



InterviewBit

Soap UI Interview Questions



To view the live version of the page, [click here](#).

© Copyright by Interviewbit

Contents

Soap UI Interview Questions for Freshers

1. What do you understand about web services? State some of the communication channels which are available for a web service.
2. State your understanding of SoapUI automation.
3. How can we use SoapUI for the testing of Restful Web Services?
4. What are the advantages of using SoapUI? How does SoapUI structure functional tests?
5. In SoapUI, what is the most basic approach for automating web services?
6. In SoapUI, how do you test an ASMX web service?
7. In SoapUI, how do you test an SAP Web Service?
8. How can we use SoapUI to test a WCF rest service?
9. What are assertions in SoapUI? State some of the important assertions present in SoapUI.
10. State and explain the default properties in SOAPUI.
11. State a few important functionalities of SOAP UI.
12. What are the steps to parameterize endpoints in SOAP UI?
13. What do you understand about XPath Assertion in SoapUI? Also, state your understanding of Data Driven Testing.
14. State the steps to add assertions in SoapUI.
15. What do you understand about WSDL?
16. Differentiate between SOAP and REST.

Soap UI Interview Questions for Experienced

17. Can we use SOAPUI to do API testing? If yes, then state the steps to do the same.
18. How can we use SoapUI to test web services?

Soap UI Interview Questions for Experienced

(.....Continued)

19. Can we do automation for the test cases using SoapUI? If yes, then how?
20. How can we create and import Test Suite in SoapUI?
21. In SoapUI, how can we do the reading of data from excel spreadsheet using a Groovy Script?
22. Write a Groovy Script to save request and response in SoapUI?
23. How to stop a Groovy Script in SoapUI?
24. How can we set a property value in SoapUI using Groovy Script?
25. Using Groovy Script, get the test case name in SoapUI.
26. How can we use SoapUI to import an XSD (XML Schema Definition) file?
27. Which service in SoapUI helps us to call the SoapUI mock service from Java?
28. Can we classify SOAP as a RPC (Remote Procedure Call)?
29. What are the options to debug Groovy Scripts in SoapUI?
30. What do you understand by API Testing? Why is it needed?
31. Write a few differences for the following: SoapUI vs Postman.

Let's get Started

What is SOAP UI?

SoapUI is a [web service](#) testing application for the Simple Object Access Protocol (SOAP) and the Representational State Transfer (REST) Protocol that is free to use. Web service inspection, invocation, development, simulation and mocking, functional testing, load and compliance testing are all covered by its capabilities. Eviware Software AB also developed a commercial version of SoapUI, SoapUI Pro, which focuses on features that increase the productivity of the testers. Eviware was purchased by SmartBear Software in 2011. In September 2005, SoapUI was first released on SourceForge. It is open-source software that is licenced under the European Union Public License. The user interface is designed entirely in Swing and is totally based on the Java platform. SoapUI is therefore cross-platform.



SoapUI

SoapUI now includes support for IDEA, Eclipse, and NetBeans. SoapUI can test SOAP and REST web services, JMS (Java Message Service) and AMF (Adobe's ActionScript Messaging Format), as well as HTTP (HyperText Transfer Protocol), HTTPS (HyperText Transfer Protocol Secure) and JDBC (Java Database Connectivity) call. Some of the key features of SoapUI are as follows:

- Development of web services
- Invoking of web services
- Inspection of web services
- Simulation and Mocking of web services
- Functional, Compliance and Security testing of web services

Impact And Usage Of SOAP UI:

SoapUI has been downloaded over 2,000,000 times since its initial release. SoapUI's invention is being used by a number of Fortune 1000 companies around the world to suit their consumers' needs. SoapUI is used in a variety of businesses.

What You Can Find In This Article?

Our page provides you with nitty-gritty information such as SoapUI prospective employee meeting questions and answers in order to have a terrific development in SoapUI work. These interview questions and answers are extremely valuable for anyone looking for a new challenging job with a reputable organisation, whether they are a fresher or an experienced professional.

