

Understanding the SOAP Protocol

Before we create an 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 **S**imple **O**bject **A**ccess **P**rotocol. 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, Python, or C++.
- SOAP requests/response are transported via **HTTP**.

Learn the SOAP Message FORMAT

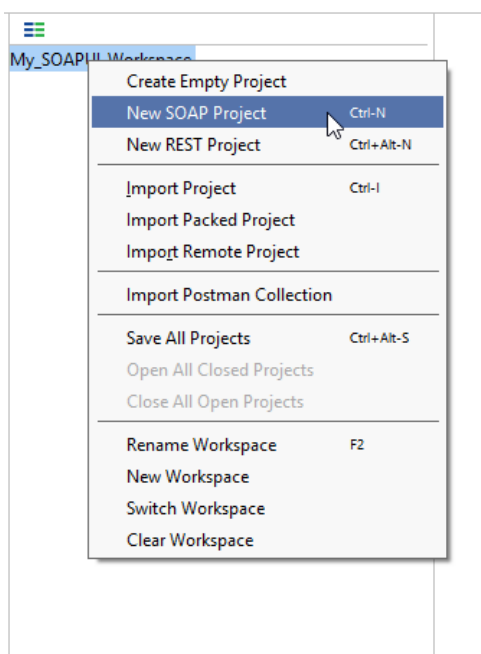
A SOAP message is an ordinary XML document containing the following elements. Message can be either a request message or a response message.

- An **Envelope**(Mandatory) element that identifies the XML document as a SOAP message
- A **Header**(Optional) element that contains header information
- A **Body**(Mandatory) element that contains call and response information
- A **Fault**(Optional) element containing errors and status information

Header (Optional): Used in processing the message, either at an intermediary point or at the ultimate end point.

Create a Project

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



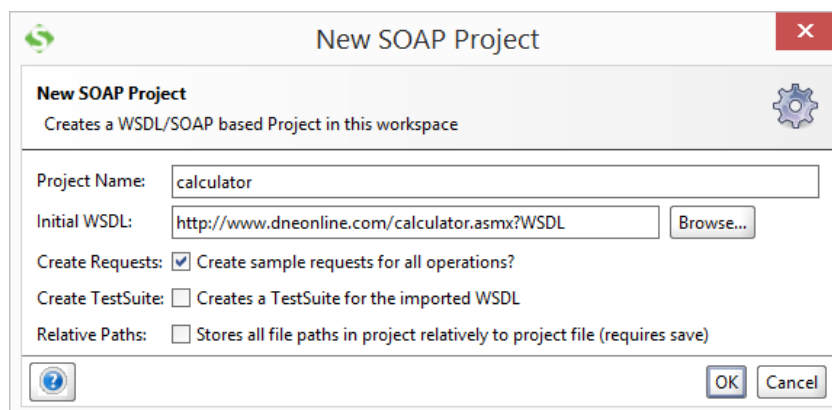
Step 2: We will make use following SOAP request

<http://www.dneonline.com/calculator.asmx?WSDL>

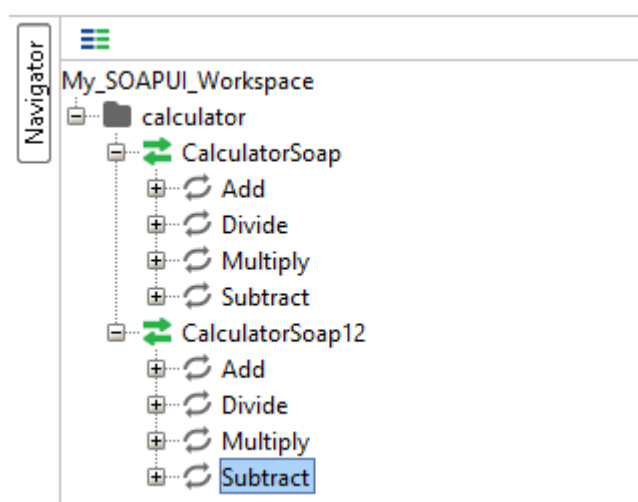
1. Enter the Project Name
2. Enter the path of the WSDL request. In this case
<http://www.dneonline.com/calculator.asmx?WSDL>
3. Click OK

Note:

- Upon checking – **Create sample request for all operation'** 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.
- Upon checking – **Create, a Test Suite for the imported WSDL'** creates a test suite within the project for the imported WSDL.
- Upon Checking – **Relative Path'** enables the user to save all the files relative to the project file.

