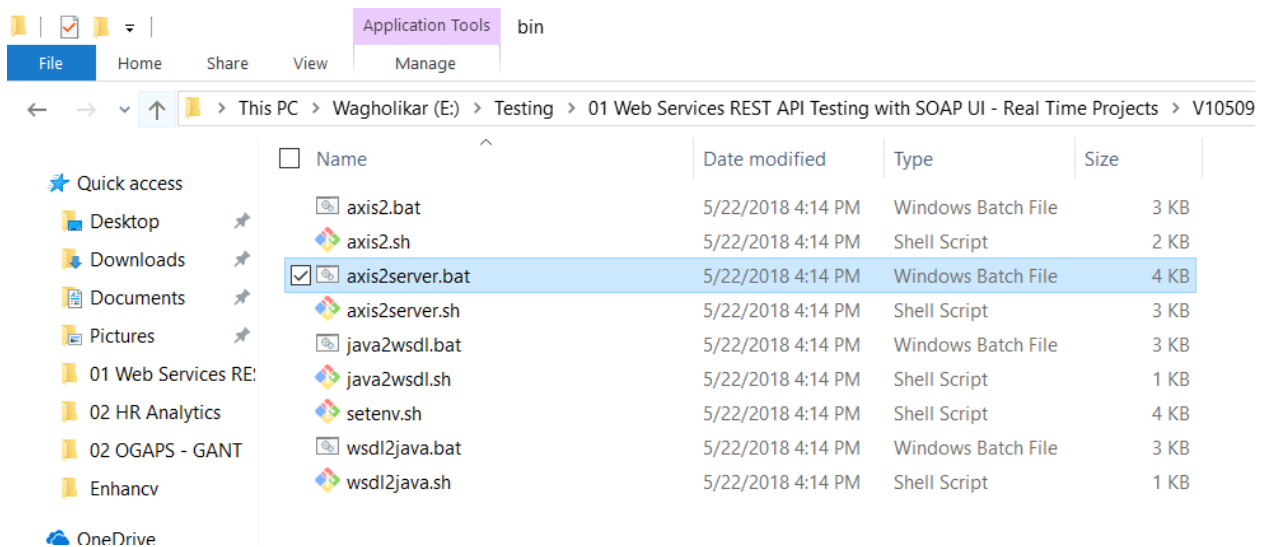


Figure 34 - axis2server batch file to be run

D. Webservices Testing(Manual) with SOAPUI Free Version

11. Assertions

Add Employee Testcase :

Verify Response returning True. – **Contains Assertion**

Verify Response not having null or False Value - **Not contain Assertion**

Verify Response is Valid Soap Response - **Soap Assertion**

Verify Response have Valid HTTP Code - **Http status code assertion**

Verify that Response time is with in boundaries - **Response status**

Verify Security Threat - **Security Assertion**

Verify ID of the Employee is 26 - XPath


Verify Department is CS -XPath

Check SoapUI Response pattern


Verify Age attribute display

Verify only Unique name returns

Figure 35 - Assertions



XPath Match Configuration



Specify xpath expression and expected result

declare namespaces with declare namespace <prefix>=<namespace>;


XPath Expression

Declare

```
declare namespace ax21='http://sql.java/xsd';
declare namespace soapenv='http://schemas.xmlsoap.org/soap/envelope/';
declare namespace ns='http://sample.com/reservation/guest/types';
exists(//ns:age)
```


Expected Result

Select from current Test ☐ Allow Wildcards ☐ Ignore namespace prefixes ☐ Ignore XML Comments




Save

Cancel



XPath Match Configuration



Specify xpath expression and expected result

declare namespaces with declare namespace <prefix>=<namespace>;

XPath Expression


Declare

```
declare namespace ax21='http://sql.java/xsd';
declare namespace soapenv='http://schemas.xmlsoap.org/soap/envelope/';
declare namespace ns='http://sample.com/reservation/guest/types';
count(//ns:name)
```

Expected Result

Select from current Test ☐ Allow Wildcards ☐ Ignore namespace prefixes ☐ Ignore XML Comments

1



Save

Cancel

SOAP Request

http://localhost:8080/axis2/services/EmployeePortalService.EmployeePortalServiceHttpSoap11Endpoint/