Now that we do have a brief understanding of what is Soap UI is and what its features are, let us dive deep into this article and look at some of the most commonly asked one's.

Soap UI Interview Questions for Freshers

- 1. What do you understand about web services? State some of the communication channels which are available for a web service.**

Web services are web components that allow clients and servers to exchange data. The client submits a web request to the server, which is subsequently responded to by the server. The response and the request are linked, and different requests elicit different responses. A web component or software program that can be accessed through the Internet is known as a web service. Through XML communications ideas, it is mostly used to communicate with web-based applications. For example, if we wish to utilize Google Maps to find a specific location, we can use the relevant web service's URL. To do so, we must provide the necessary inputs.

In general, web services are used in conjunction with the protocols listed below:

- **HTTP / POST:** The HTTP / POST protocol is used to securely send data between clients.
- **HTTP / GET:** Clients can display sent data partially in the browser's address bar using the HTTP / GET protocol.
- **SOAP:** SOAP is a protocol for securely transmitting private data.

These channels will be utilised to communicate with customers when exposing web services.



2. State your understanding of SoapUI automation.

SOAP or **Simple Object Access Protocol** is a protocol that uses XML to interact with web applications. With the help of XML based content, it can make communication happen between two client machines across any network. SoapUI is a cross-platform functional testing solution that is free and open source. It allows us to develop and run automated regression, compliance, functional, and load tests quickly and easily. SoapUI is currently being used to test Web Services. SOAP UI automation refers to the use of SOAP UI software for automation testing and implementation.

3. How can we use SoapUI for the testing of Restful Web Services?

In order to use SoapUI to test restful web services we can do the following: Using SoapUI to test restful web services requires first developing test cases, after which the following actions can be followed.

- The next step is to make a REST Request. By using the "Add to TestCase" button (found on the top left of the message editor) or the "Add to TestCase" action from the Request Popup
- Menu, TestStep can be added to a new or existing TestCase.
- After that, we must include Assertions in order to validate the response message.

4. What are the advantages of using SoapUI? How does SoapUI structure functional tests?

The **advantages of using SoapUI** are as follows:

- SoapUI can run load tests (Load testing is a type of non-functional software testing in which a software application's performance is evaluated under a certain load which determines how the software application operates when numerous people access it at the same time. Load testing is used to identify performance bottlenecks and ensure the stability and smooth operation of software applications prior to deployment), scenario-based tests (Scenario-based testing is one way to document the project's software specs and needs. Scenario-based testing is a method of designing tests for individual user scenarios in order to ensure that their job is correct. Scenarios focus on the most important goals and requirements. The scenario passes if it runs from beginning to end. Scenario-based software testing may be described as an inspection-based collection of tests that are run to ensure that the user can execute the specified actions from start to finish. There are alternative ways to depict scenarios, such as a table with the first column describing the user's actions and the second describing the system's response), and data-driven tests (Data-driven testing (DDT), also known as table-driven testing or parameterized testing, is a software testing methodology that describes testing that uses a table of conditions directly as test inputs and verifiable outputs, as well as the process of not hard-coding test environment settings and control. In its most basic version, the tester provides inputs from a table row and anticipates outputs from the same row. Values that correspond to border or partition input spaces are often found in the table. The test configuration is "read" from a database in the control approach) automatically.
- It has the capabilities to generate reports for the tests performed using it.
- SoapUI provides the unique ability to impersonate Web services and perform functional and load tests against them prior to their execution.

Functional tests are organised using three layers in SOAP UI:

- **Test Suite:** A Test Suite is a collection of test cases that are used to group functional tests into logical units.
- **Test Case:** A Test Case is a collection of test procedures used to evaluate a single feature of your services. Any number of Test Cases can be added to test suites.
- **Test Steps:** In SOAPUI, Test Steps are the "building blocks" of functional tests. They work together to create Test Cases, which determines the service's execution flow.

