**Creating a TestSuite**

**What is a Test Suit in SOAP UI?**

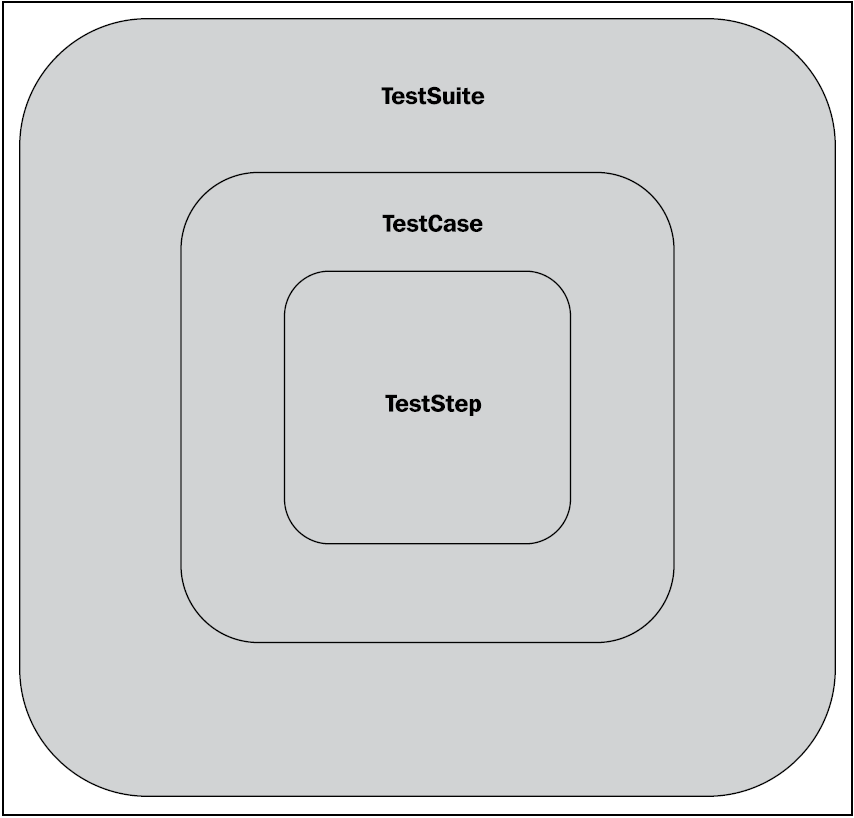
soapUI **TestSuites** are one of the key building blocks in a project which allow us to **structure and execute functional tests.**

A soapUI functional test consists of three key elements as follows:

• **TestStep**: A TestStep is the foundation of any functional test. It is used to manage the execution flow of the test and validate the test results. A TestStep is directly associated with a TestCase.

• **TestCase**: In a soapUI project, a TestCase is a collection of TestSteps organized for testing some functionality of the service under test.

• **TestSuite**: A TestSuite is a collection of TestCases which work together as a logical unit to test some specific functionality.



**Adding test assertions**

* Assertions allow users to validate the responses by comparing some properties of the message with expected values. In soapUI, **assertions are applied to TestSteps.**
* You can add any number of assertions to a TestStep. After the TestStep is executed, all of the associated assertions are applied to the response. The TestStep is failed if any of the assertions fail.

Assertions Feature

Types :-

**Contains Assertion**  Looks for a pattern starting from the extreme left top end corner cursor. It can search for any part of a doc. Can be added multiple times

**Not Contains Assertion**  Exactly opposite to Contains. Can be added multiple times

**XPath Assertion:**

It does the same for XML what path expressions do for OS to search for a file in directory. It is used to search for a data element or group of XML tags in XML. It contains two windows.

***Xpath Expression*** : - Clicking on ‘Declare’, all the name spaces gets declared. Here we mention the Xpath expression against which value is expected and to be validated.

***Expected Result*** :- Here the expected content within that Xpath is put.

***Select From Current*** :- Clicking on that populates the Expected Result window with the result from the received response corresponding to the Xpath.

***Test*** :- If the Xpath is present & value in the Expected Result window matches with the one received in the response, then on clicking on ‘Test’ shows the success message. Otherwise

**Note:**

* When you specify the XPath expression as shown in the preceding screenshot, make sure to declare any namespace prefix which you use in the expression.
* All namespaces must be declared before they are used in the XPath expression.

**SOAP Response assertion:**

Validates if the received response is SOAP compliant. If yes, then green. It is added at the time of adding a test step.Can be added one time

**Not SOAP Fault assertion:**

It is use to check whether it is a SOAP Fault or a valid SOAP response. If the response is a SOAP Fault, the TestStep will be marked as failed.

Schema Compliance :- Checks if the received response is in accordance with the schema of the WSDL. If yes, then green.Can be added one time

SOAP Fault :- If the received response has any SOAP fault then it is green.Can be added one time

Valid HTTP Status Codes :- Checks if the received response’s HTTP code is what is put as a part of the Assertion. If yes, then green.Can be added one time

Invalid HTTP Status Codes : - Checks if the received response’s HTTP code is not what is put as a part of the Assertion. If yes, then green. Can be added one time

SLA Response :- Checks if the response is received within a specified time(in milliseconds) as mentioned in the SLA Response Assertion. If received within, then green.Can be added one time

Sensitive Information Exposure :- Behaves exactly same as Not Contains, but here multiple items can be validated as a part of single assertion unlike Not Contains.Can be added one time