Raw XML

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:getEmployeeDetails>
      <!--Optional:-->
      <typ:guestName>sun</typ:guestName>
    </typ:getEmployeeDetails>
  </soapenv:Body>
</soapenv:Envelope>
```

Raw XML

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:getEmployeeDetailsResponse xmlns:ns="http://sample.com/reservation/guest"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="ns:Employee"
      xmlns:ax21="http://sql.java/xsd"
      xmlns:ax22="http://sql.java/xsd">
      <ns:age>26</ns:age>
      <ns:dept>IT</ns:dept>
      <ns:id>888</ns:id>
      <ns:name>sun</ns:name>
    </ns:getEmployeeDetailsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Headers (6) Attachments (0) SSL Info WSS (0) JMS (0)

Online Help

- SOAP Response - VALID
- XPath Match - VALID
- Age xpath - VALID
- Age assertion - VALID
- Department xpath - VALID
- Age node verification - VALID → EXIST
- Unique validation - VALID → COUNT

Assertions (7) Request Log (1)

response time: 127ms (491 bytes) 8:14

Test Case Log

Specify xpath expression and expected result

declare namespaces with declare namespace <prefix>='<namespace>';

XPath Expression

Declare

```
declare namespace ax21='http://sql.java/xsd';  
declare namespace soapenv='http://schemas.xmlsoap.org/soap/envelope/';  
declare namespace ns='http://sample.com/reservation/guest/types';  
//ns:return
```



Expected Result

Select from current Test ☐ Allow Wildcards ☐ Ignore namespace prefixes ☐ Ignore XML Comments

```
<ns:return xsi:type="ns:Employee" xmlns:ax21="http://sql.java/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns="http://  
<ns:age>26</ns:age>  
<ns:dept>IT</ns:dept>  
<ns:id>888</ns:id>  
<ns:name>sun</ns:name>  
</ns:return>
```



01:56

Save

Cancel

Specify xpath expression and expected result

declare namespaces with declare namespace <prefix>=<namespace>;

XPath Expression

Declare

```
declare namespace ax21='http://sql.java/xsd';  
declare namespace soapenv='http://schemas.xmlsoap.org/soap/envelope/';  
declare namespace ns='http://sample.com/reservation/guest/types';  
//ns:return
```

Expected Result

Select from current Test ☒ Allow Wildcards ☐ Ignore namespace prefixes ☐ Ignore XML Comments

