

## Experiment 4

**AIM:** SOAP API testing using

SoapUI.

**Theory:** Before we create a SOAPUI Test case, let us understand basics about the SOAP Protocol. This will help you use SOAP UI to test SOAP requests and response effectively.

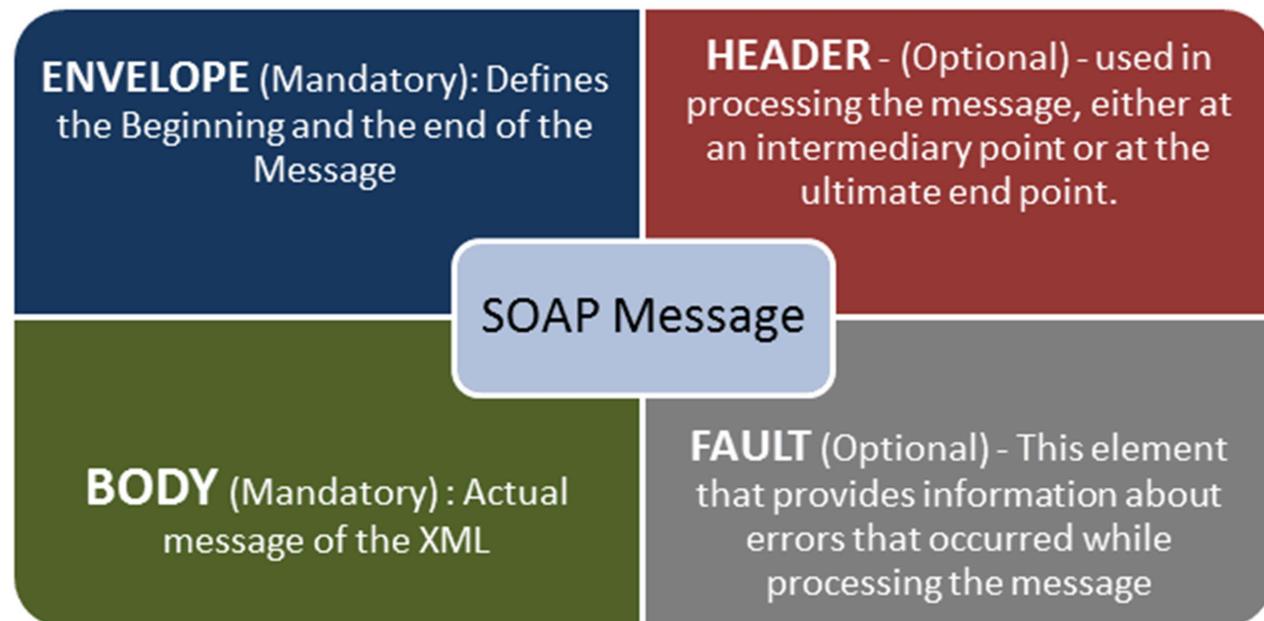
SOAP stands for Simple Object Access Protocol. Below are the properties of a SOAP Protocol.

- It is an XML-based protocol for communicating between two different systems.
- It is a platform and language independent. Hence, a system developed using Java can communicate with a system developed in .NET.
- SOAP requests/response is transported via HTTP.

**Requirement Analysis:** - Google Chrome, SOAPUI, Online tool

**Hardware Requirement:** - Computer, Windows Power Supply.

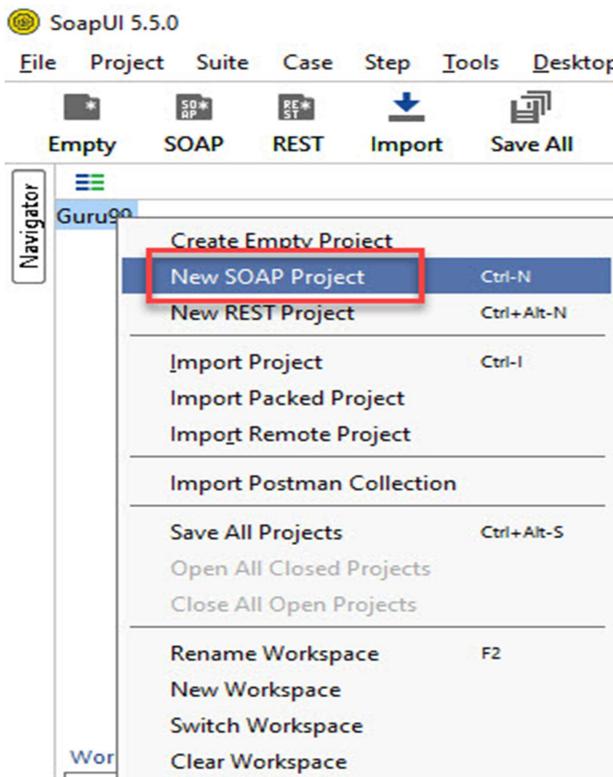
SOAP message is an ordinary XML document containing the following elements. Message can be either a request message or a response message.



After setting up the workspace which we had performed in the last tutorial, we have to create SoapUI project structure, test suites, test cases in order to test a given web service. Let us understand a SoapUI Project example to create a new SOAP project.