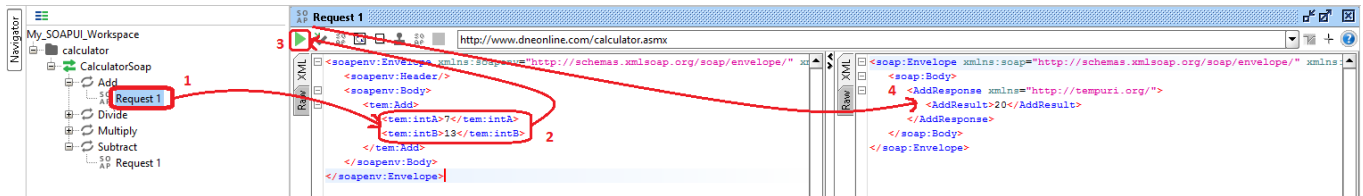


Step 3: Upon creating the SOAP project with the above-said WSDL, we will be able to see that there are two operations that will be imported into the project.



Step 4)

1. Expand the first request and double click on the 'Request1'. It will display the SOAP request in the XML format.
2. Enter the numbers to add two numbers.
3. Click on the submit button
4. Response XML will be displayed right side pane.

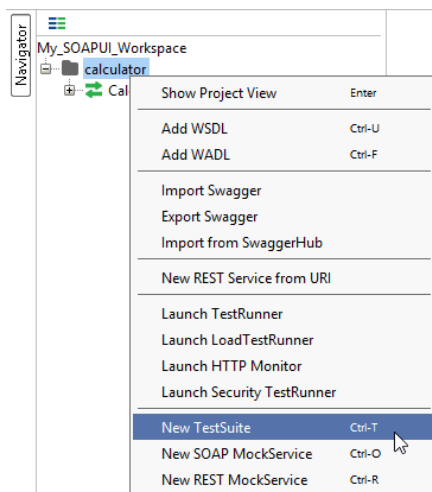


You may wonder why create Test Cases? When you can directly test WebService here...

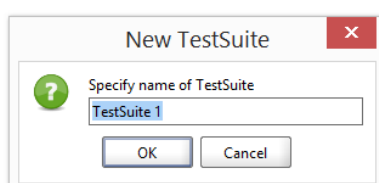
Well, you can send a request for one operation. What about others? **How many combinations you can do using this operation.** You have to edit the request for each and every combination. So, one has to create a test suite/cases to have all possible scenarios tested without having to directly edit the operation itself.

Creating Test Suite

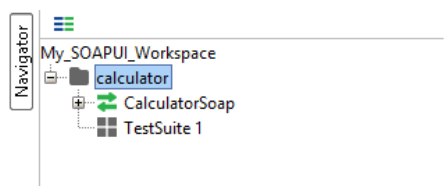
Step 1: Within the project, testers can create a test suite by performing a right click on the root of the project.



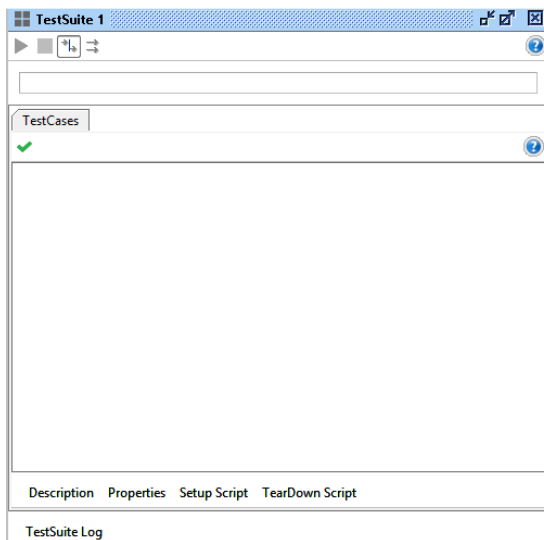
Step 2: We need to enter the name of the test suite and press OK.



