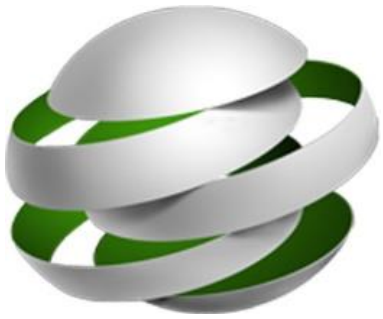


WEB SERVICE TESTING

Lecture Notes

Learn about Web Service Test, SOAP UI Pro.

Softech Solutions Inc.

www.softechsolutions.tech

saney.alam@softechsolutions.tech

Web Service – Overview

What is Web Service? Why is it needed?

In General, software applications are developed to be consumed by the human beings, where a person sends a request to a software service which in-turn returns a response in human readable format.

In the modern era of technology if you want to build a software application you don't need to build each and everything from scratch. There are lots of readymade services available which you can plug into your application and you can start providing those services in your application.

For example you want to display whether forecast information you don't need to collect, process and render the data in your application. You can buy the services from the people who already well-established in processing and publishing such kind of data.

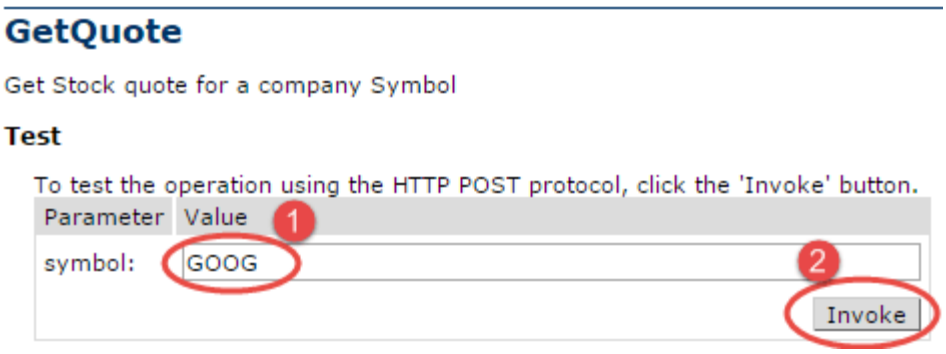
Web services allow us to do these kinds of implementations. ***Web Services is the mechanism or the medium of communication through which two applications / machines will exchange the data irrespective of their underline architecture and the technology.***

As an example, consider the following Web Service

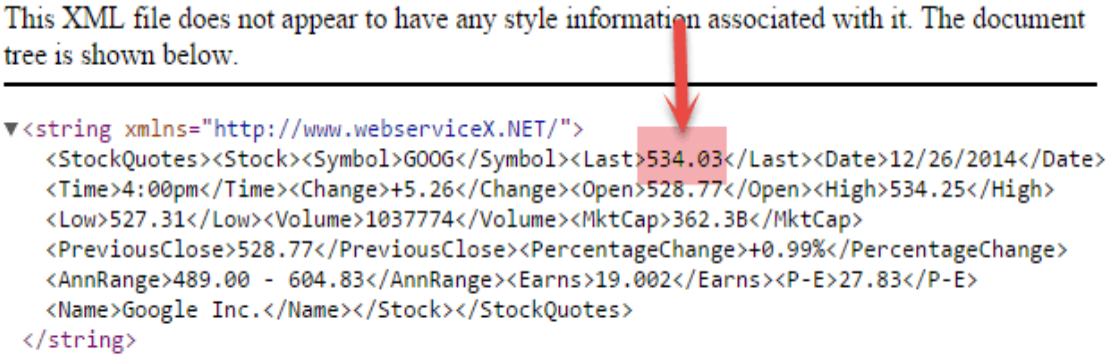
http://www.webservicex.net/stockquote.asmx?op=GetQuote

It gives Share Value for a Company.

Let's find share price for Google (Symbol: GOOG)



The response XML gives the stock price.



This Web Service can be called by a Software Application using SOAP or HTTP protocol.

Web Services can be implemented in different ways, but the following two are the popular implementations approaches.

1. SOAP (Simple Object Access Protocol)
2. REST (Representational State Transfer architecture)

SOAP

SOAP is a standard protocol defined by the W3C Standard for sending and receiving web service requests and responses.

SOAP uses the **XML format to send and receive the request** and hence the data is platform independent data. SOAP messages are exchanged between the provider applications and receiving application within the SOAP envelops.

As SOAP uses the simple http transport protocol, its messages are not got blocked by the firewalls.

REST

REST means REpresentational State Transfer; it is an architecture that generally runs over HTTP. The REST style emphasizes the interactions between clients and services, which are enhanced by having a limited number of operations. REST is an alternative to SOAP (Simple Object Access Protocol) and instead of using XML for request REST uses simple URL in some cases. Unlike SOAP, RESTFUL applications uses HTTP build in headers to carry meta-information.

There are various code that REST use to determine whether user has access to API or not like code 200 or 201 indicates successful interaction with response body while 400 indicates a bad request or the request URI does not match the APIs in the system. All API request parameters and method parameters can be sent via either **POST** or **GET** variables.

Rest API supports both XML and JSON format. It is usually preferred for mobile and web apps as it makes app work faster and smoother

There is one more thing one needs to learn:

WSDL

WSDL (Web Services Description Language) is an XML based language which will be used to describe the services offered by a web service.

WSDL describes all the operations offered by the particular web service in the XML format. It also defines how the services can be called, i.e what input value we have to provide and what will be the format of the response it is going to generate for each kind of service.

Web Service Testing

Web Services Testing basically involves

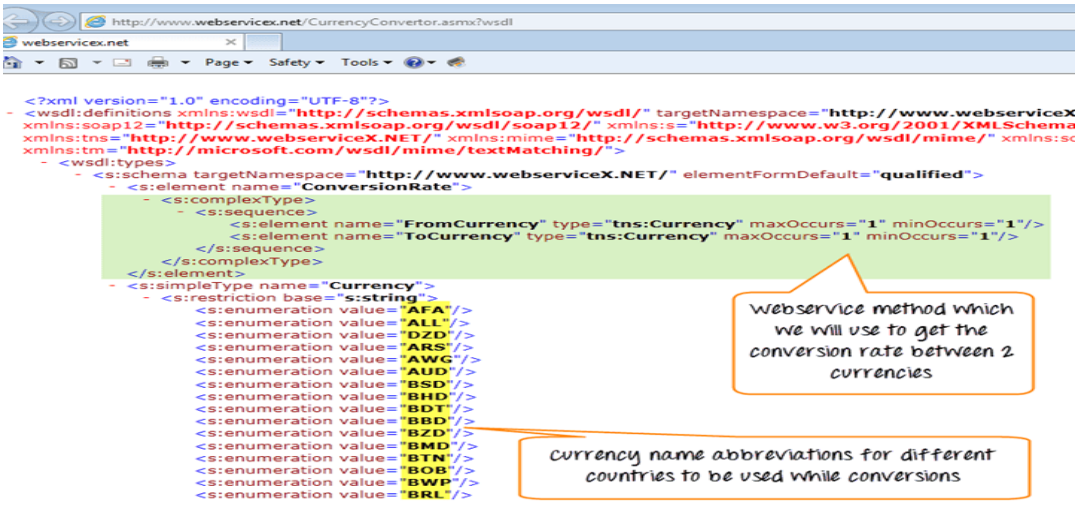
1. *Understand the WSDL file*
2. *Determine the operations that particular web service provides*
3. *Determine the XML request format which we need to send*
4. *Determine the response XML format*
5. *Using a tool or writing code to send request and validate the response*

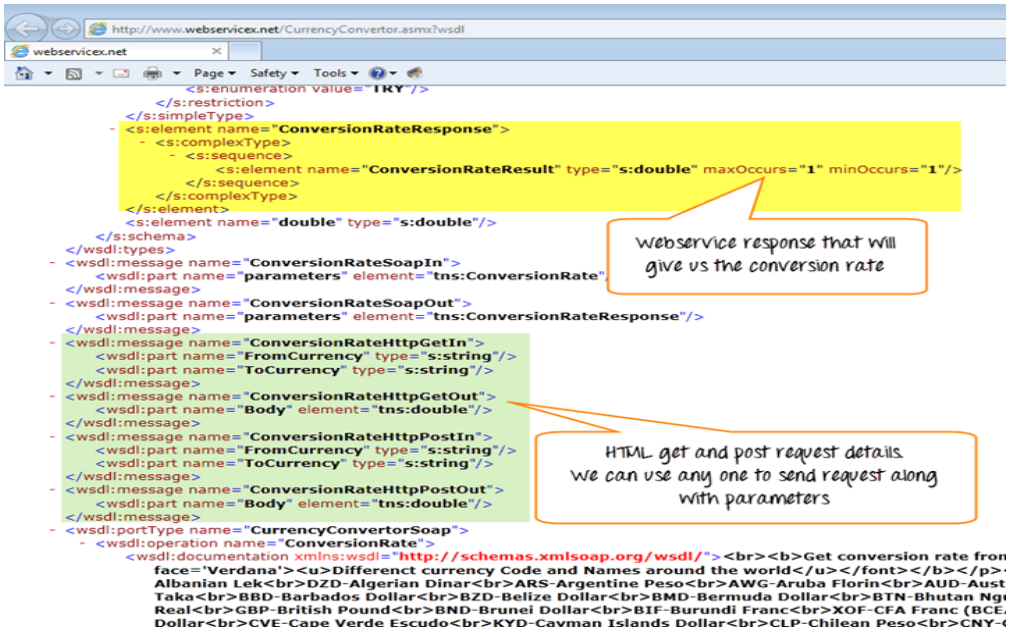
Suppose we want to test a Web Service which provides Currency Conversion Facility. It wills the current conversion rates between the different countries currency. This service we can use in our applications to convert the values from one currency to the other currency.

Now let's look at above steps

Step 1 to 4: Understanding WSDL and determining operations & XML formats

Currency Convertor WSDL file can be seen @ (http://www.webservicex.net/CurrencyConvertor.asmx?wsdl) which will give the information about the Currency Convertor web service methods which it will support, the parameter which we need pass and the type of parameters… etc

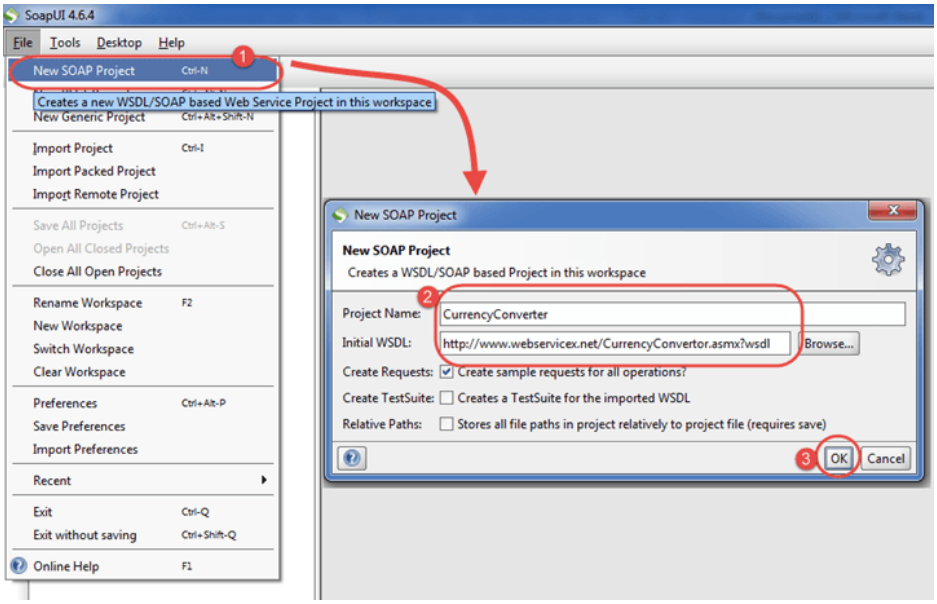




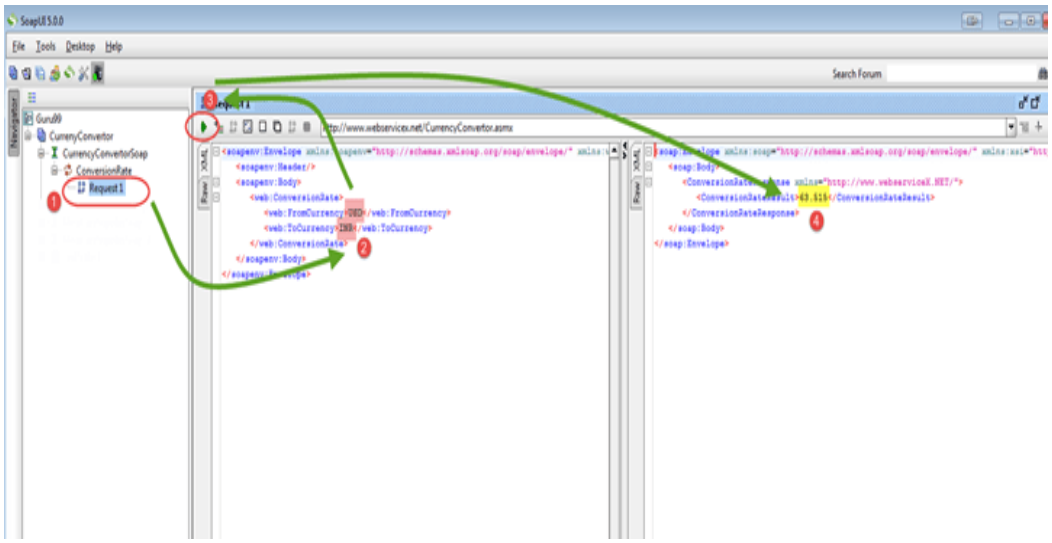
Step 5: Using SoapUI to Test the Web Service

In SoapUI

1. Go to File > New Soap Project
2. Enter the project Name and the WSDL URI location
3. Click OK



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



SOAP UI – Overview

What is SOAP UI?