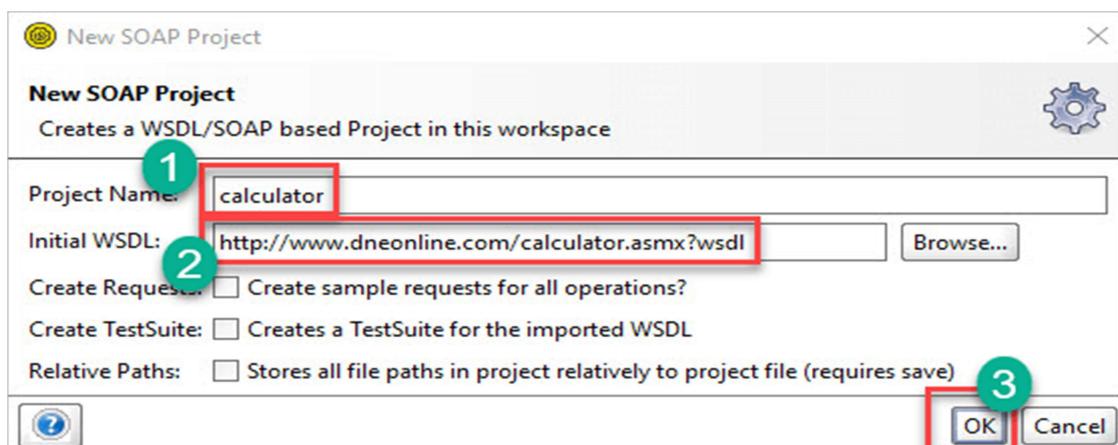
## Creating a SOAP Project

**Step 1)** Now, depending upon the project, we need to import SOAP/REST protocol. We will create a new SOAP Project



**Step 2)** We will make use following SOAP request <http://www.dneonline.com/calculator.asmx?wsdl>

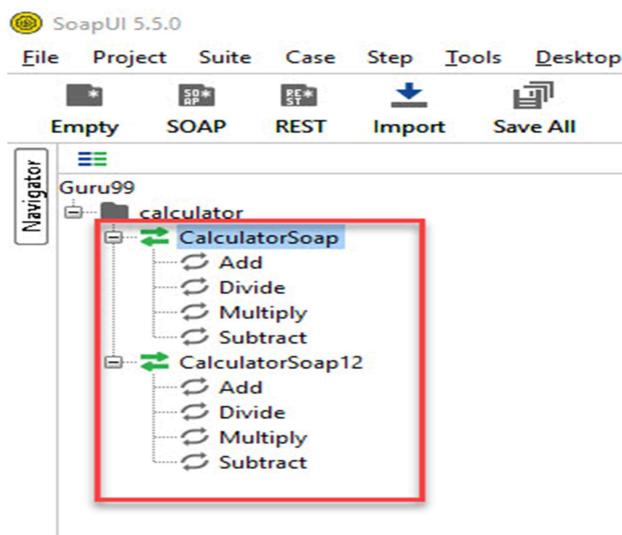
- Enter the Project Name
- Enter the path of the WSDL request. In this case <http://www.dneonline.com/calculator.asmx?wsdl>
- Click OK



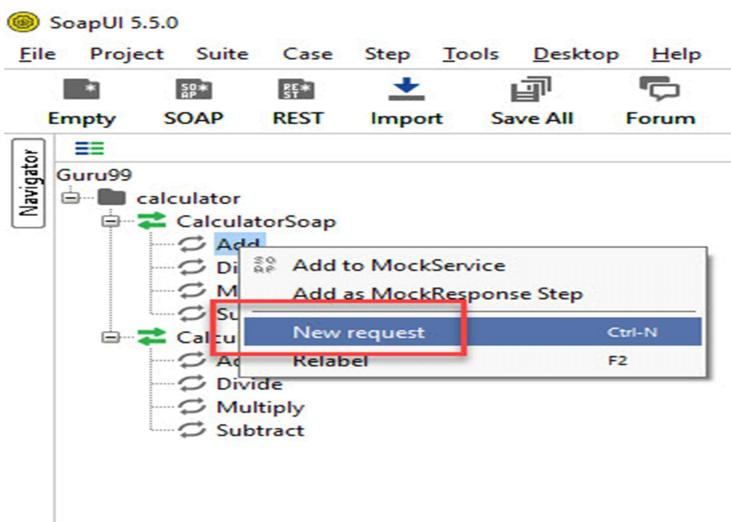
**Note:**

- Create sample request for all operations? It creates a sample request for all the available operations in the given WSDL. As soon as you enter the WSDL address, this option is checked automatically. You may uncheck it.
- Create a Test Suite for the imported WSDL: Creates a SoapUI test suite within the project for the imported WSDL.
- Relative Paths: It enables the user to save all the files relative to the project file.

**Step 3)** Upon creating the SOAP project with the above-said WSDL, we will be able to see that there are two operations that will be imported into the project.



**Step 4)** Expand the first request and right-click on the ‘Add.’ Then click on ‘New Request’.



Then Click on ‘OK’. It will display the SOAP request in the XML format



1. Enter the ‘intA’ and ‘intB’
2. Click on the submit button
3. Response XML will be displayed right side pane.

You may wonder why create Test Cases? When you can directly test Webservice here...

Well, you can send a request for one operation. What about others? How many combinations of inputs for Additions can you do using this operation? You have to edit the request for each and every combination.

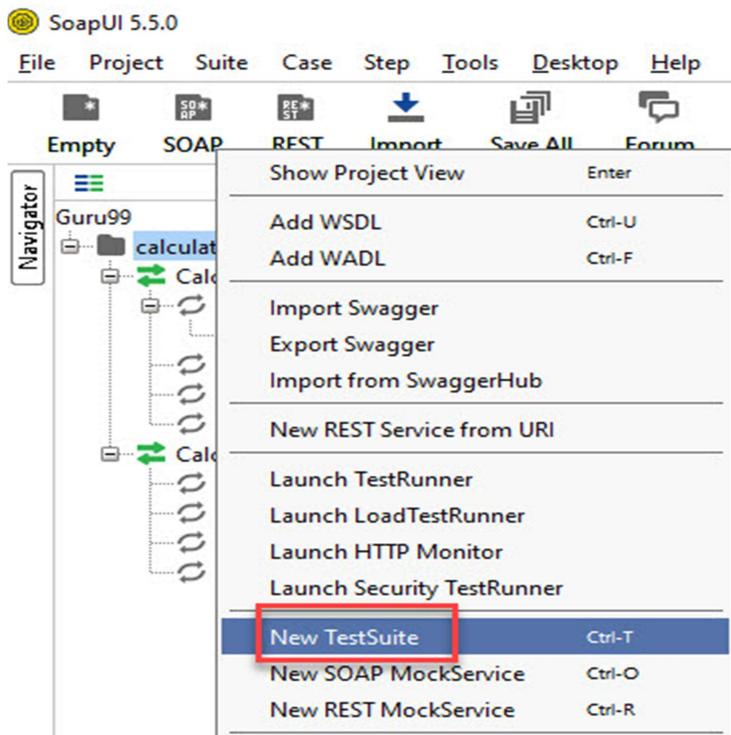
For example: If you want to add from 4 and 4 instead of 5 and 5... You need to edit the operation again. So, one has to create a test suite/cases to have all possible scenarios tested without having to directly edit the operation itself.