Step 3: The created test suite is displayed the navigator pane as shown below.

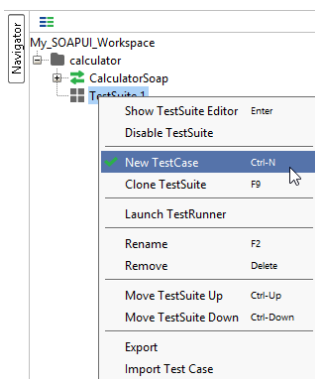


Step 4: The test Suite window opens in the Right Pane. As we have just created there are NO test cases. Hence all options are disabled.



Creating Test Case

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

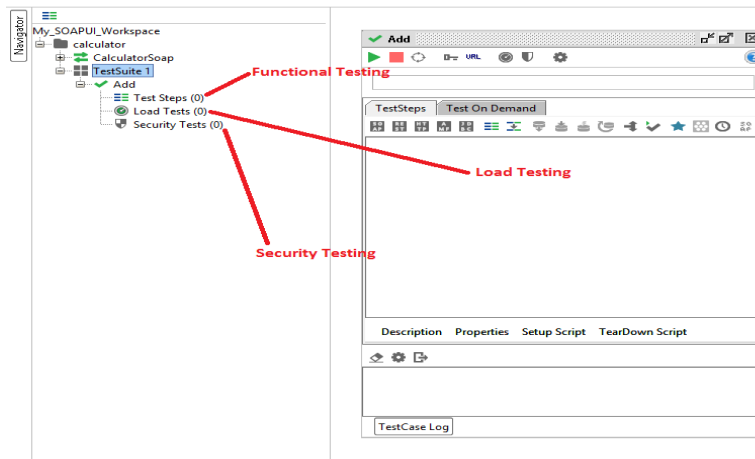


Step 2: Specify the name of the [Test Case](#) and click 'OK'.

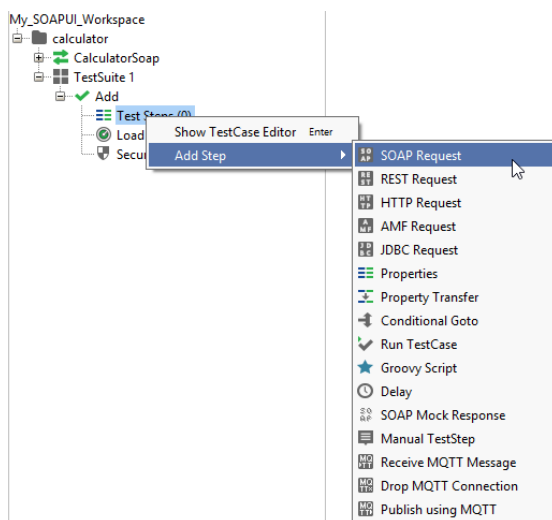


Step 3: The created test case has zero steps as shown below.

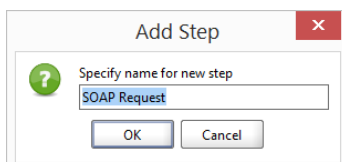
Note: We can see that the test case is added with zero test steps for all kinds of tests available. Upon adding the test steps the numbers in the bracket would change automatically. The functional test step



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.

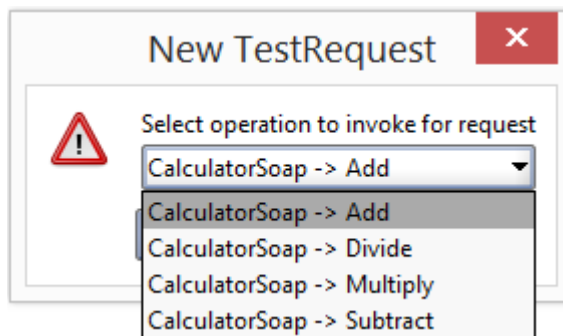


Step 2: Enter the step name and click OK.