- SOAP-UI is the leading open source cross-platform API Testing tool
- SOAP-UI allows testers to execute automated functional, regression, compliance, and load tests on different Web API.
- SOAP-UI supports all the standard protocols and technologies to test all kinds of API's.
- SOAP UI interface is simple that enables both technical and non-technical users to use seamlessly.

Why use SOAP UI?

SOAP-UI is not just a functional API testing tool but also lets us perform non-functional testing such as performance and security test.

Let us discuss the 5 important features of SOAP-UI

1) Functional Testing

- A powerful tool allows testers to write Functional API Tests in SoapUI
- Supports Drag-Drop feature which accelerates the script development
- Supports debugging of tests and allows testers to develop data driven tests.
- Supports Multiple Environments - Easy to switch between QA, Dev and Prod Environments
- Allows advanced scripting (tester can develop their custom code depending on the Scenario)

2) Security Testing

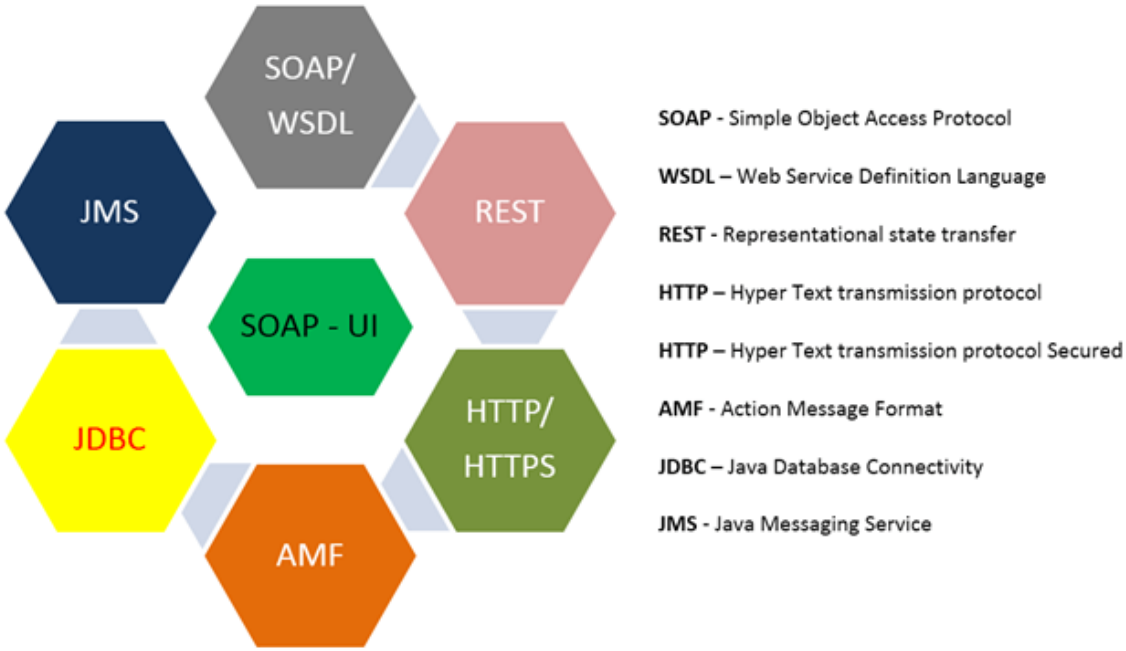
- Has the capability to perform a complete set of vulnerability scan.
- Prevents SQL Injection to secure the databases
- Scans for Stack overflows that are caused by documents huge in size
- Scans for Cross Site Scripting, which usually occurs when service parameters are exposed in messages.
- Performs Fuzzing scan and Boundary scan to avoid erratic behavior of the services.

3) Load Testing

- Distribute the Load Tests across any number of load UI Agents.
- Simulate high volume and real-world load testing with ease.
- Allows advanced custom reporting to capture performance parameters.
- Allows End-to-End System Performance Monitoring

4) Supported Protocols/Technologies:

- SoapUI has the most comprehensive Protocol Support



5) SOAP-INTEGRATION with Other Automation Tools:

SoapUI integrated very well with popular tools

- Maven

  Apache **Maven** is a software project management tool that can manage a project's build, reporting and documentation from a central repository. Maven can also execute SOAP-UI tests within Maven Build using simple commands.

- HUDSON

  HUDSON, a JAVA based Continuous integration tool and integrates with tools such as CVS, Subversion, Git, Perforce, Clearcase, and RTC. SOAPUI also integrates with HUDSON, which helps us to spot bugs quickly for each and every commit by the developers.

- JUnit

  JUnit is a unit testing framework built in Java, which can control the flow of tests from SOAP-UI as well.

- Apache - Ant

Apache Ant, a Java library which is a command-line tool that helps in building software. Using SOAP-UI's Command line, we can execute tests within an ANT Automated Build.

SOAP UI Vs Selenium

Let's compare SoapUI with Selenium

SOAP UI	Selenium
SOAP UI is NOT used for User Interface Testing. It is only used for Web API or Web Service Testing	Selenium is used for User Interface Testing.
Capability to test the data sent and received between the web browser and a web server. Can test the protocols/technologies such as REST, SOAP.	Selenium cannot test protocols, but they can test the UI behavior.
Able to perform functional, load and security testing of the above-mentioned technologies.	Selenium can perform only functional testing. Performance testing to some extent because we can track execution time with regards to the performance but cannot test multi user and multi tenancy. Selenium certainly cannot be used for security testing.
It is PROTOCOL Dependent and NOT browser dependent.	Selenium depends on the browser capabilities.

When use SoapUI PRO Version?

Data Driven: PRO Version, helps us to work with an external data source such as text files, XML, Groovy, Excel, file, and Databases. This helps us to scale our tests with a range of inputs driven through the above-said sources.

Test Coverage: PRO Version allows testers to get a statistic report which shows the functionalities that are well tested and also the areas that are NOT thoroughly tested as well. The drill down reports even pinpoints exactly what has NOT been tested and what has NOT been asserted.

Test Debugging: You can then run the test to that breakpoint and view the current value of the SoapUI properties. The Test Debugging Interface simplifies following Test Flow, Variables, Properties, Requests, Context, and much more, making test creation and improvement more streamlined.

Multi-Environment Support: Working with multiple environments such as DEV, QA, Pre-PROD environment can be a daunting task with the open source version as testers need to change the end points to execute in different environments. PRO version helps us to switch between environments seamlessly.

Reporting: PRO version is loaded with many options to customize reports that generate detailed reports at Project, TestSuite, TestCase or LoadTest level. It also produces reports in various formats such as PDF, HTML, Word or Excel.

Security Testing: Both SOAP UI versions has capabilities to test for security vulnerabilities such as XML bombs, SQL injections, fuzzing, cross-site scripting. However, only SOAP UI PRO can perform vulnerability scans using Security Test Generator using a mouse click.

SQL Builder: For Non-Technical testers writing complex SQL Query can be cumbersome. The SOAP UI PRO's SQL Builder can help them in creating SQL Query using the SQL Builder's graphical interface. This feature helps us to accelerate the implementation of data-driven testing.

Support: As part of the license agreement SOAP UI Pro has exclusive support apart from the online forum support.

SOAP UI – Installation & Configuration

System Requirements

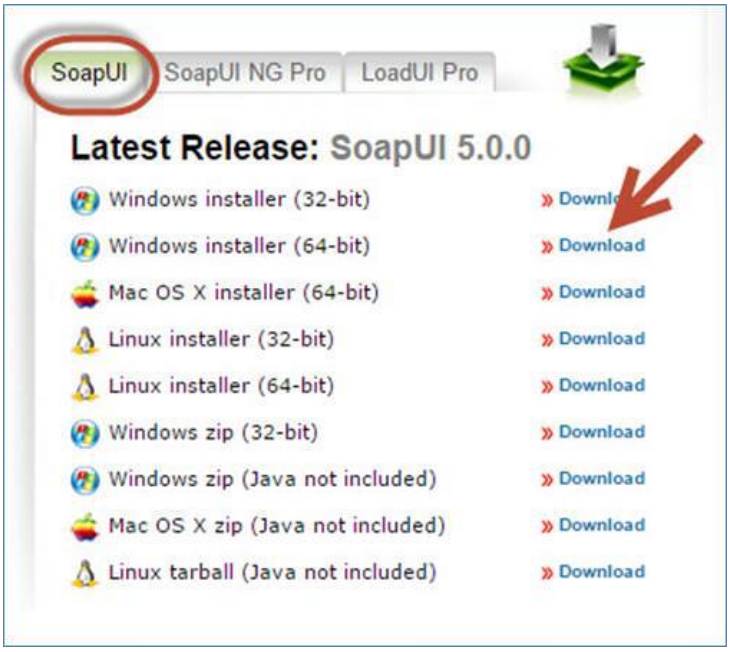
SOAP UI Installation Prerequisite		
		
1GHz or higher 32-bit or 64-bit processor	1GHz or higher 32-bit or 64-bit Intel or AMD64 processor	1GHz or higher 32-bit or 64-bit Intel or PowerPC processor
512MB of RAM	512MB of RAM	512MB of RAM
Min 200MB of hard disk space for installation	Min 240MB of hard disk space for installation	140MB of hard disk space for installation
Windows XP or later	Ubuntu, Red Hat, Fedora or other distributions	Mac OS X 10.4 or later; Mac OS X Server 10.4 or later
Java 6 or later	Java 6 or later	Java 6 or later

Installation Process Roadmap

Here is the roadmap for the installation Process

Part A – Downloading

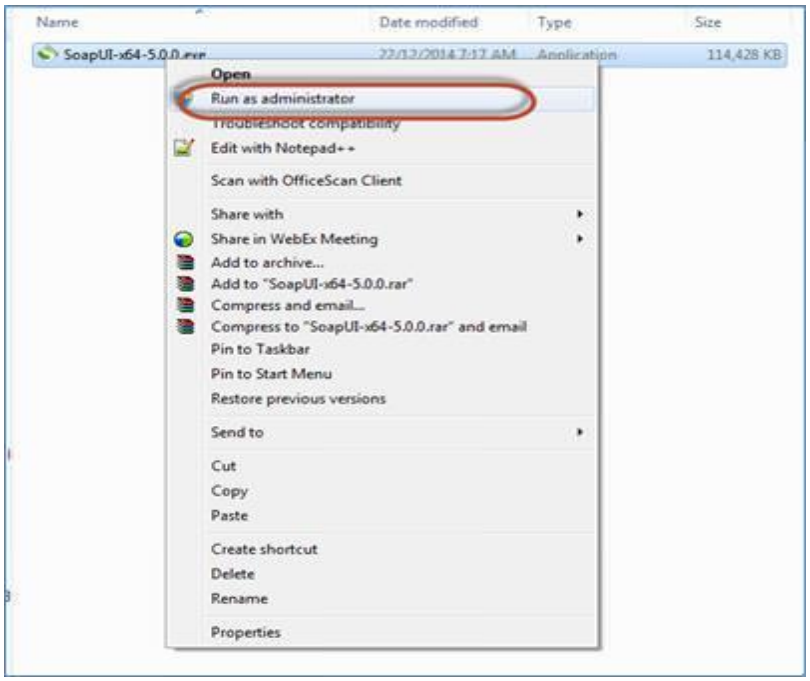
- Navigate to http://www.soapui.org/
- Scroll down and choose the downloader based on your operating system. In this tutorial, we will install SOAP UI on a 64 bit Microsoft Windows operating system.



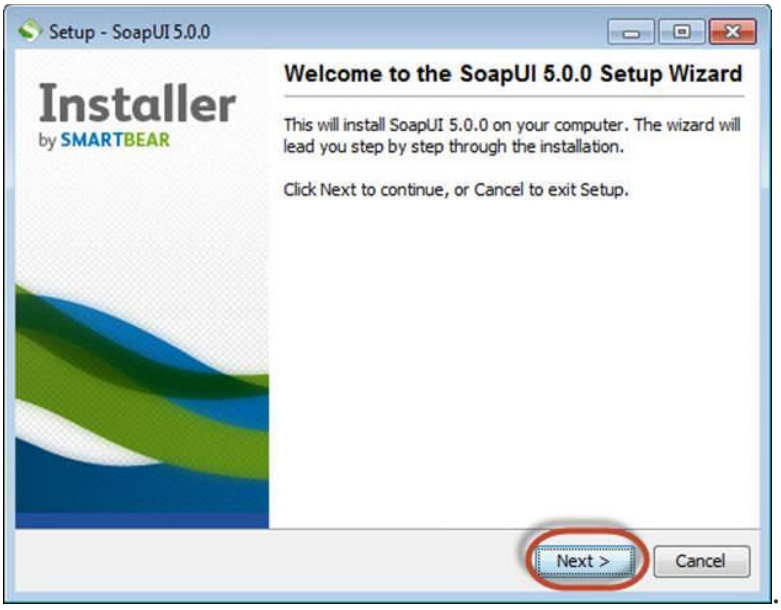
- Upon clicking download, the user is automatically forwarded to http://sourceforge.net/ and the installer download starts automatically.