# How to Create Test Suite in SoapUI

Below are the steps to create Test Suite in SoapUI:

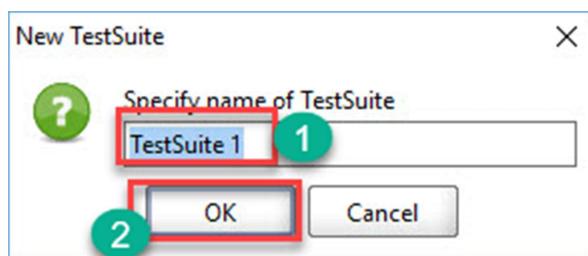
## Step 1) Right-click on the root of the project

Within the project, testers can create a test suite by performing a right-click on the root of the project.



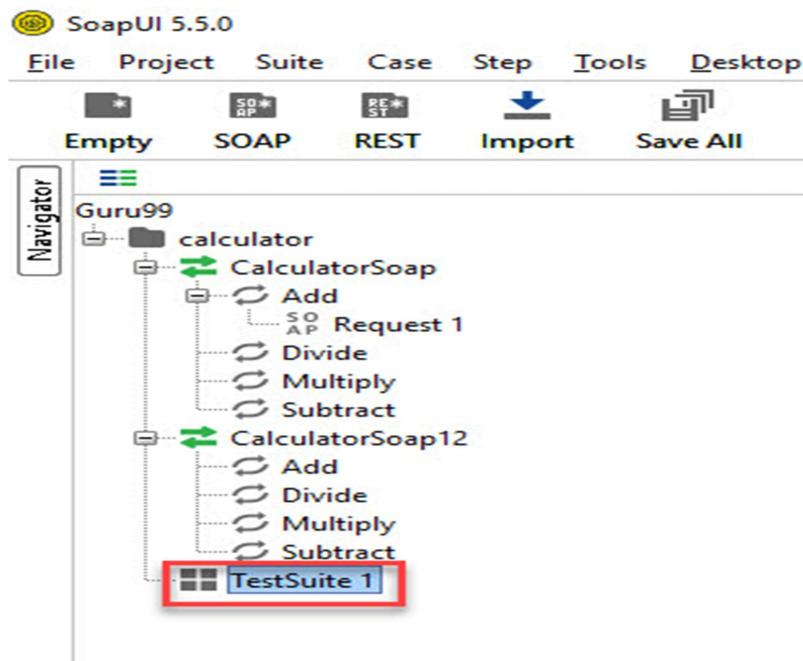
## Step 2) Enter the Test suite details

We need to enter the name of the test suite and press OK.



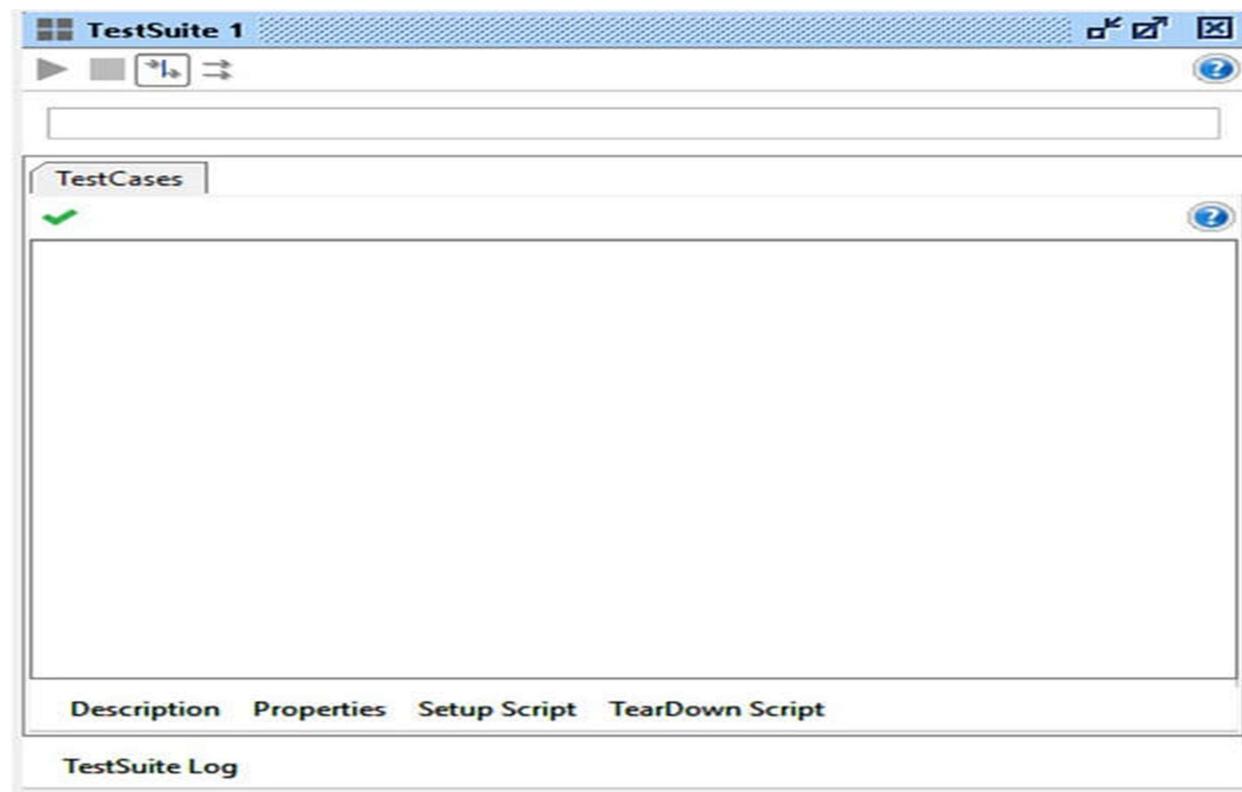
## Step 3) Verify the created Test suite

The created test suite is displayed the navigator pane as shown below.