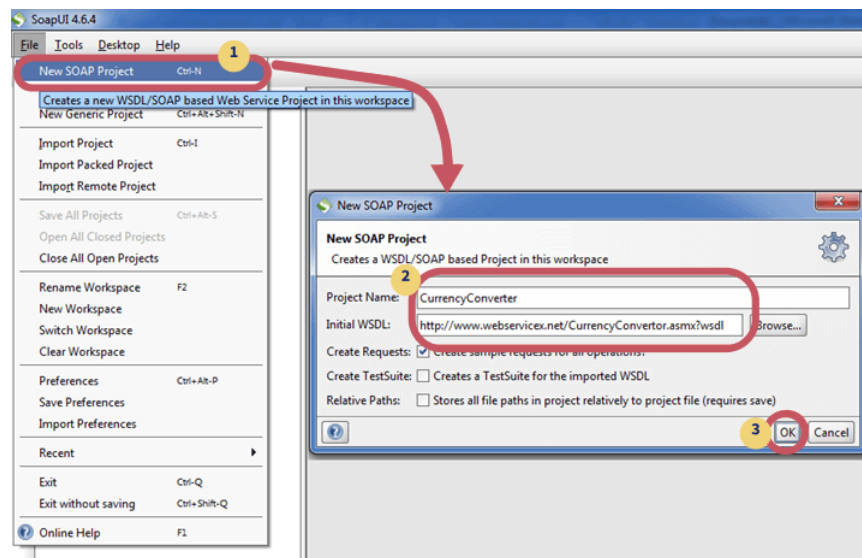
5. In SoapUI, what is the most basic approach for automating web services?

The most basic approach for automating web services in SoapUI is as follows:

- Make a project and include the WSDL (Web Service Description Language) file in it.
- Include custom programming/validation utilising Groovy steps by adding test suites, test cases, and test cases in that order.
- If you are using external data sources, add assertions if necessary, and then RUN.

6. In SoapUI, how do you test an ASMX web service?

ASMX provides online services with a messaging feature based on the Simple Object Access Protocol, or SOAP. The basic files required for testing SOAP-based web services are WSDL files. As a result, SoapUI uses WSDL files to generate test requests, mock services, and assertions. The following are the steps to test an ASMX web service in soapui that requires WSDL.



- In SoapUI, select Create SOAP Project or File > New SOAP Project to start a new one.
- In the dialogue box that displays, type the ASMX URI in the Initial WSDL area.
- Click OK to accept the default settings.

7. In SoapUI, how do you test an SAP Web Service?

A web service is a modular and self-describing function or service that is supplied over HTTP. For service description and availability, web services use XML and other standards, as well as other internet or HTTP protocols. The WSDL files are essential for testing SOAP-based services. Test assertions, test requests, and mock services can all be described using WSDL files. Steps to test an SAP Web Service using SoapUI, which requires a WSDL file are as follows:

- In SoapUI, select Create SOAP Project or File > New SOAP Project to start a new one.
- In the dialogue box that displays, type the SAP URI in the Initial WSDL area.
- Click OK to accept the default settings.

8. How can we use SoapUI to test a WCF rest service?

For a constantly accessible service on an IIS (Internet Information Services) web server or just a service in an application, WCF gives the ability to send asynchronous messages between service endpoints. WCF (Windows Communication Foundation) is a core framework that allows you to create services with a web endpoint and deliver data using XML or JSON without using a SOAP wrapper. The steps to use SoapUI to test a WCF rest service are as follows:

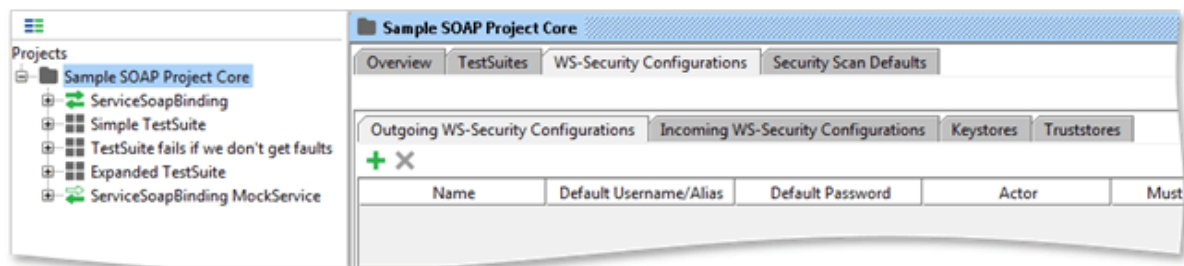
- Clicking on File -> New SoapUI Project will start a new project. Fill in the blanks in the dialogue box, and do not forget to include the Initial WSDL parameter by adding the WCF URI.
- A new SoapUI project is created after clicking Ok.
- Then, much like the WCF Test Client, make the identical request.

9. What are assertions in SoapUI? State some of the important assertions present in SoapUI.

In SoapUI, assertions are used to validate the message that a Test Step receives during execution, usually by comparing sections of the message (or the full message) to an expected value. A sample Test Step can include any number of assertions, each validating a particular element or content of the answer. The parts/all of the answer responses are compared to the expected outcome in assertions.

Some of the important assertions present in SoapUI are as follows:

- **WS security status:** SoapUI manages WS Security configurations at the project level, allowing them to be utilised in several locations within a project:
 - For incoming and outgoing requests, as well as responses.
 - Incoming MockServices requests and their MockResponses.
 - In the SOAP Monitor, for monitored requests and their monitored answers.



On the project level, there are four sub-tabs under the WS Security Configurations tab:

1. **Outgoing WS-Security Configurations:** configurations for outgoing messages, such as requests and MockResponses. Encryption, signing and adding SAML, timestamp, and username headers are all done with this configuration type.
 2. **WS Security Configurations for Incoming Messages:** configurations that should be applied to incoming messages, such as responses, MockRequests, or monitored requests and responses. This configuration type is used to decode and validate incoming messages' signatures.
 3. **Keystores:** Encryption, decryption, and signature keys are stored in keystores.
 4. **Truststores:** Truststores used for signature verification.
- **XPath Match:** The XPath assertion applies a defined XPath expression to the incoming message and compares the nodes to a predetermined value. The assertion succeeds if the values match; else, it fails. Let us take a look at a login response message that we would like to verify:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sam='
  <soapenv:Header/>
  <soapenv:Body>
    <sam:loginResponse>
      <sessionid>12406604040526722</sessionid>
    </sam:loginResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Essentially, we want to look for the loginResponse and sessionId components in the SOAP Body, but ignore the sessionId's actual value because it will change between requests.

- **XQuery Match:** To pick information from the target property, XQuery employs an XQuery expression. It compares an XQuery expression's result to an expected value and it is applicable to any property containing XML.
- **Script Assertion:** The Script Assertion enables arbitrary message validation, such as message content, HTTP headers, and so on. Validation scripts are written in the project scripting language (Groovy or JavaScript) and run when the assertion status of the containing sampler Test Step changes.
- **Simple Contains:** It searches for the existence of a string in the property value and supports regular expressions. It is applicable to any property.
- **Soap Faults:** Soap Faults check whether the response is a SOAP Fault (used for negative testing).
- **Not Soap Faults:** Not Soap Faults check whether the response is not a SOAP Fault.
- **Response SLA:** Response SLA checks the response time against a particular value which is given.
- **WS Addressing Response:** WS Addressing Response validates the availability of WS Addressing Headers in the response
- **Schema Compliance:** Schema compliance validates the response message against the WSDL (Web Service Description Language) definition and the XML Schema it contains.
- **Simple not Contain:** It searches for the non-existence of a string in the property value and supports regular expressions. It is applicable to any property.
- **SOAP Response:** Verifies that the response is a genuine SOAP message; used to ensure that we are receiving a response (if no assertions are added a connection error will not cause a failure).
- **JMS Timeout:** When utilising a JMS endpoint with a response channel provided, this assertion verifies that the response is received within the specified time.
- **JMS Status:** This assertion verifies that no JMS related issues occurred when using a JMS endpoint.