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.

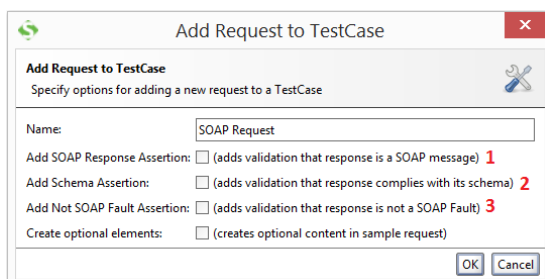
- There are two operations that will be listed. Both the Operations are the same except the SOAP version used.



- The Version does not matter for us in this context. Hence you can select the one of your choice.

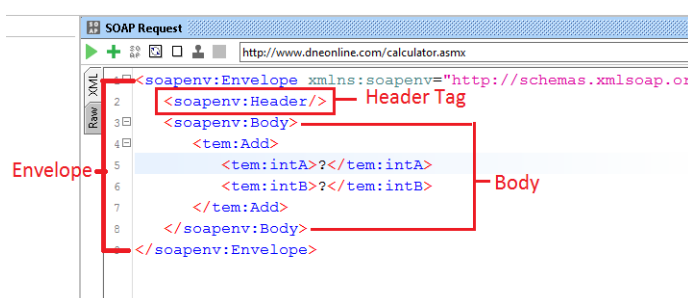
Step 4: While adding a test case, we can add standard assertions. Assertions also called as checkpoints/validation points which we will be dealing in detail later.

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

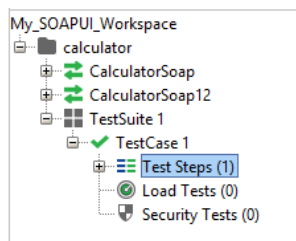


- 1- Verifies if the response message is SOAP, upon executing the test.
- 2- Verifies if the response schema is valid.
- 3- Verifies if the SOAP response contains FAULT.

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

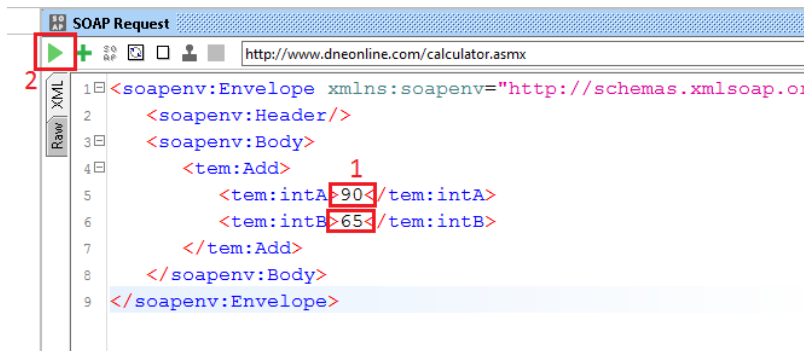


Step 6: The test step count is now incremented to one as we have just added one test step. Similarly upon adding load and security tests step, the corresponding number would be automatically incremented based on the number of steps added

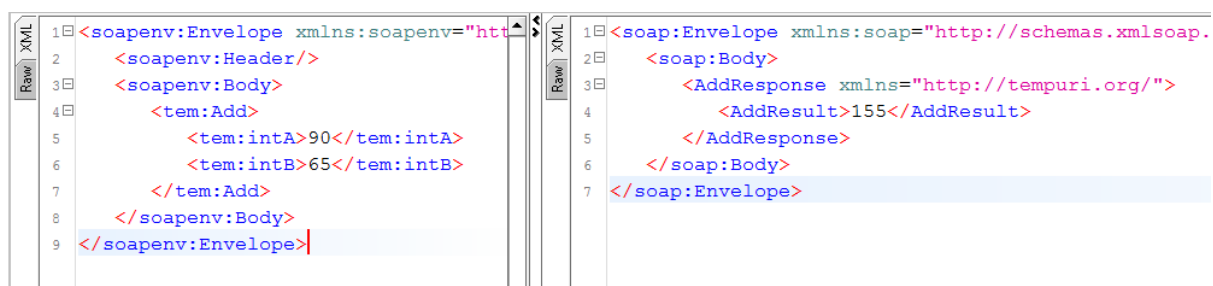


Send Request Manually & Reading Response

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.