#### Step 4) Open the Test suite

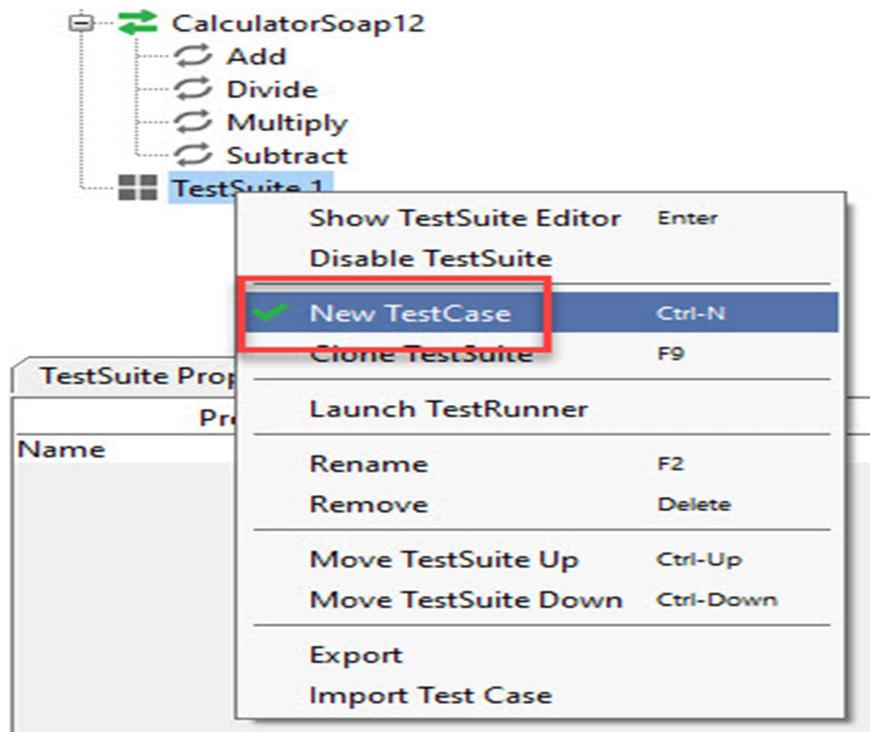
The test Suite window opens in the Right Pane. As we have just created there are NO SoapUI test cases. Hence all options are disabled.



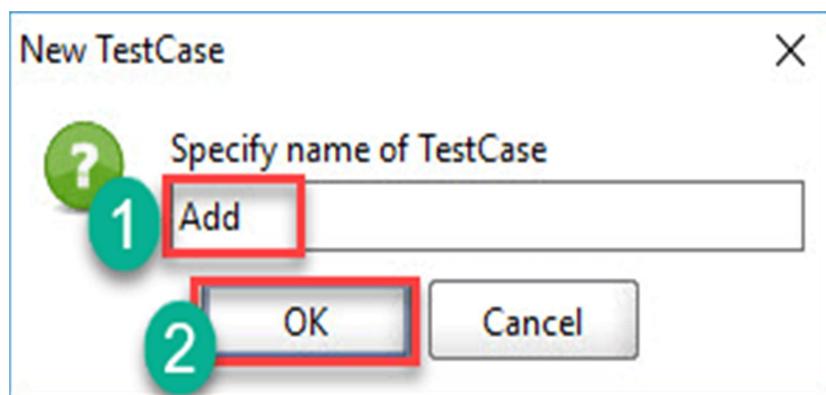
## How to Create Test Case in SoapUI

Here is a step by step process for creating a test case in SoapUI:

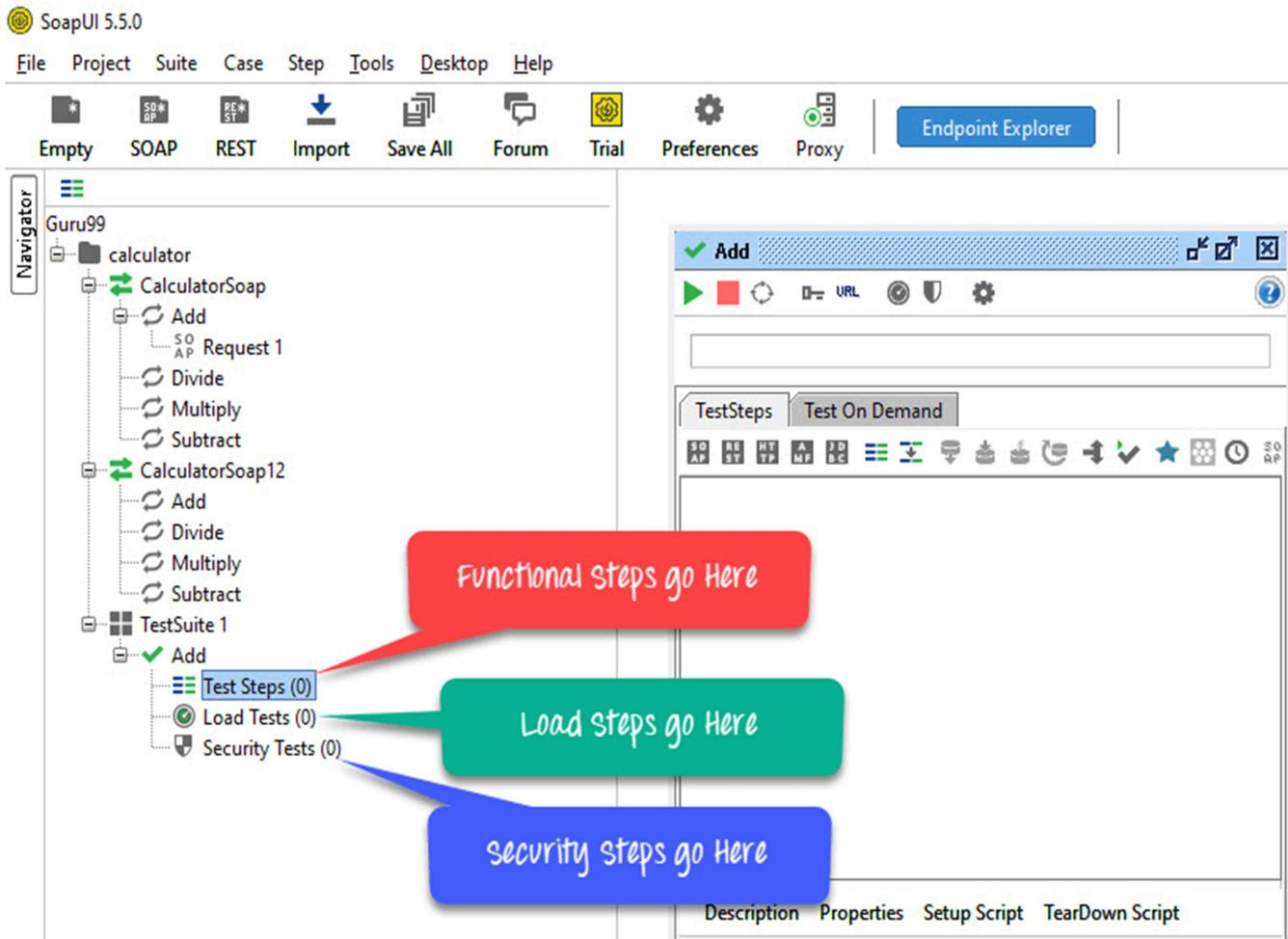
**Step 1)** Within a test suite, we can create multiple tests by performing right click on the ‘test suite’ and choosing ‘New Test Case’.



**Step 2)** Specify the name of the Test Case and click ‘OK’.



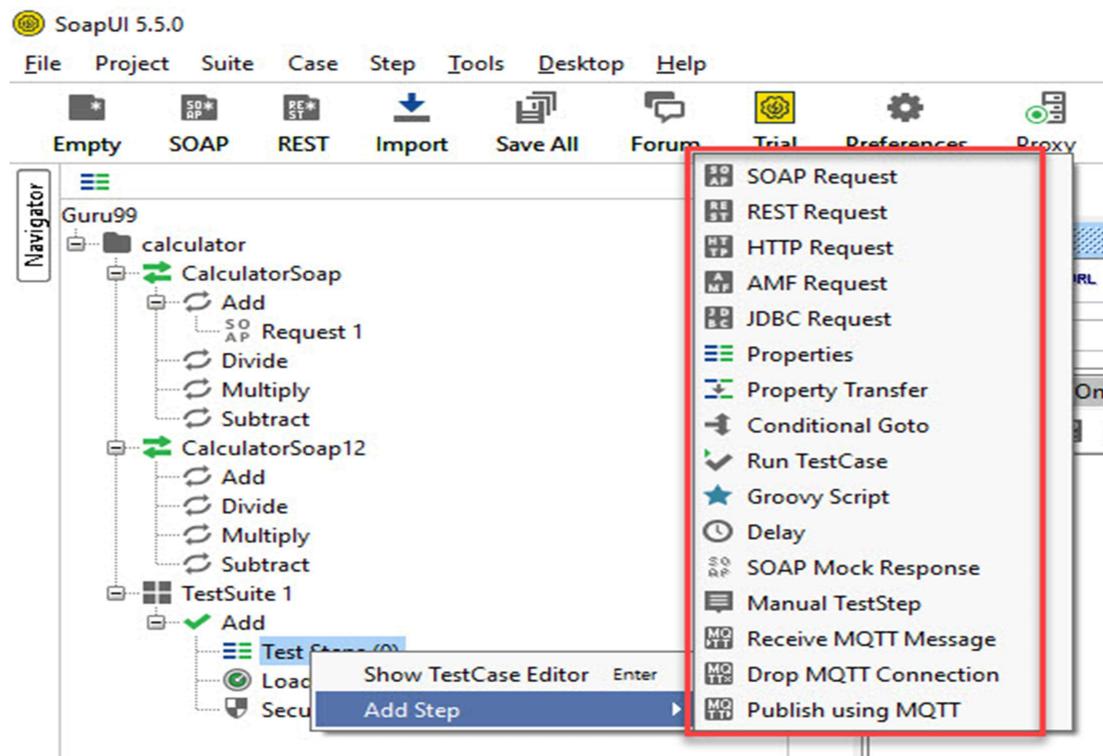
**Step 3)** The created test case has zero steps as shown below.



**Note:** We can see that the test case is added with zero test steps for all kinds of tests available. Upon adding the test steps, the numbers in the bracket would change automatically.

The functional test step should go into ‘Test Steps’ while a performance test step should go into ‘Load Test’ and a security test step should go into ‘security Tests’.