Part B: Installing

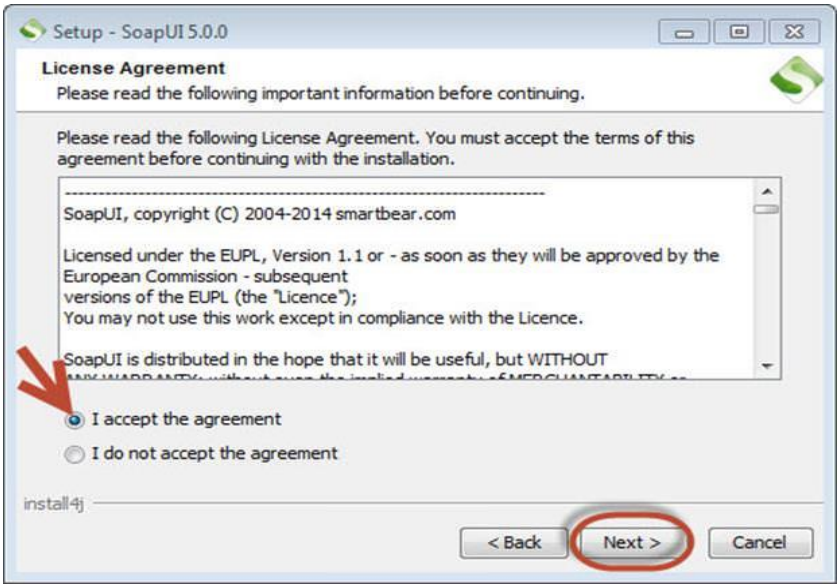
Step 1: After downloading, execute the file as 'Administrator' as shown below



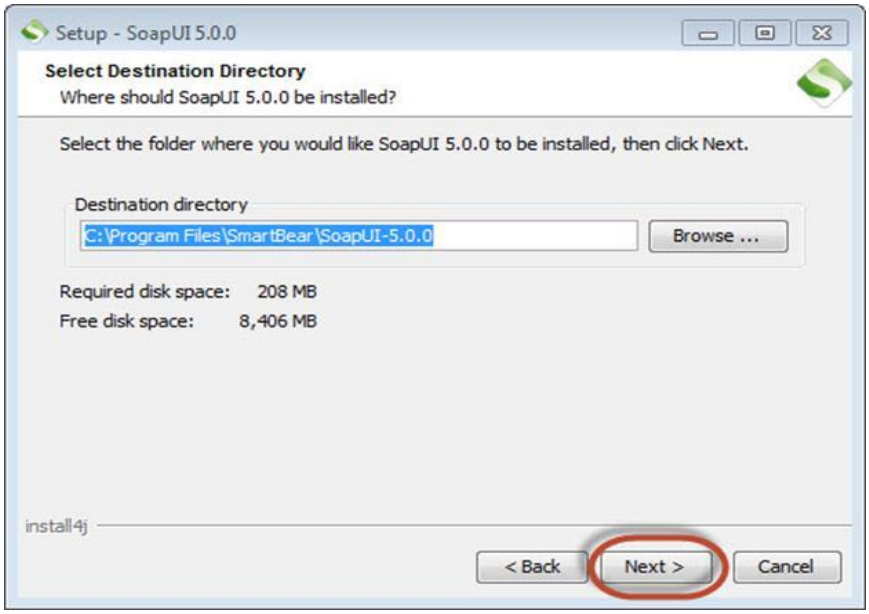
Step 2: In the setup wizard, click 'Next' to continue.



Step 3: Accept the license agreement and click 'Next' to continue.



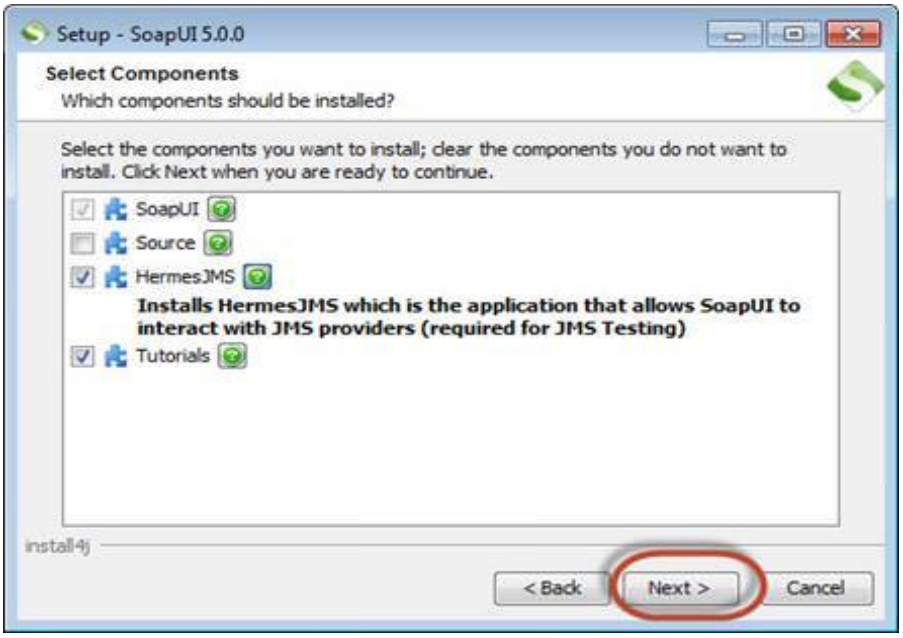
Step 4: Choose the installation directory or leave the default installation directory as is.



Step 5: Choose the components that you wish to install.

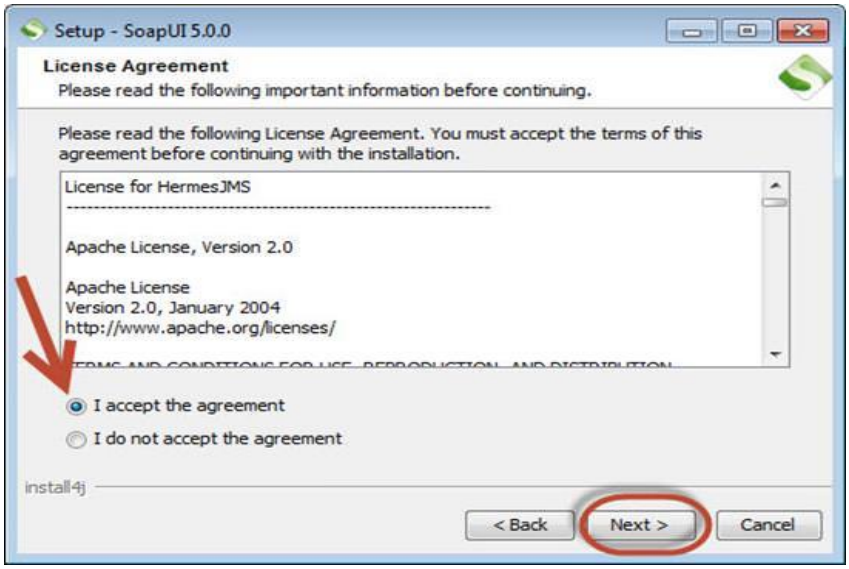
- SOAP UI is checked by default and NOT user configurable.
- Source – Enable, if you would like to get access to the source code of SOAP-UI. We have not selected it.
- Hermes JS – Enable, if the application requires JMS testing.

- Tutorial – Enable, if you want to access SOAP-UI tutorials Post installation.

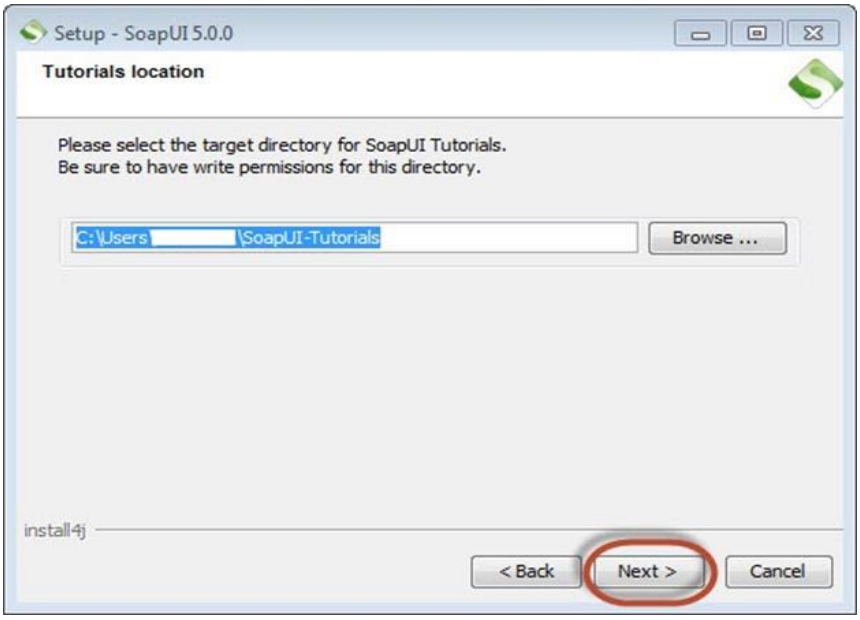


Step 6: The installation wizard asks the user to download and install 'Load UI'. Since Load testing is not the context of the discussion, we can proceed without selecting it.

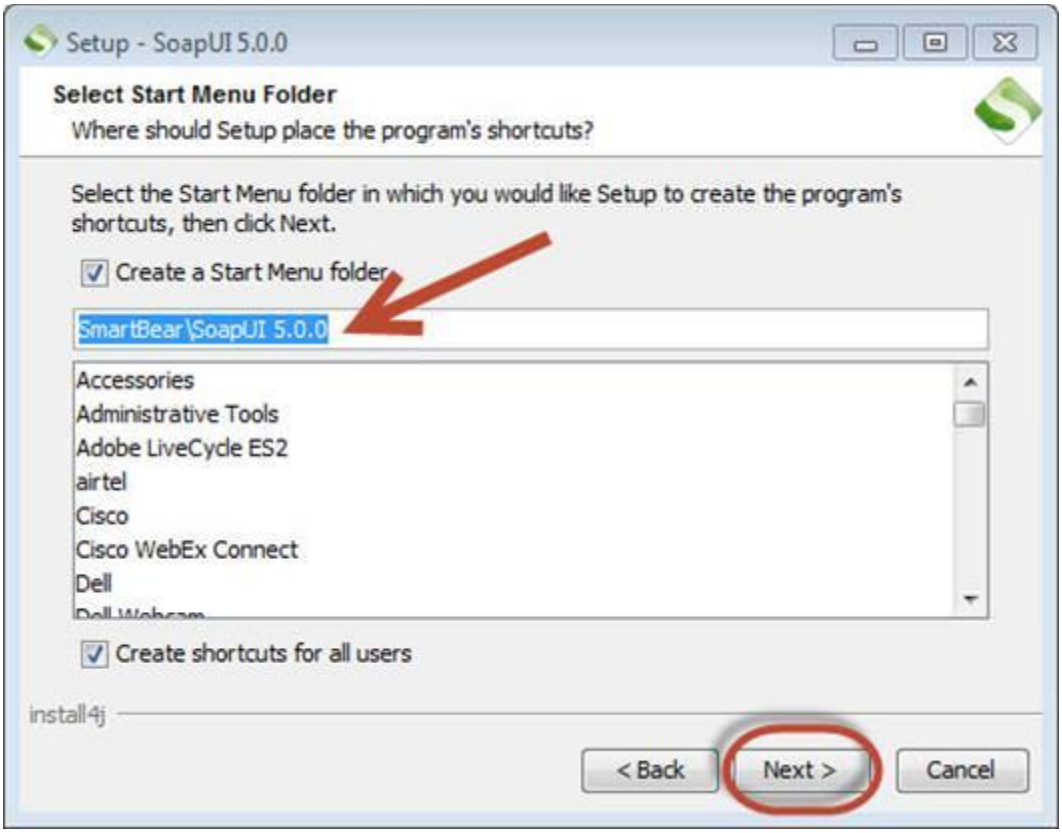
Step 7: If **'Hermes JMS'** is selected in **step#5**, then the license agreement for 'Hermes JMS' pops up. Accept the license agreement and click 'Next'.



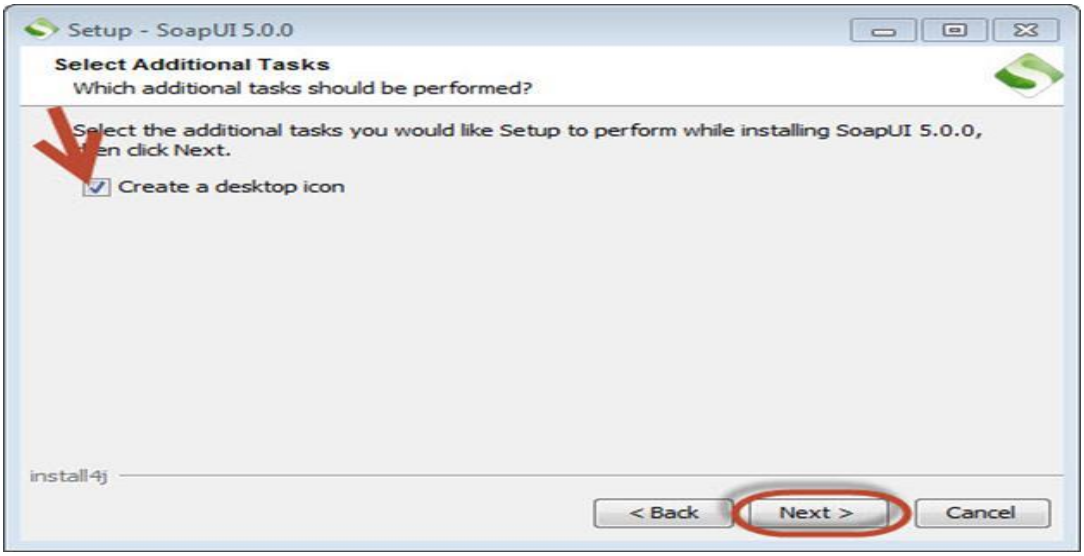
Step 8: Choose the folder location for tutorials or else leave the default location as is and click 'Next'.