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

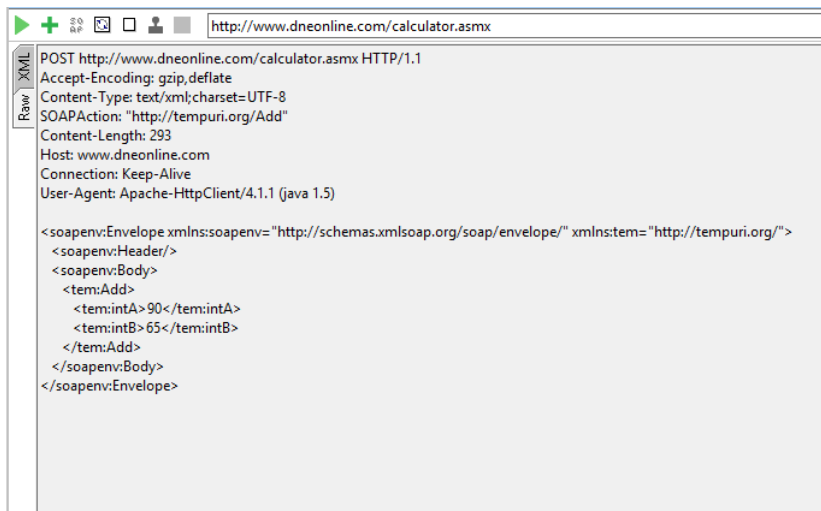


Understanding the Soap Response & Log Panels

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 as shown below

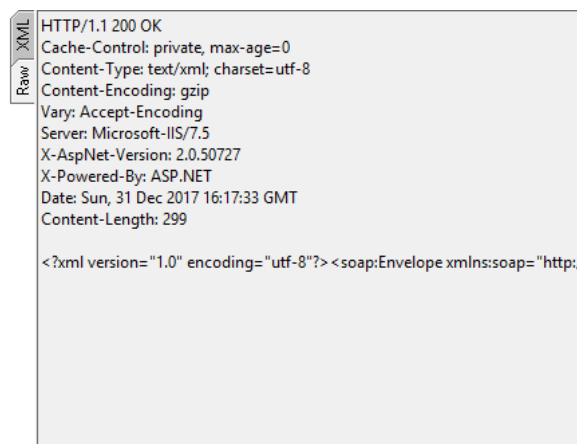


```
POST http://www.dneonline.com/calculator.asmx HTTP/1.1
Accept-Encoding: gzip, deflate
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://tempuri.org/Add"
Content-Length: 293
Host: www.dneonline.com
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:Add>
      <tem:intA>90</tem:intA>
      <tem:intB>65</tem:intB>
    </tem:Add>
  </soapenv:Body>
</soapenv:Envelope>
```

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.

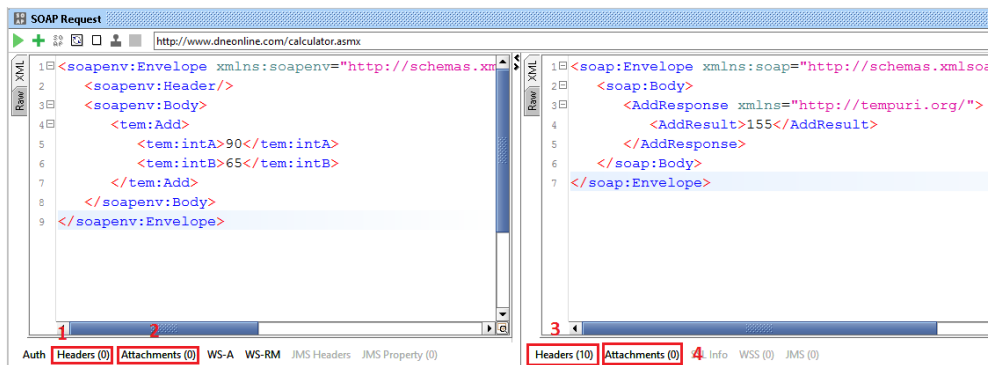


```
HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Content-Encoding: gzip
Vary: Accept-Encoding
Server: Microsoft-IIS/7.5
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Sun, 31 Dec 2017 16:17:33 GMT
Content-Length: 299

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http:
```