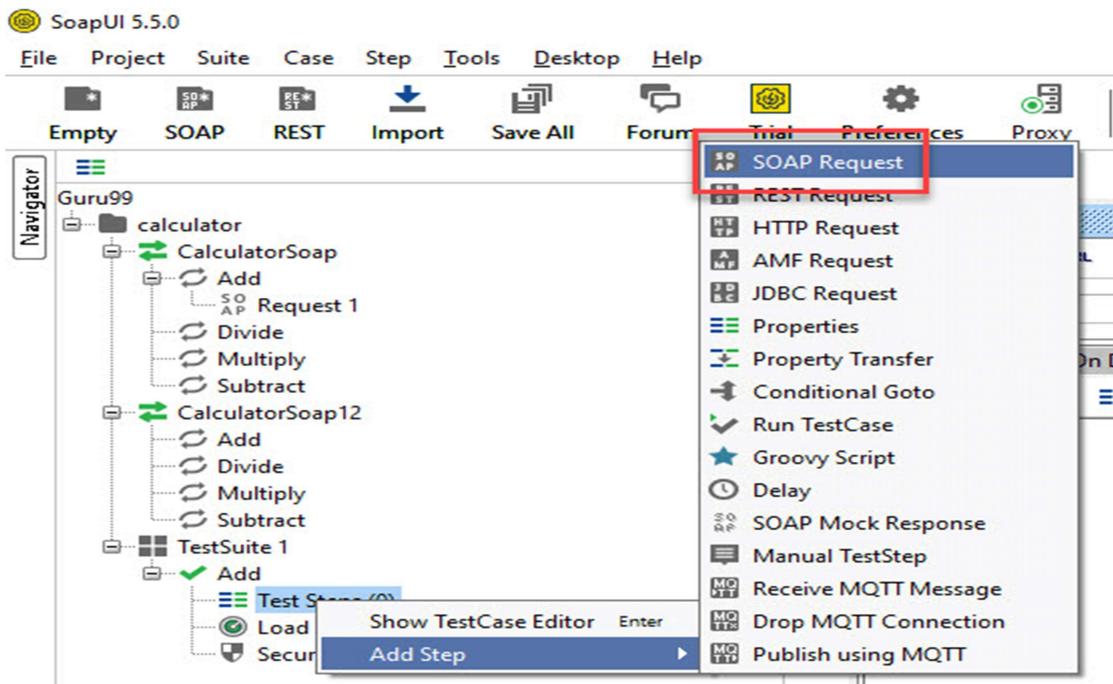
**Step 4)** We can insert a variety of test steps by performing a right-click on test steps and selecting an appropriate test step as shown below. So, if you were to test a REST Webservice, you would select the REST Test Request.



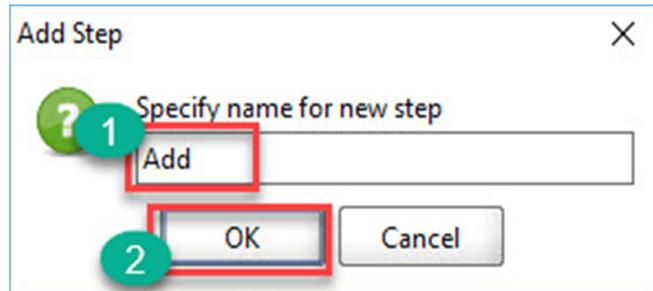
### Adding a Test Step in SoapUI

Now let us add a test step to validate the imported SOAP testing request:

**Step 1)** Add a new step ‘SOAP Request’ as shown below.

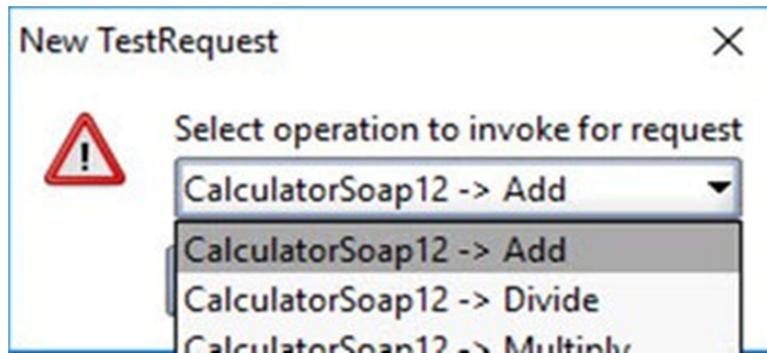


**Step 2)** Enter the step name and click OK.

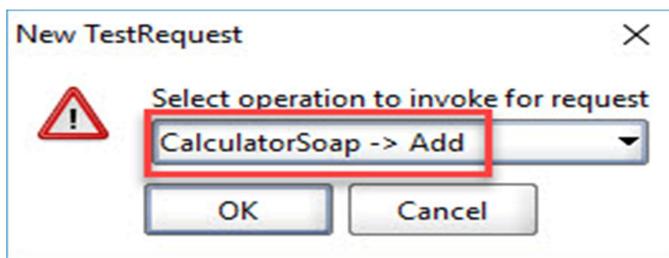


**Step 3)** Upon clicking ‘OK’, a dialog pops up to select the operation to invoke. All the operations are listed, and user can select the operation that they would like to invoke.

- There are many operations that will be listed. The Operations are the same except the SOAP version used. CalculatorSoap – uses SOAP version 1.1 whereas, CalculatorSoap12 – uses SOAP version 1.2
- The Version does not matter for us in this context. Hence you can select the one of your choice.