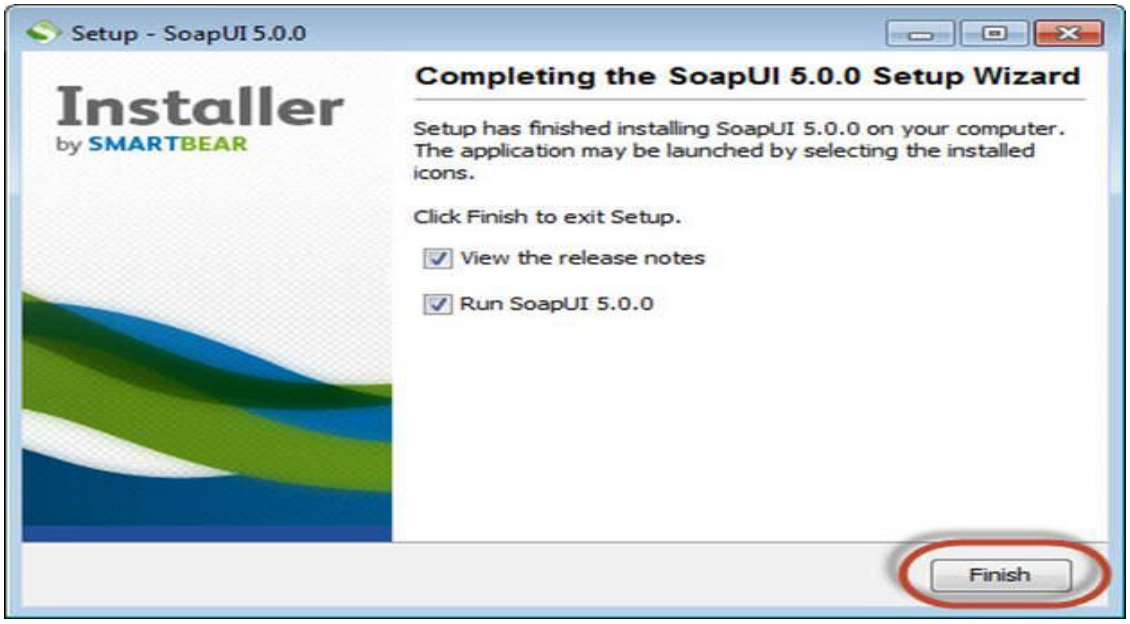
Step 9: Choose the start menu folder location or else leave the default location as is and click 'Next'.



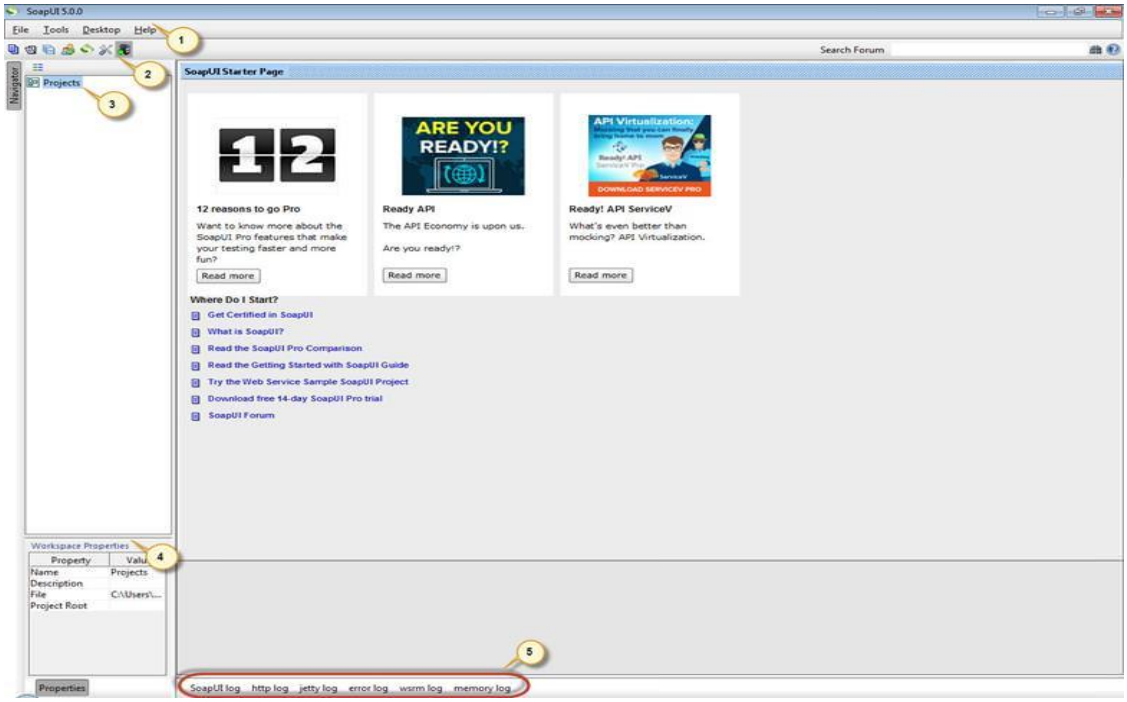
Step 10: Enable the checkbox 'create a desktop icon' and click 'Next'.



Step 10: The Installation starts and upon completing the same, the wizard shows the below status. Click 'Finish'.

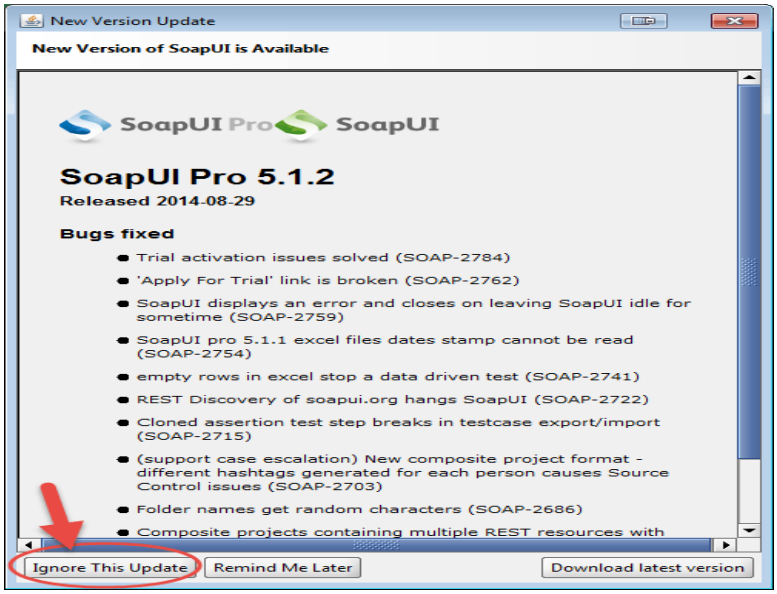


Step 11: Upon clicking the 'Finish' button, SOAP UI is launched.



1. Menu Bar
2. Quick access toolbar
3. Project Navigation Area
4. Workspace Properties
5. Log area

NOTE: You might see an upgrade screen like below. You can just "***Ignore the Update***"

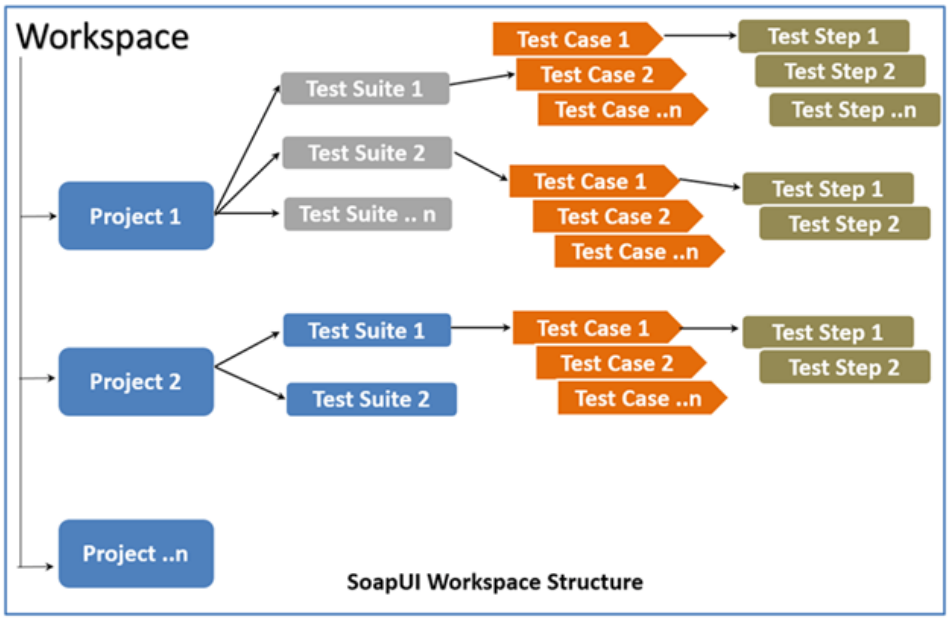


Part C: Configuring

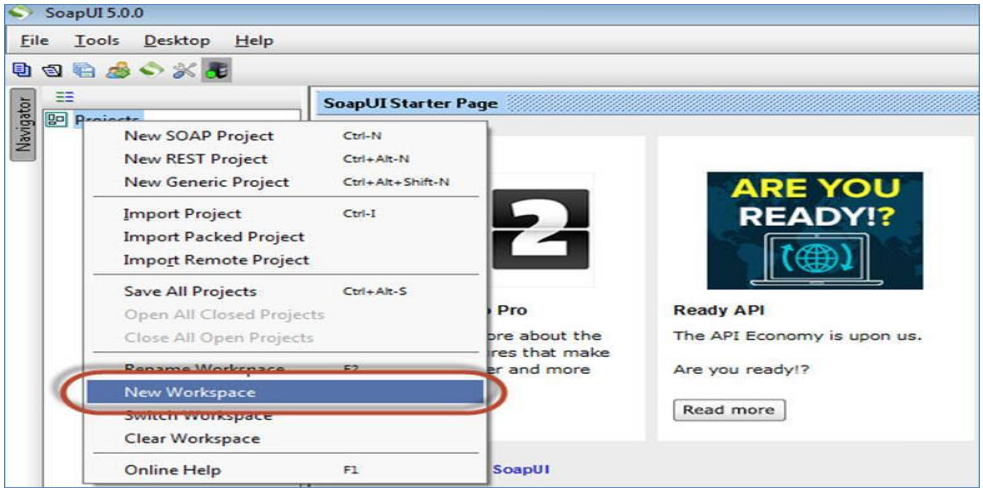
Let us first understand the project structure in SOAP UI.

- First step in SOAP UI is to create a workspace. There can be several projects associated with a workspace. Users can create more than one workspace. In SOAP UI Pro version, we can seamlessly switch environments to map it to different end points.
- For Each project, we can create several test suites.
- For Each Test Suite, there can be several test cases attached to it.
- For Each Test Case, there can be several test steps associated with it.

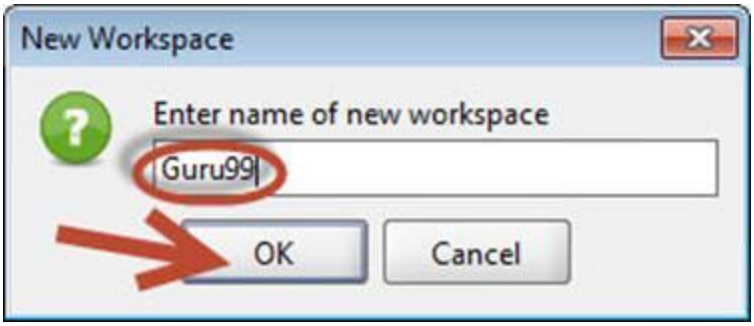
Below is the pictorial representation of a workspace structure in SOAP-UI.



Step 1: First step is to create a workspace. Create a workspace as shown below. All the artifacts that we are going to create from now on would be contained in this workspace.

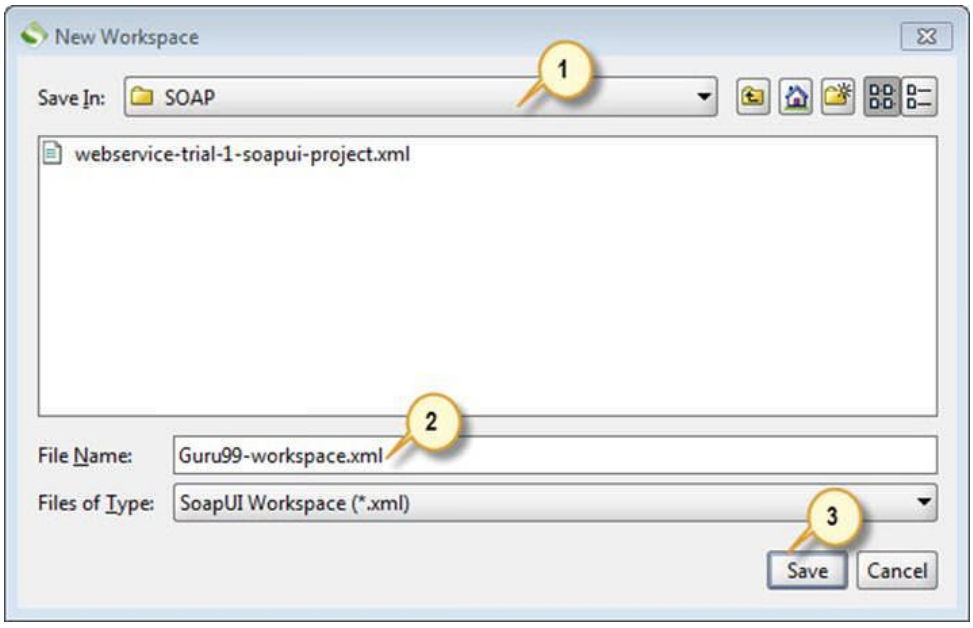


Step 2: Enter a name for the workspace and click 'OK.'