A quick over view of Status Code in Http Response:

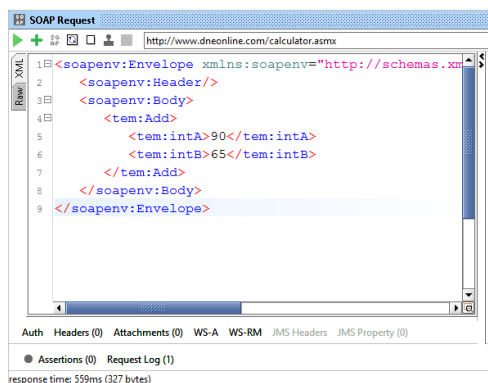
Http Code	Description
1xx:	Informational - This means a request received and continuing process.
2xx:	Success - The action was successfully received, understood, and accepted.
3xx:	Redirection - This means further action must be taken in order to complete the request.
4xx:	Client Error - This means the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



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 the 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.



Log Panel

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

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

```

Sun Dec 31 21:32:42 IST 2017:INFO:importing operation Subtract
Sun Dec 31 21:33:14 IST 2017:DEBUG:Connection closed
Sun Dec 31 21:36:02 IST 2017:INFO:Settings saved to [C:\Users\User\soapui-settings.xml]
Scheduling garbage collection every 60 seconds
Sun Dec 31 21:36:08 IST 2017:INFO:Reloading updated settings file
Sun Dec 31 21:36:08 IST 2017:INFO:initialized soapui-settings from [C:\Users\User\soapui-settings.xml]
Sun Dec 31 21:36:35 IST 2017:INFO:Settings saved to [C:\Users\User\soapui-settings.xml]
Scheduling garbage collection every 60 seconds
Sun Dec 31 21:36:38 IST 2017:INFO:Reloading updated settings file
Sun Dec 31 21:36:38 IST 2017:INFO:initialized soapui-settings from [C:\Users\User\soapui-settings.xml]
Sun Dec 31 21:47:34 IST 2017:DEBUG:Attempt 1 to execute request
Sun Dec 31 21:47:34 IST 2017:DEBUG:Sending request: POST /calculator.asmx HTTP/1.1
Sun Dec 31 21:47:34 IST 2017:DEBUG:Receiving response: HTTP/1.1 200 OK
Sun Dec 31 21:47:34 IST 2017:DEBUG:Connection can be kept alive indefinitely
Sun Dec 31 21:47:34 IST 2017:INFO:Got response for [CalculatorSoap.AddSOAP Request] in 559ms (327 bytes)
Sun Dec 31 21:48:09 IST 2017: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.

```
Sun Dec 31 21:47:34 IST 2017:DEBUG:>> " </soapenv:Body>[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:>> " </soapenv:Envelope> "
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "HTTP/1.1 200 OK[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "Cache-Control: private, max-age=0[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "Content-Type: text/xml; charset=utf-8[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "Content-Encoding: gzip[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "Vary: Accept-Encoding[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "Server: Microsoft-IIS/7.5[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "X-AspNet-Version: 2.0.50727[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "X-Powered-By: ASP.NET[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "Date: Sun, 31 Dec 2017 16:17:33 GMT[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "Content-Length: 299[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "[v]\n]"
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "[0x1f][0x8b][0x00][0x00][0x00][0x00][0x00][0xed][0xbd][0x77][0x1e
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "[0xfb][0xdf][0xd5][0xd5][0x0a][0x1e][0xde][0xdb][0xd9][0xdf
Sun Dec 31 21:47:34 IST 2017:DEBUG:<< "[0xba][0xef][0x10][0x00]"
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.