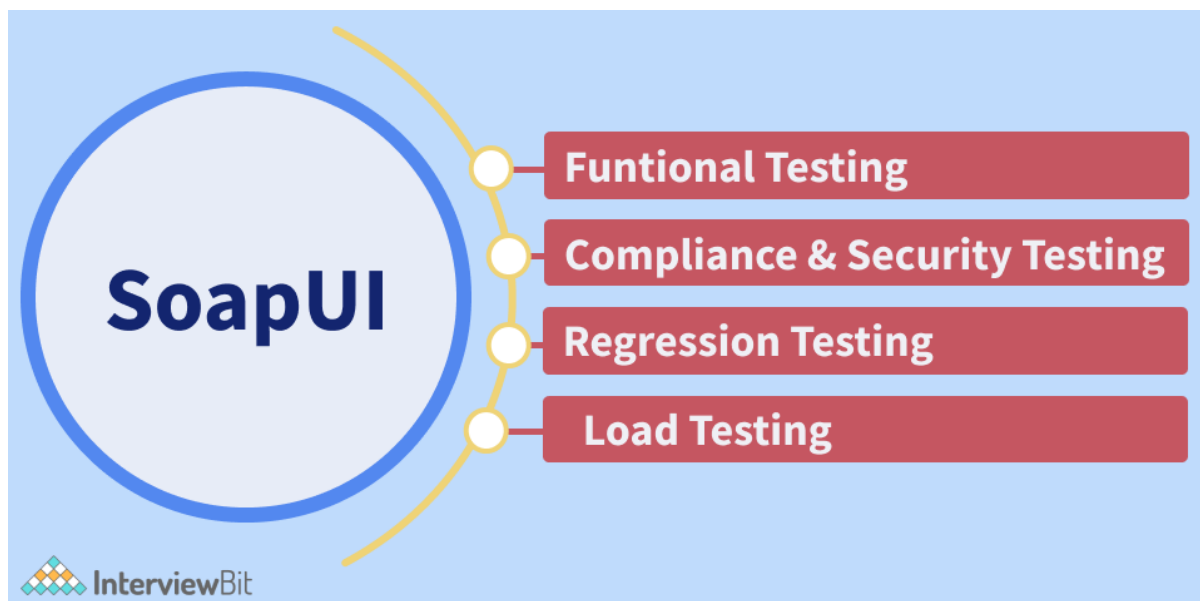
10. State and explain the default properties in SOAPUI.

SOAPUI provides three layers of properties by default:

- **Project level default or custom properties:** The user can add any number of properties at the project level, and it can be opened from any test step in any test case under any test suite.
- **Test Suite level default or custom properties:** Any number of characteristics can be appended at the test suite level, and they can be utilised from any test step in any test case under this test suite.
- **Test Step level default or custom properties:** Users can add any number of properties to a test case at the test case level, and these properties can be utilised in any test step from this test case.

11. State a few important functionalities of SOAP UI.

A few important functionalities of SOAP UI are as follows:



- It builds a new project-based just on the web service WSDL link; it retrieves all method information and imports it automatically.
- It generates test cases from the web method request itself.
- It evaluates web techniques individually or in combination. It includes a very important feature that allows you to transmit data and arguments from one method to another. It allows you to preserve data in variables so that it may be accessed by other methods.
- It allows you to use assertions to validate web method results; assertions may be written in either Xpath or Xquery.
- It aids in the creation of various validations such as schema compliance, SOAP fault, response speed, and so on.
- It allows you to organise your test cases into test suites.
- It allows you to execute all of the test cases in a test suite and get fail/pass results for each one.
- It allows parameterization of test case request fields, allowing a test case to access various values from a table or database.
- It is possible to create complicated validation scripts using Groovy.
- It has some load testing features for web services.
- It has various features to help with testing coverage and requirements management.