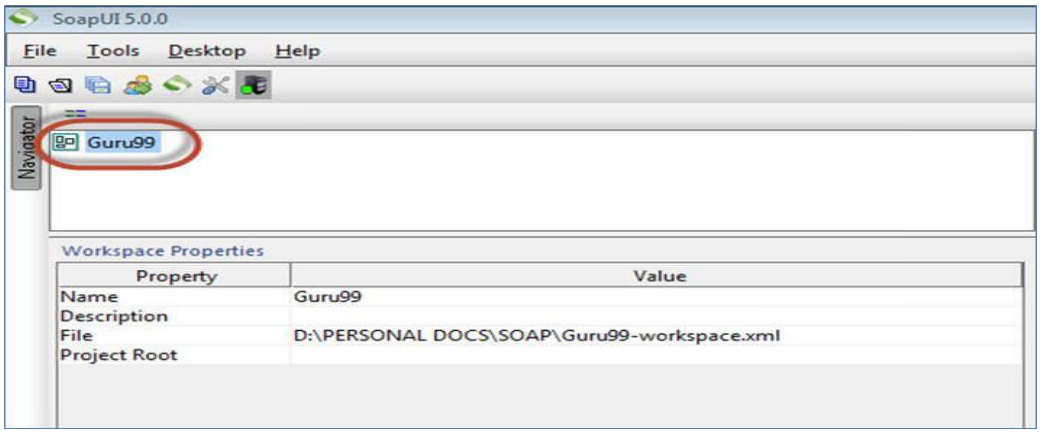


Step 3: Now the user has to select the path where this workspace has to be saved.

1. Select the path where the workspace has to be saved
2. The name of the workspace XML, which has to be located when user wants to open the workspace in the future.
3. Click 'Save'.



Step 4: The workspace is created as shown below. We can also access the workspace properties under 'Workspace Properties' Tab.

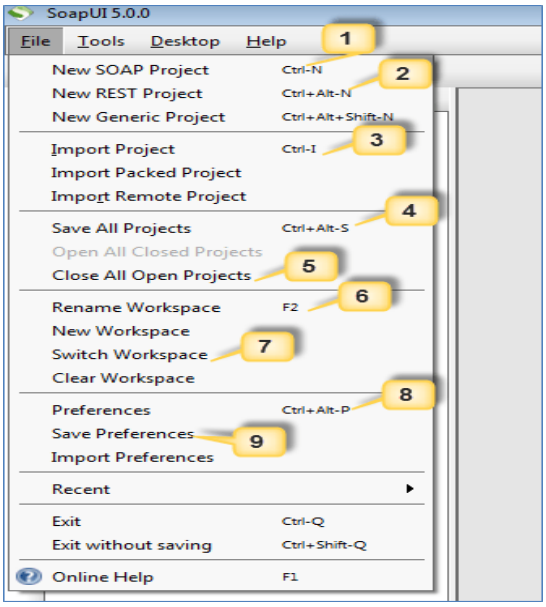


Now we have successfully configured SOAP-UI after downloading and installing such that we can continue to perform testing.

Basic GUI Navigation

FILE MENU:

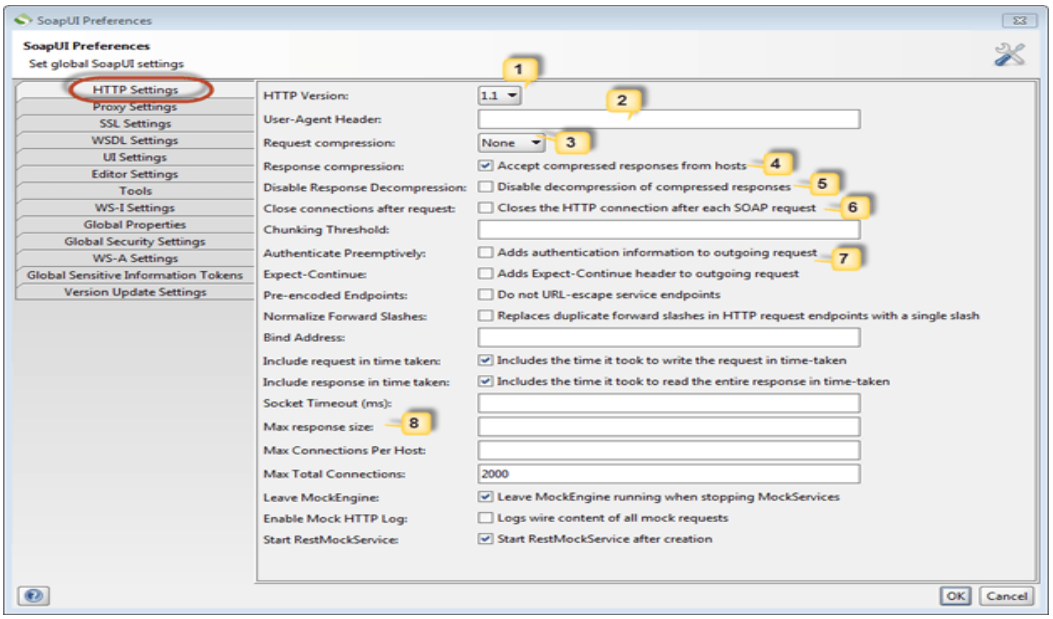
- Understanding the most frequently used navigations within SOAP-UI is very important for seamless navigation while working on real-time projects. Let us take a look at the file menu first.



1. 'New SOAP Project' allows user to create a Project by importing SOAP Request.
2. 'New REST Project' allows user to create a Project by importing REST Request.
3. 'Import Project' allows user to import the entire project by locating the corresponding XML.
4. 'Save All Projects' allows user to save all the opened projects in a single click.
5. 'Close All Open Projects' closes all the projects opened in that workspace.
6. 'Rename Workspace' allows user to rename the previously created workspace.
7. 'Switch Workspace' allows user to switch between workspaces.
8. 'Preferences' allows user to customize SOAP UI. We will deal with it in next section.
9. 'Save Preferences' allows user to save their customized settings. When SOAP UI opened for the next time, it uses the user saved preferences.

FILE >> PREFERENCES >> HTTP SETTINGS:

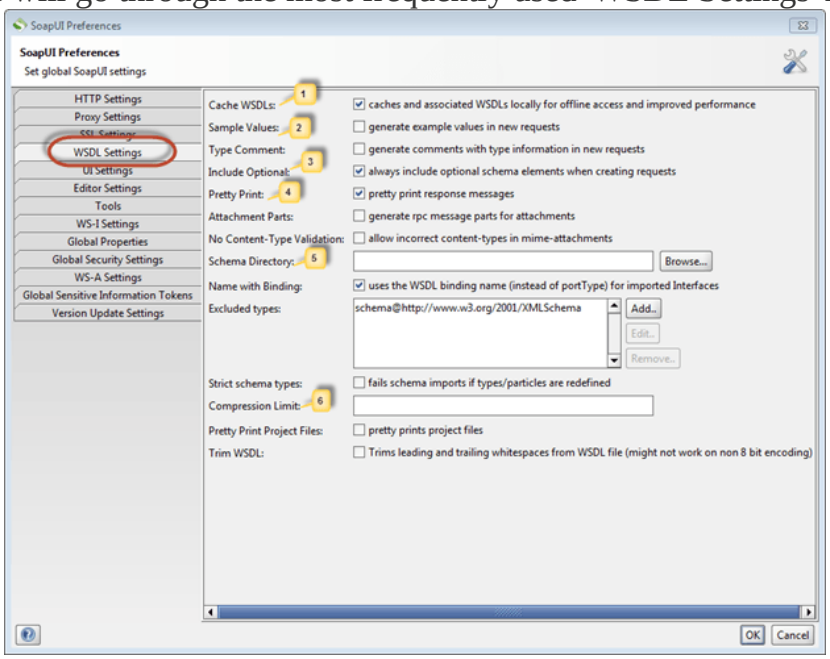
- Now, let us understand the 'Preferences' from the file menu. Upon Clicking 'Preferences' from 'File' Menu, the below dialog opens.
  - We will go through the most frequently used 'HTTP Settings' in details.



1. Denotes the HTTP Version to be used for request and response.
2. 'User-Agent Header' allows user to can be predefined using this option. If not defined, it uses the default http client header.
3. Allows user to specify the compression method. It can be either gzip or deflate or None.
4. 'If Checked', allows compressed response from hosts.
5. 'If Checked' disables decompression of the compressed responses.
6. 'If Checked' closes HTTP connection for each SOAP Request.
7. 'If Checked', allows user to specify authentication information for the outgoing requests.
8. Allows user to restrict the maximum number of bytes to be read from a response. ZERO corresponds to unlimited size.

FILE >> PREFERENCES >> WSDL SETTINGS:

- Now, we will go through the most frequently used 'WSDL Settings' in details.

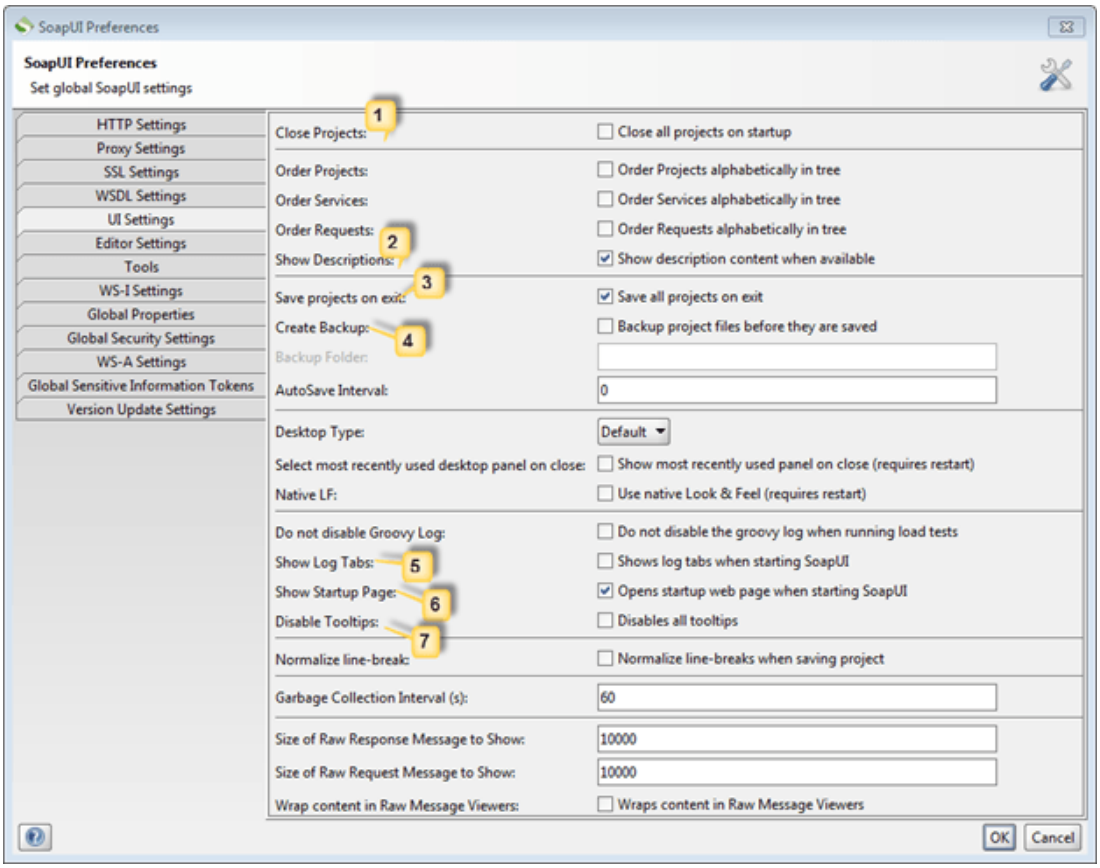


1. Cache WSDLs Turns on and off caching of WSDL's
2. Generates example values in requests
3. Allows users to always include optional elements in generated requests

4. Response messages are printed in the response editor
5. Allows user to specify a directory containing schema (.xsd) files while validating WSDL requests. Upon changing the contents of this directory SOAP UI requires a restart.
6. For the purpose of preserving space, the minimum message size to be compressed in the SoapUI project file.

FILE >> PREFERENCES >> UI SETTINGS:

- Now, we will go through the most frequently used 'UI Settings' in details.

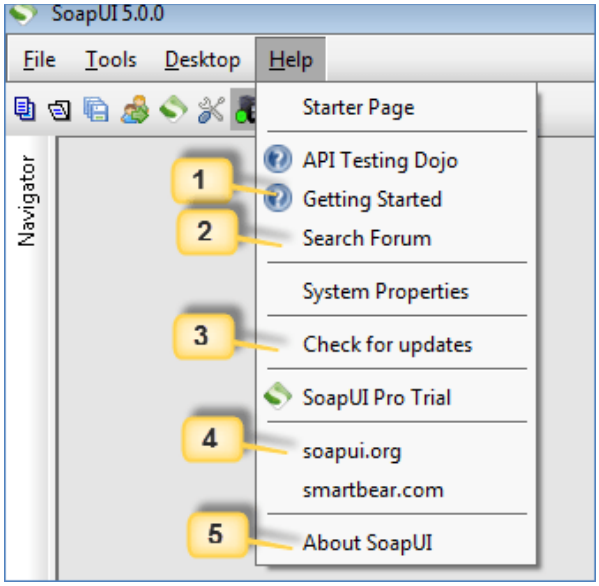


1. Closes all projects while launching SOAP UI for better startup-time and consumes less memory.
2. Displays description whenever available.
3. Automatically saves all projects while exiting SOAP UI.
4. Before saving, SOAP UI creates a backup of the project. If enabled, back up folder has to be mentioned.

5. Displays and expands the log tabs upon starting SOAP UI.
6. Displays the 'start up page' dialog upon starting SOAP UI.
7. Upon disabling tool tip, disables tool tip when user hovers mouse over the options/buttons while navigation.

HELP:

- Let us take a look at the important Help menu options.



1. Shows the home page of the online help available at www.soapui.org
2. Allows registered users to post questions in forum and get online help from the community.
3. Checks for the recent updates and installs if there it is available.
4. Allows user to navigate to the home page of www.soapui.org
5. Displays the build and version information of the SOAP UI.