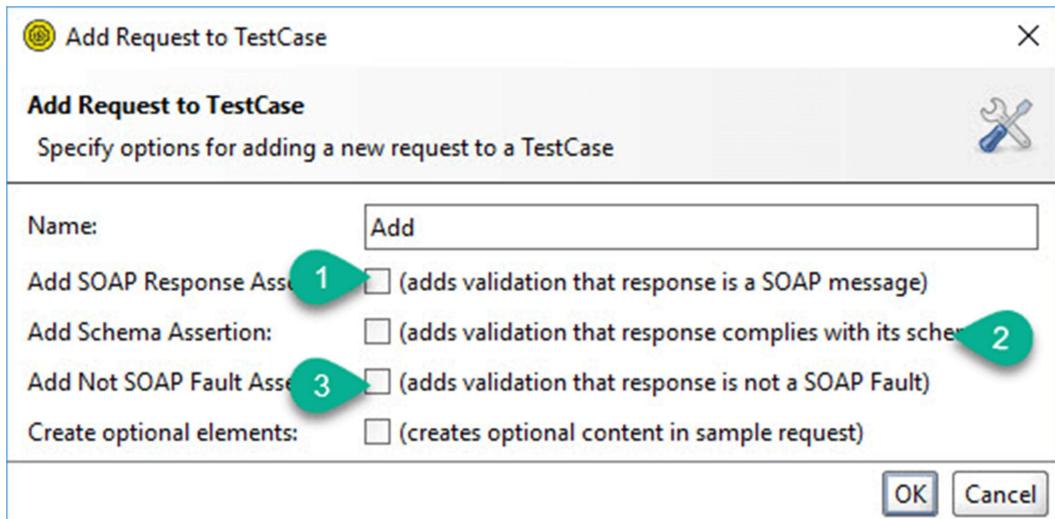


- Upon Selecting the operation, click ‘Ok’



**Step 4)** While adding a test case, we can add standard assertions. Assertions also called as checkpoints/validation points which we will be dealing in detail in the next tutorial.

We can add following checkpoints/assertions while creating test case. Let us create a test case with the option which means creating test step WITHOUT any of the below validation points

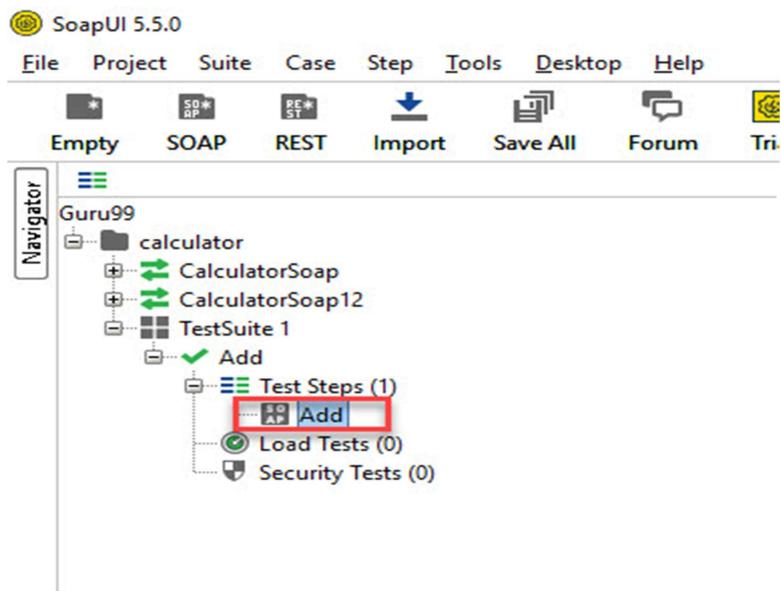


1. Verifies if the response message is SOAP, upon executing the test.
2. Verifies if the response schema is valid.
3. Verifies if the SOAP response contains FAULT.

**Step 5)** Upon creating the test case, the request XML is shown below. The structure of the XML is explained within the below snapshot.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<tem:Add>
<tem:intA>?</tem:intA>
<tem:intB>?</tem:intB>
</tem:Add>
</soapenv:Body>
</soapenv:Envelope>
```

**Step 6)** The test step count is now incremented to one as we have just added one test step. Similarly, upon adding load and security tests step, the corresponding number would be automatically incremented based on the number of steps added.



## Sending Request Manually & Reading Response in SoapUI

**Step 1)** We would like to add two Integer Number.

- intA – 5
- intB – 5

Next,

1. We need to enter these inputs in place of the question mark which will be sent as request XML.
2. After inputting those values into the corresponding XML tags, click ‘submit request’ button to check the response.

This screenshot shows the SoapUI Request editor. The title bar says 'Add'. The URL field contains 'http://www.dneonline.com/calculator.asmx'. On the left, there are tabs for 'Raw' and 'XML'. The 'XML' tab is selected and shows an XML envelope. The code is:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/CalculatorService">5</tem:intA>5</tem:intB>
```

Annotations are present: a green circle with the number '2' is at the top left of the XML tab. A red rectangle highlights the entire XML code area. A green circle with the number '1' is at the bottom center of the XML code area, pointing to the value '5' in the 'intB' tag.

**Step 2)** Upon submitting a request the web service request is processed by the webserver and sends back a response as shown below.

By reading the response, we are able to conclude 5 plus 5 is 10.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header/>
    <soapenv:Body>
        <tem:Add>
            <tem:intA>5</tem:intA>
            <tem:intB>5</tem:intB>
        </tem:Add>
    </soapenv:Body>
</soapenv:Envelope>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
    <soap:Body>
        <AddResponse xmlns="http://tempuri.org/">
            <AddResult>10</AddResult>
        </AddResponse>
    </soap:Body>
</soap:Envelope>
```

### Understanding the Soap Response & Log Panels

As explained at the beginning of this SoapUI testing tutorial, the SOAP messages are transported via protocol. Let us take a look at the RAW messages. This will help us learn how the SOAP request and response were transported by HTTP.

**Step 1)** Click ‘RAW’ Tab in both SOAP-UI request Window.

1. The Request is posted to the webserver. Hence, the POST method of Http is used.
2. The SOAP Request is transported in the body of the Http message.

```
POST http://www.dneonline.com/calculator.asmx HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: "http://tempuri.org/Add"
Content-Length: 291
Host: www.dneonline.com
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

The actual content of the last submitted request

**Step 2)** Now click ‘RAW’ Tab in SOAP-UI Response Window to understand how the response is sent via HTTP.

1. After processing the request, the Http response code (200) is shown which means it is a success. The webserver has processed it successfully.
2. The SOAP response is sent back to the client as part of the body of the HTTP message.