```
<ns:return xsi:type="ns:Employee" xmlns:ax21="http://sql.java/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns="http://sample.com/reservation/guest/types">  
  <ns:age>26</ns:age>  
  <ns:dept>IT</ns:dept>  
  <ns:id>1</ns:id>  
  <ns:name>sun</ns:name>  
</ns:return>
```

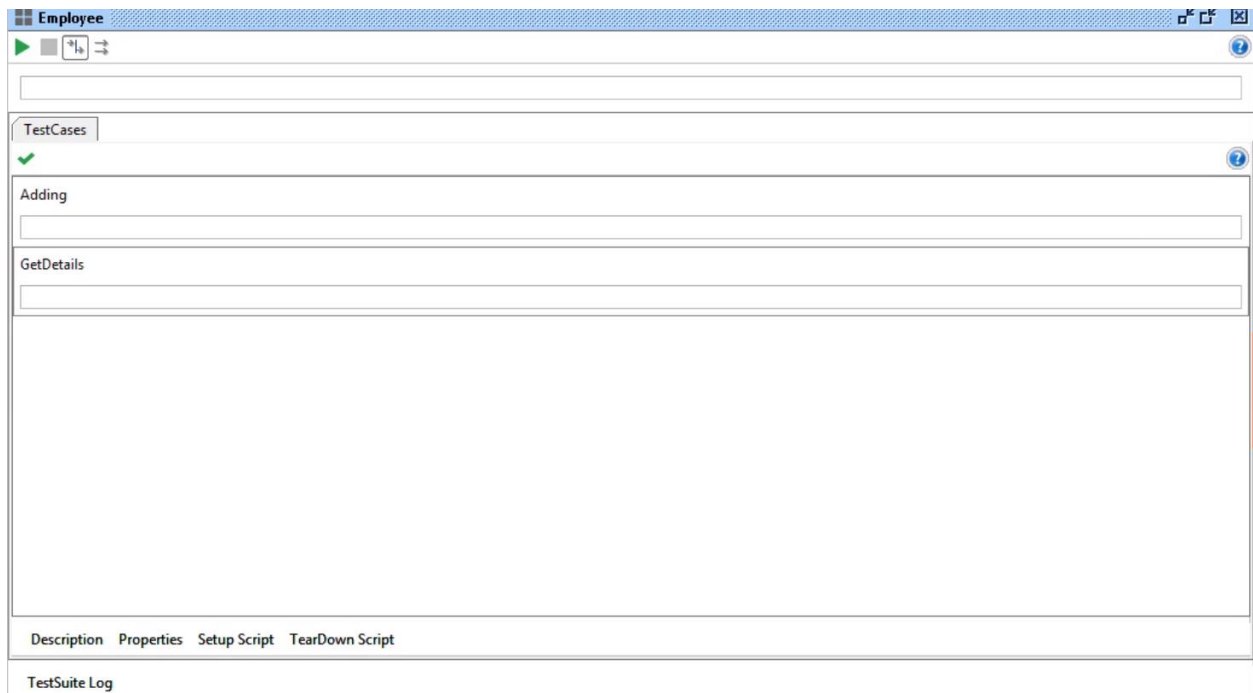


Figure 36- Test Suite



Figure 37 - Test Suite Execution Result

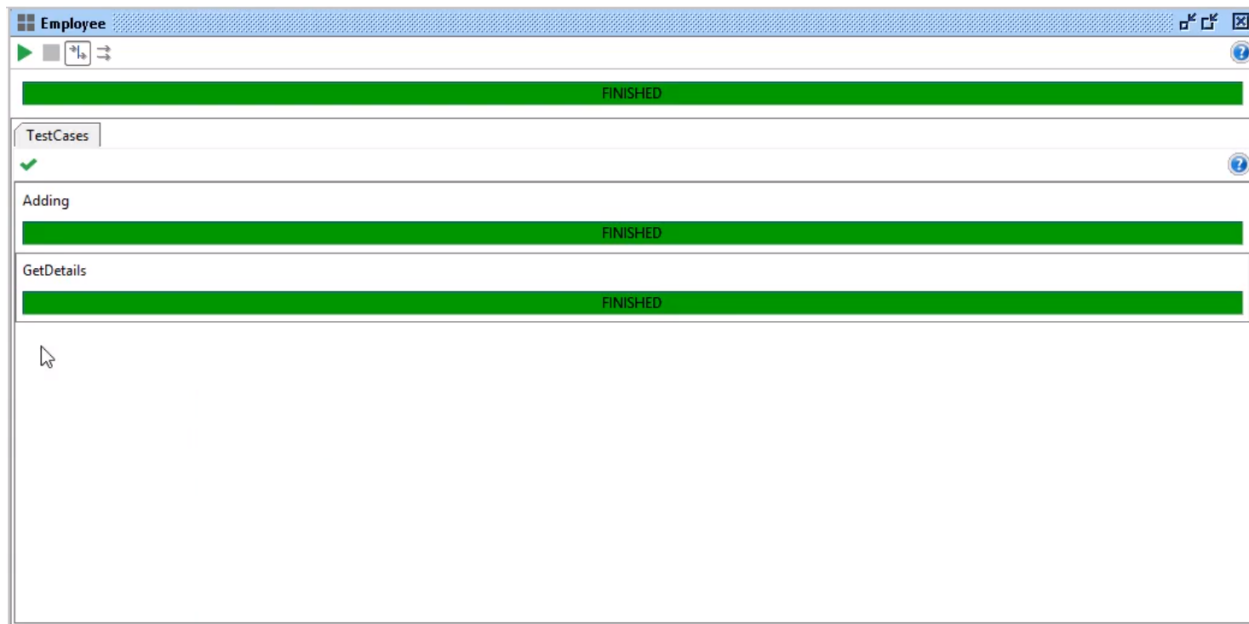


Figure 38 - Successful Test Suite Execution

E. Properties Access-Data driven Strategy

Properties in SoapUI

1. TestCase level Properties Access
2. TestSuite level Properties Access
3. Project level Properties Access
4. Properties TestStep
5. Properties upload from external file
6. Property Transfer testStep
7. Accessing Custom properties.

Figure 39 - Properties in SOAPUI