SOAP UI – Create a Project, Test Suite, and Test Case

Understanding the SOAP Protocol

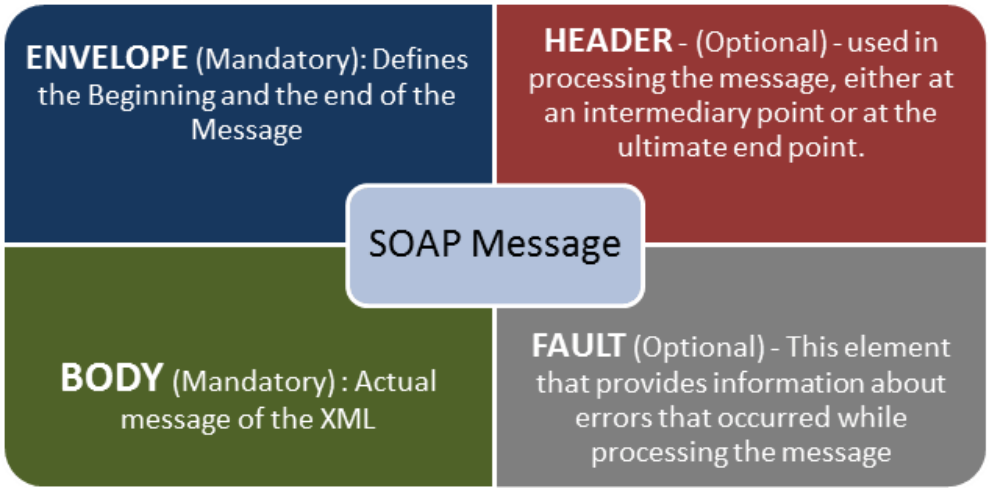
Before we create an SOAPUI Test case, let us understand basics about the SOAP Protocol. This will help you use SOAP UI to test SOAP requests and response effectively.

SOAP stands for 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.

Understanding the SOAP Message Format

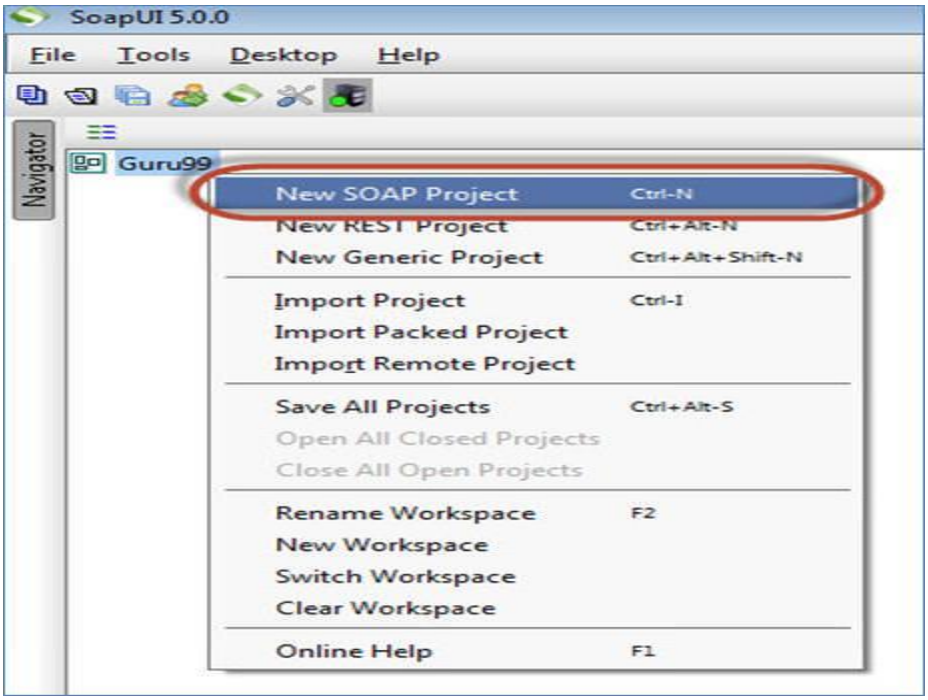
A SOAP message is an ordinary XML document containing the following elements. Message can be either a request message or a response message.



After setting up the workspace which we had performed in the last tutorial, we have to create projects, test suites, test cases in order to test a given web service. Let us understand the steps involved in doing the same.

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.webservicex.net/CurrencyConvertor.asmx?WSDL

1. Enter the Project Name
2. Enter the path of the WSDL request. In this case

   http://www.webservicex.net/CurrencyConvertor.asmx?WSDL

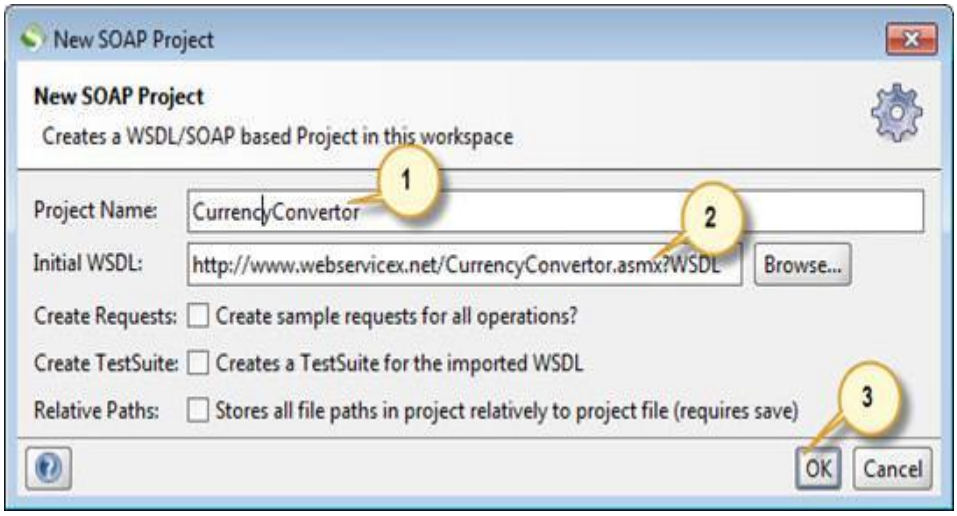
3. Click OK

Note:

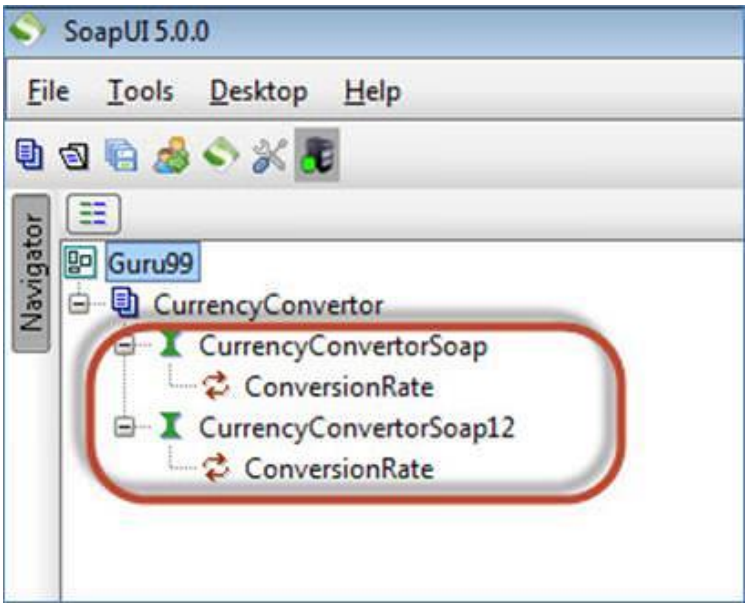
- Upon checking – Create sample request for all operation' creates a sample request for all the available operations in the given WSDL. As soon as you

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.

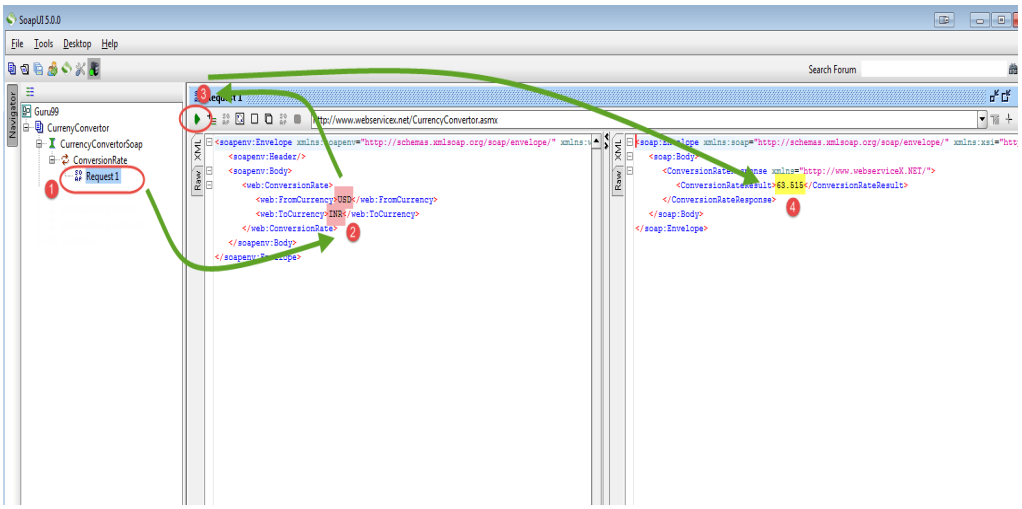


Step 3: Upon creating the SOAP project with the above-said WSDL, we will be able to see that there are two operations that will be imported into the project.



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 From Currency and To Currency
3. Click on the submit button
4. Response XML will be displayed right side pane.



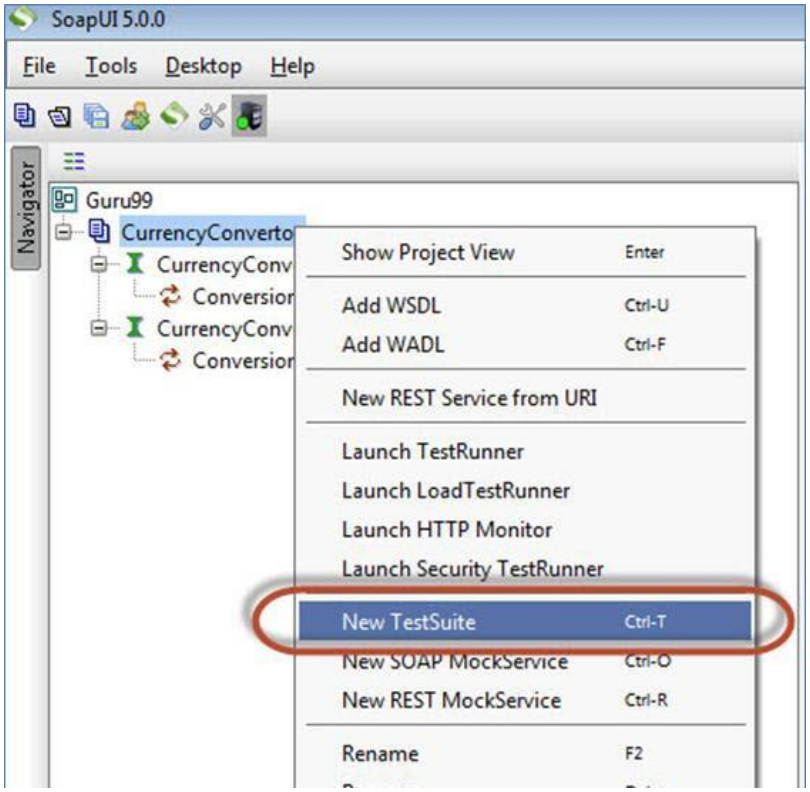
You may wonder why create Test Cases? When you can directly test WebService here…

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

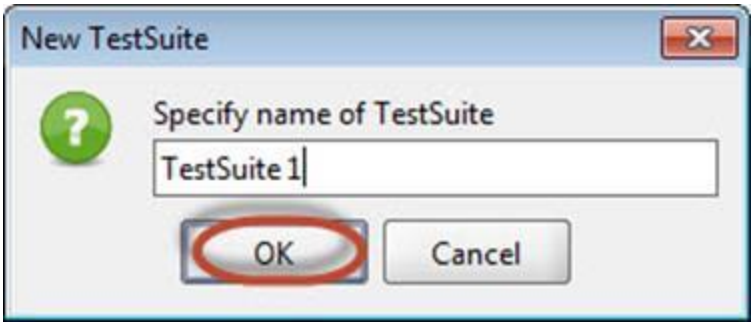
For example: If you want to convert from GPB to INR instead of USD to INR… You need to edit the operation again. So, one has to create a test suite/cases to have all possible scenarios tested without having to directly edit the operation itself.

Creating Test Suite

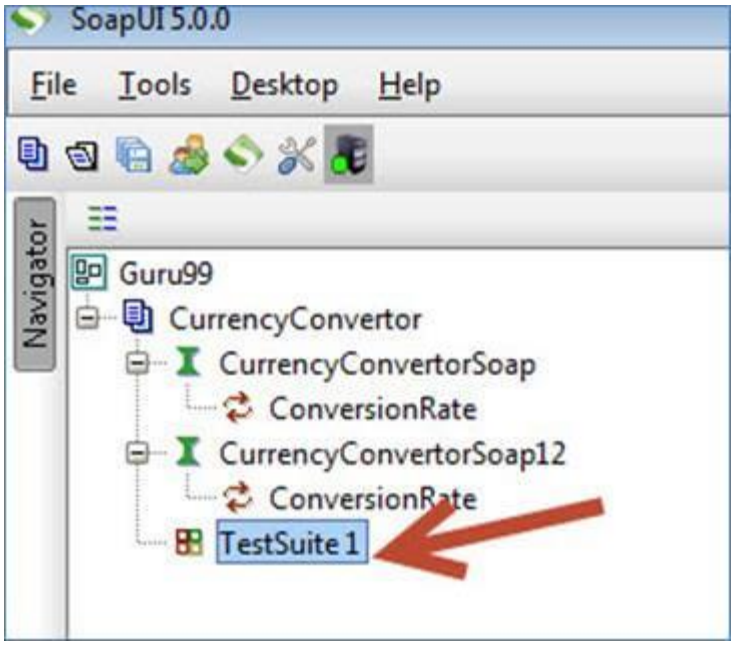
Step 1: Within the project, testers can create a test suite by performing a right click on the root of the project.