The screenshot shows the SOAP-UI interface with the 'Raw' tab selected. The status bar at the top displays 'HTTP/1.1 200 OK'. The main pane shows the raw HTTP response headers and the XML response body. A red callout box labeled 'HTTP Response' points to the status bar. Another red callout box labeled 'SOAP Response in body of HTTP' points to the XML content in the body of the response.

```
HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Content-Encoding: gzip
Vary: Accept-Encoding
Server: Microsoft-IIS/7.5
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Wed, 04 Sep 2019 12:47:35 GMT
Content-Length: 298

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

A Quick snapshot of the Http Response codes for easy understanding and debugging. The below table

Http Code	Description
1xx:	<b>Informational</b> – This means a request received and continuing process.
2xx:	<b>Success</b> – The action was successfully received, understood, and accepted.
3xx:	<b>Redirection</b> – This means further action must be taken in order to complete the request.
4xx:	<b>Client Error</b> – This means the request contains bad syntax or cannot be fulfilled
5xx:	<b>Server Error</b> – The server failed to fulfil an apparently valid request

will help you to trouble shoot based on the HTTP code received from the webserver.

**Step 3)** Let us understand the other information that are displayed in the test case window.

The screenshot shows the SoapUI interface with the following components:

- Request pane (Left):** Displays the SOAP request XML. It includes sections for Header and Body. The Body contains an `<tem:Add>` block with two integer parameters, `intA` and `intB`, both set to 5.
- Response pane (Right):** Displays the SOAP response XML. It includes a `<AddResponse>` block containing a `<AddResult>10</AddResult>`.
- Headers pane (Bottom Left):** A table showing request headers. One row is highlighted with a red border, indicating the selected header.
- Attachments pane (Bottom Right):** A table showing response attachments. It shows 10 headers and 0 attachments.

Four numbered circles point to specific areas:

- 1: Points to the Headers table in the bottom left.
- 2: Points to the Attachments table in the bottom right.
- 3: Points to the selected row in the Headers table.
- 4: Points to the "Attachments (0)" entry in the Attachments table.

1. Represent NO header in the request that is being sent
2. Represents NO attachments in the request that is being sent to the web server.
3. Represents 10 header information and the same are displayed upon clicking on it.
4. Represents that there are no attachments from the response message.

#### LOGS PANE:

Logs pane has complete information regarding the transaction between the client and the server. Users will be able to see the tabs of the Log pane as shown below. We will discuss the most commonly used log panes when working with SOAP-UI.

**SoapUI log   http log   jetty log   error log   wsrm log   memory log**

**SoapUI Log** – Displays the response information from the webserver. The same information is stored in `soapui.log` file of the SOAP-UI installed folder under ‘bin’ directory.

```
Wed Sep 04 17:43:35 IST 2019:DEBUG:Receiving response: HTTP/1.1 200 OK
Wed Sep 04 17:43:35 IST 2019:DEBUG:Connection can be kept alive indefinitely
Wed Sep 04 17:43:35 IST 2019:INFO:Got response for [CalculatorSoap.Add:Request 1] in 730ms (326 bytes)
Wed Sep 04 17:44:08 IST 2019:DEBUG:Connection closed
Wed Sep 04 18:17:36 IST 2019:DEBUG:Attempt 1 to execute request
Wed Sep 04 18:17:36 IST 2019:DEBUG:Sending request: POST /calculator.asmx HTTP/1.1
Wed Sep 04 18:17:36 IST 2019:DEBUG:Receiving response: HTTP/1.1 200 OK
Wed Sep 04 18:17:36 IST 2019:DEBUG:Connection can be kept alive indefinitely
Wed Sep 04 18:17:36 IST 2019:INFO:Got response for [CalculatorSoap.Add:Add] in 1750ms (326 bytes)
Wed Sep 04 18:18:09 IST 2019:DEBUG:Connection closed
```

SoapUI log http log jetty log error log wsrm log memory log

**Http Log** – Displays all the HTTP packet transfer. All the information in ‘RAW’ is shown in HTTP log.

```
Wed Sep 04 15:31:01 IST 2019:DEBUG:>> "GET /calculator.asmx?wsdl HTTP/1.1[\r][\n]"
Wed Sep 04 15:31:01 IST 2019:DEBUG:>> "Host: www.dneonline.com[\r][\n]"
Wed Sep 04 15:31:01 IST 2019:DEBUG:>> "Connection: Keep-Alive[\r][\n]"
Wed Sep 04 15:31:01 IST 2019:DEBUG:>> "User-Agent: Apache-HttpClient/4.1.1 (java 1.5)[\r][\n]"
Wed Sep 04 15:31:01 IST 2019:DEBUG:>> "[\r][\n]"
Wed Sep 04 15:31:02 IST 2019:DEBUG:<< "HTTP/1.1 200 OK[\r][\n]"
Wed Sep 04 15:31:02 IST 2019:DEBUG:<< "Cache-Control: private, max-age=0[\r][\n]"
Wed Sep 04 15:31:02 IST 2019:DEBUG:<< "Content-Length: 7880[\r][\n]"
Wed Sep 04 15:31:02 IST 2019:DEBUG:<< "Content-Type: text/xml; charset=utf-8[\r][\n]"
Wed Sep 04 15:31:02 IST 2019:DEBUG:<< "Server: Microsoft-IIS/7.5[\r][\n]"
```

SoapUI log http log jetty log error log wsrm log memory log

**Error Log** – Error log displays all the errors that we have encountered during the entire project session. The same information is available in ‘soapui-errors.log’ present in the ‘bin’ directory of the SOAP UI installed location.

**Memory Log** – This tab monitors the memory consumption and displays it in the form of the chart as shown below. It is really helpful when there is a memory intensive operation is performed.



SoapUI log http log jetty log error log wsrm log memory log