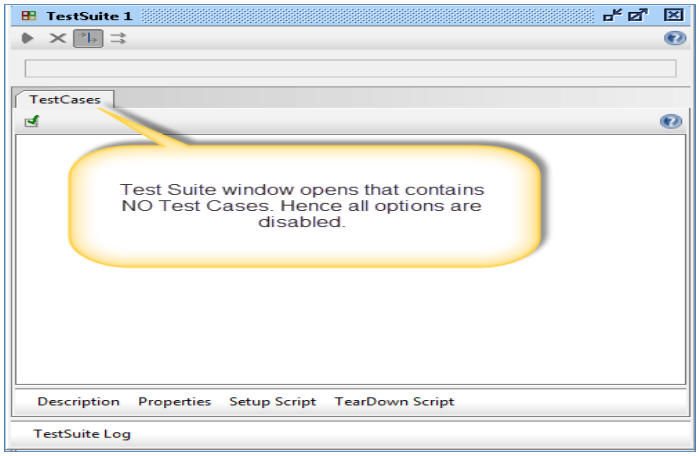
Step 2: We need to enter the name of the test suite and press OK.



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

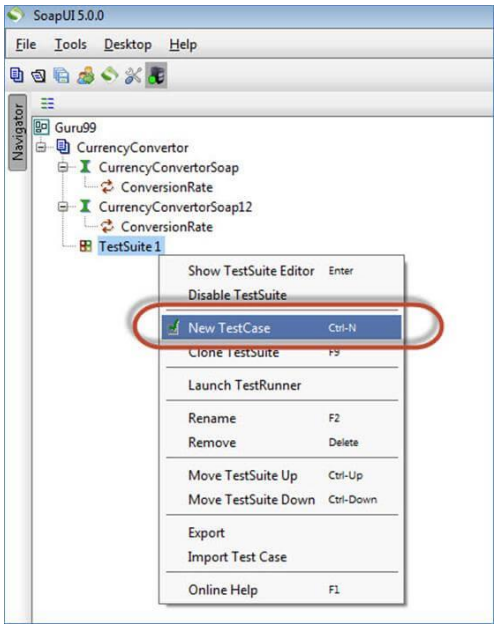


Step 4: The test Suite window opens in the Right Pane. As we have just created there are NO test cases. Hence all options are disabled.

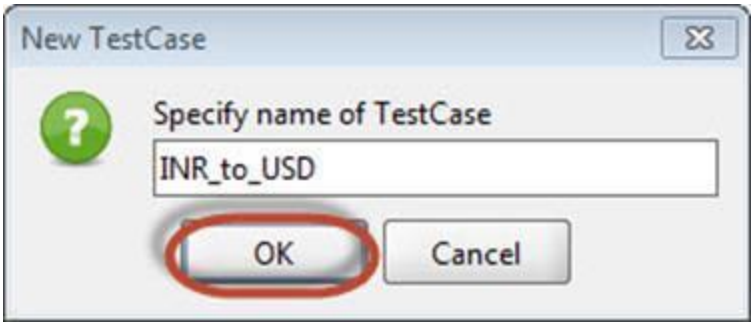


Creating Test Case

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

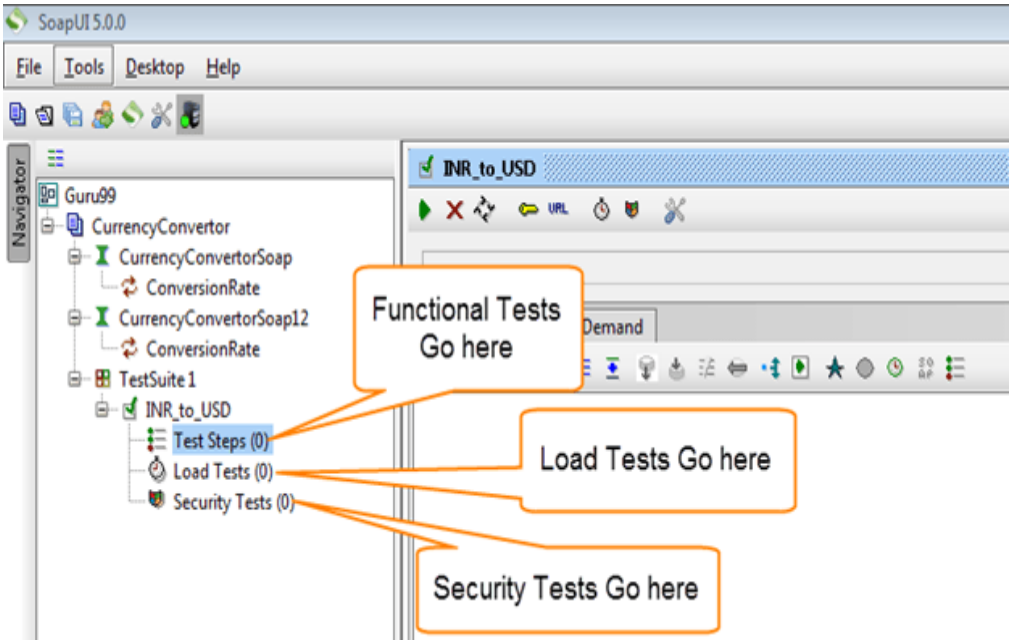


Step 2: Specify the name of the test case and click 'OK'.



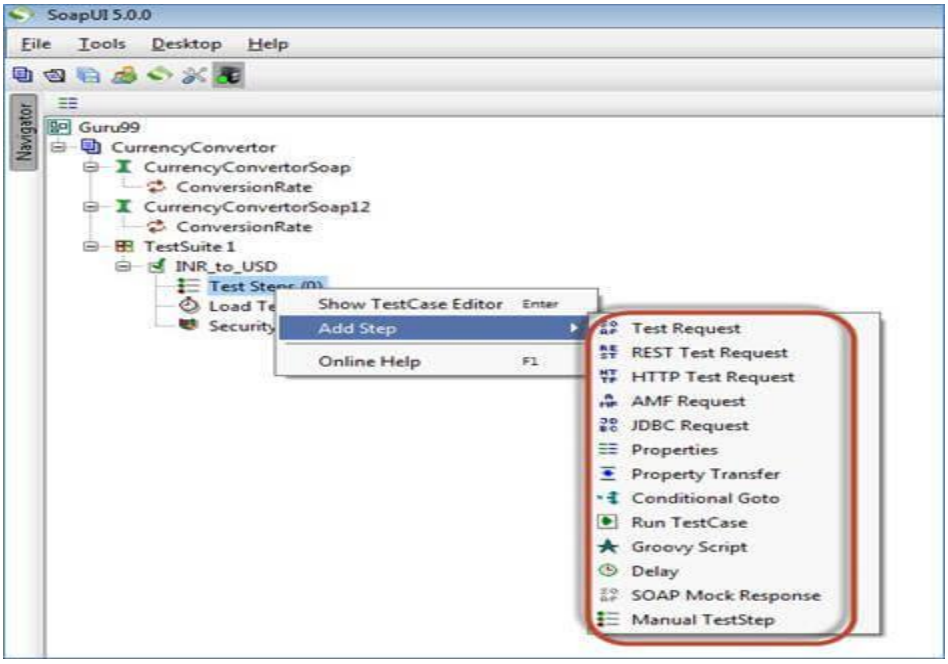
Step 3: The created test case has zero steps as shown below.

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



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'.

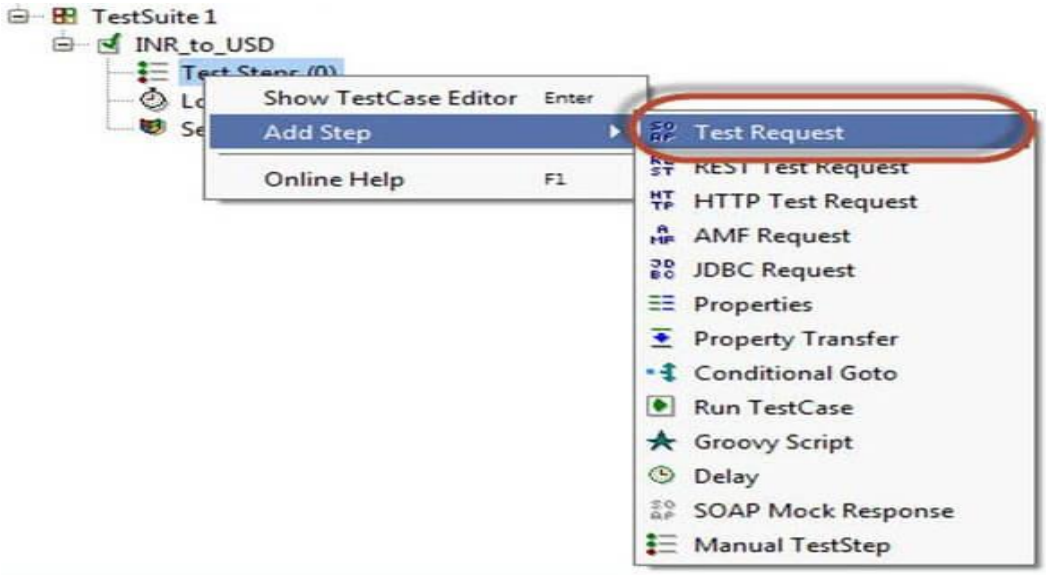
Step 4: We can insert a variety of test steps by performing a right click on test steps and selecting an appropriate test step as shown below. So if you were to test a REST Web Service, you would select the REST Test Request.



Insert Test Step

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

Step 1: Add a new step 'SOAP Test Request' as shown below.



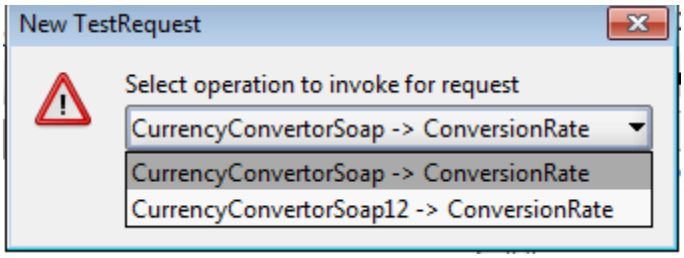
Step 2: Enter the step name and click OK.

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.

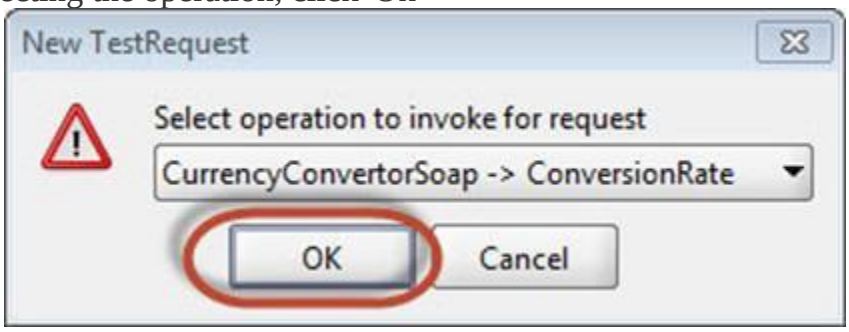
- There are two operations that will be listed. Both the Operations are the same except the SOAP version used.
  CurrencyConvertorSoap – uses SOAP version 1.1 whereas,

CurrencyConvertorSoap12 – uses SOAP version 1.2

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



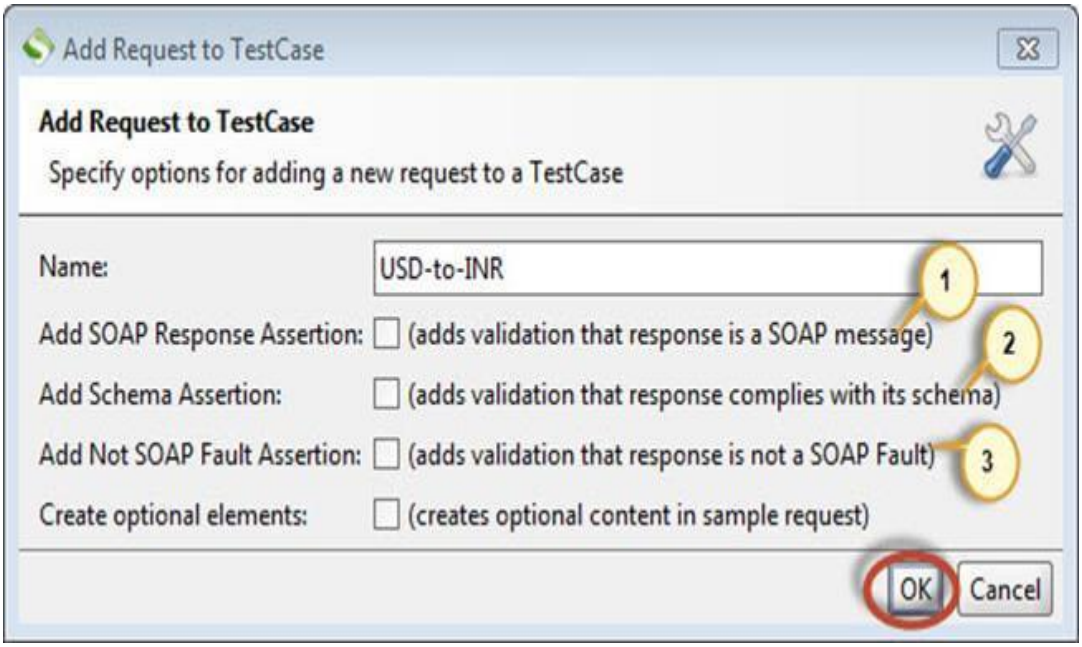
- Upon Selecting the operation, click 'Ok'



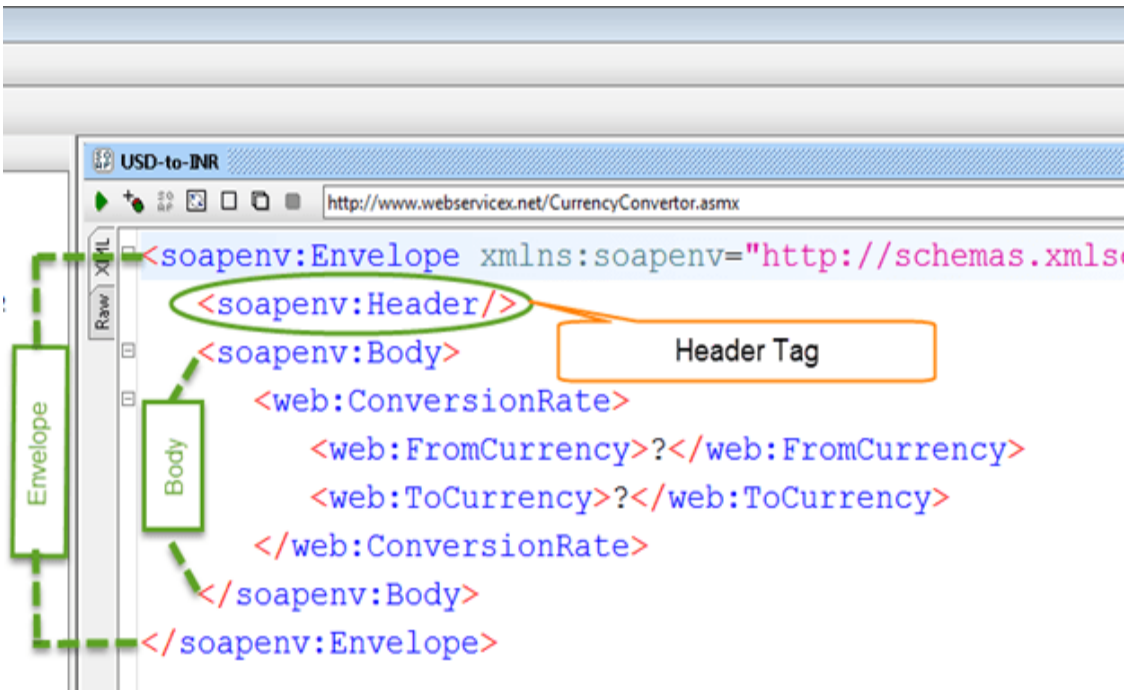
Step 4: While adding a test case, we can add standard assertions. Assertions also called as checkpoints/validation points which we will be dealing in detail in the next tutorial.

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

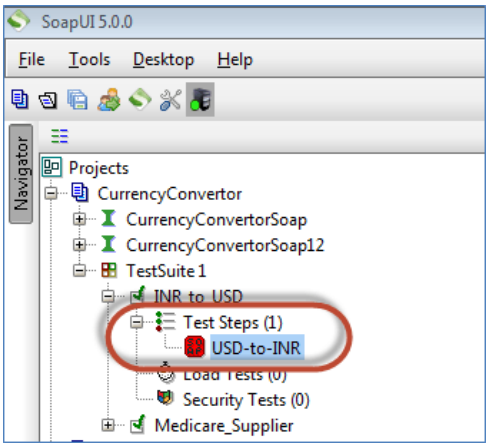
- Verifies if the response message is SOAP, upon executing the test.
- Verifies if the response schema is valid.
- Verifies if the SOAP response contains FAULT.



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

Step 1: We would like to convert the currency from USD to INR.

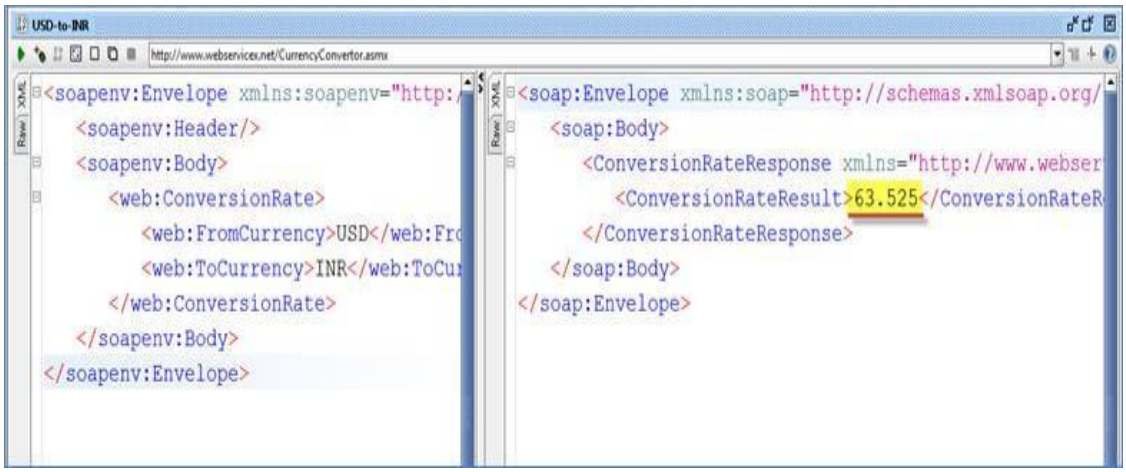
- FromCurrency – USD
- ToCurrency – INR

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.

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 that 1 unit of USD = 63.525 units of INR.

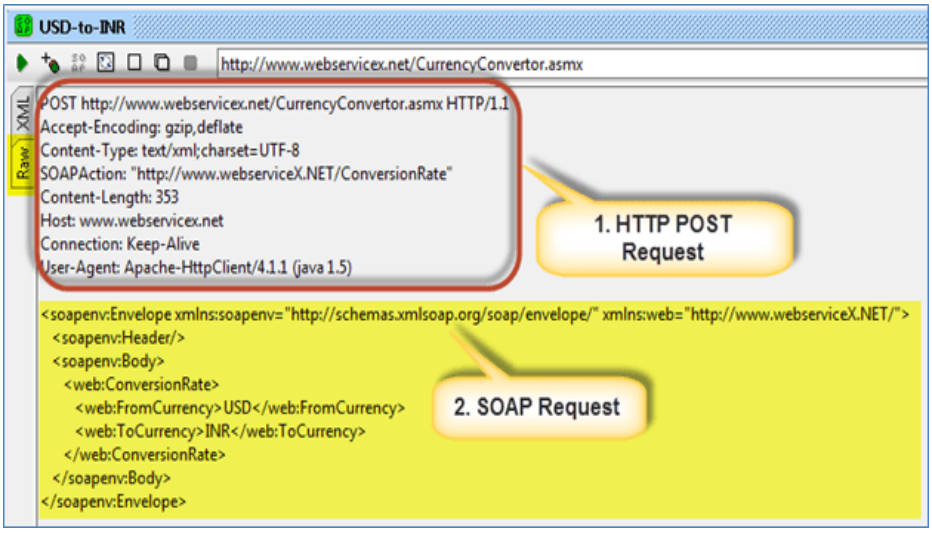


Understanding the Soap Response & Log Panels

As explained at the beginning of this tutorial the SOAP messages are transported via HTTP 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.

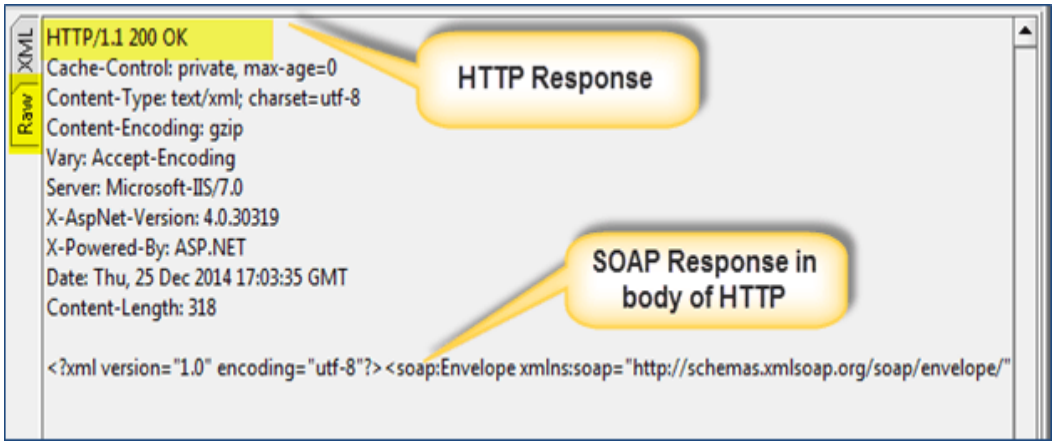
Step 1: Click 'RAW' Tab in both SOAP-UI request Window.

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 as shown below



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.

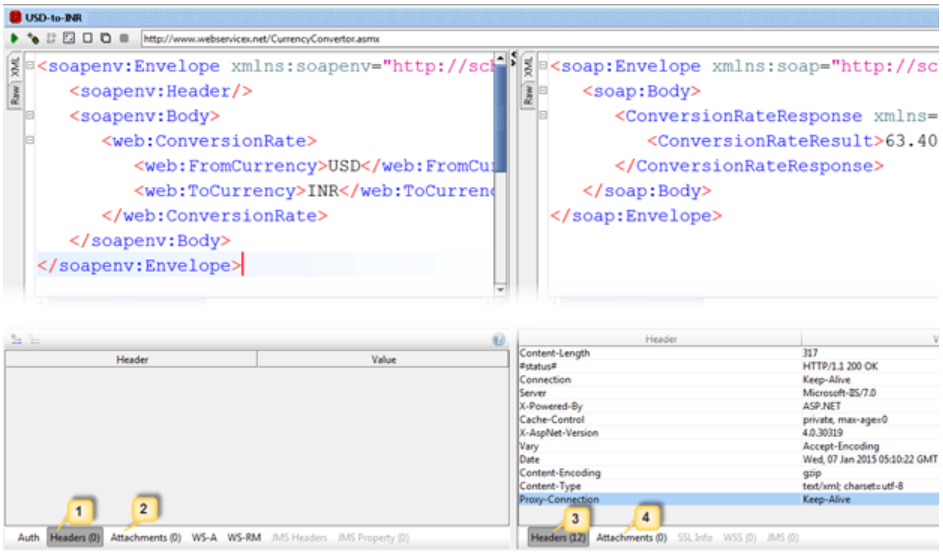


A Quick snapshot of the Http Response codes for easy understanding and debugging. The below table will help you to trouble shoot based on the HTTP code received from the webserver.

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

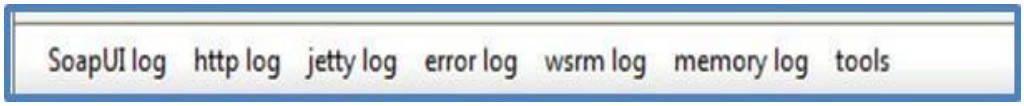
Step 3: Let us understand the other information that are displayed in the test case window.

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 12 header information and the same are displayed upon clicking on it.
4. Represents that there are no attachments from the response message.

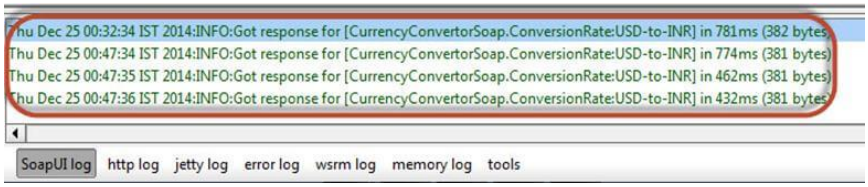


LOGS PANE:

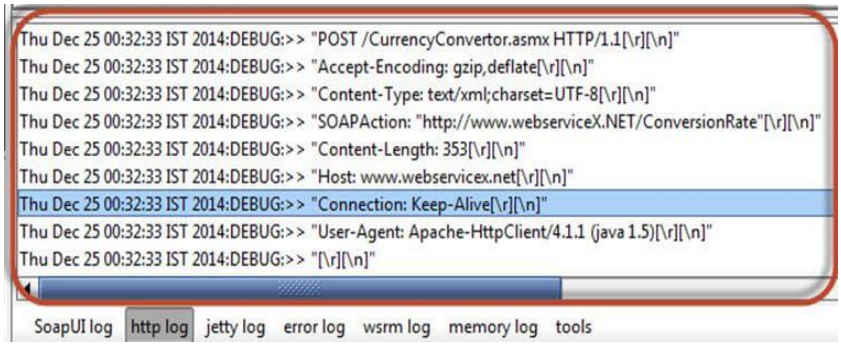
Logs pane has the complete information regarding the transaction between the client and the server. Users will be able to see the tabs of the Log pane as shown below. We will discuss the most commonly used log panes when working with SOAP-UI.



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.

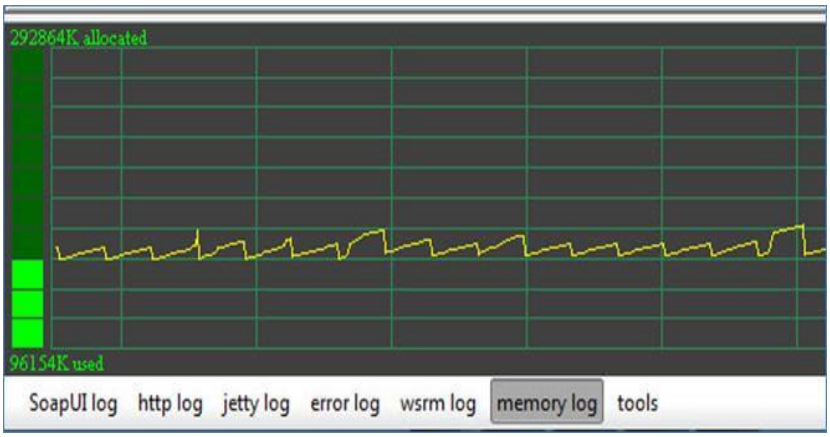


Http Log – Displays all the HTTP packet transfer. All the information in 'RAW' is shown in HTTP log.



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.



Now that we have created test suite, test case, test step and got a response, next step is to validate the response.