12. What are the steps to parameterize endpoints in SOAP UI?

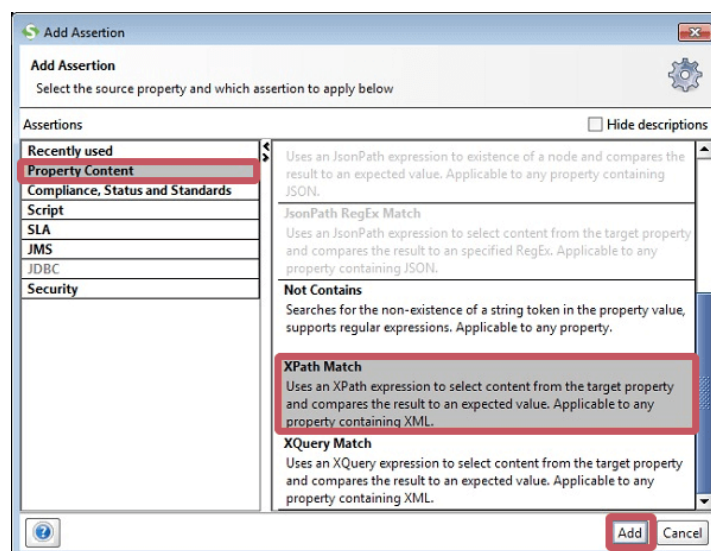
It is critical to parameterize the endpoints in SOAP UI because it is the initial step in Automation Testing. Manually updating endpoints is more time demanding. In order to accomplish this, we can do the following steps:

- Create a project property that stores the endpoint.
- Using property expansion, change the endpoint to use this property.
- Ascertain that your requests use the configured endpoint.
- The property will invariably be with its current value when a request is run. Simply configure the endpoint in the User Interface (UI) to use a different value, or use the P option from the command line.

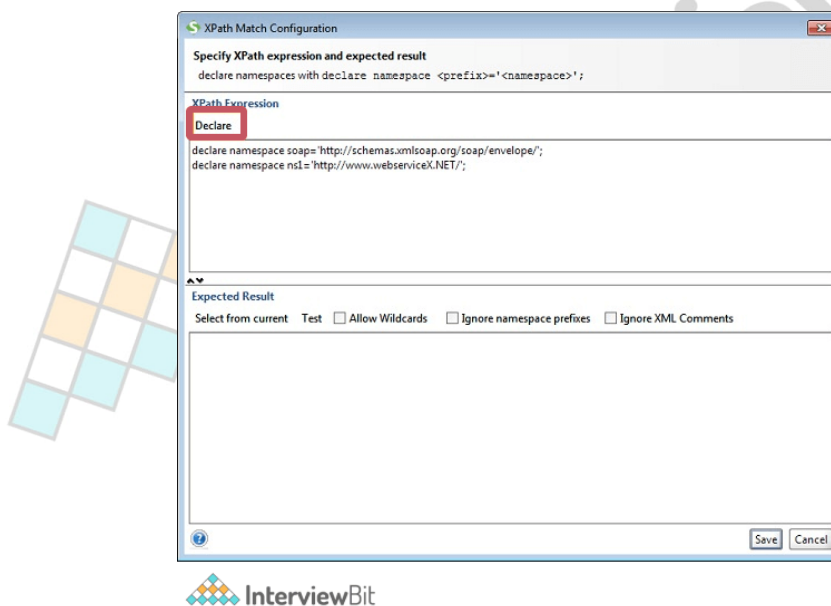
13. What do you understand about XPath Assertion in SoapUI? Also, state your understanding of Data Driven Testing.

The XPath assertion is used in SoapUI to assert the value of a web service response by specifying the absolute path. The test case or test suite will be marked as PASS if the absolute path matches the response value; otherwise, it will be marked as FAILED. The assertion results can be seen at the bottom of the screen in the Assertion tab. XPath assertion selects the target node and its values using an XPath expression. It compares an XPath expression's result to an expected value. XPath is an XML query language that allows you to pick nodes from an XML document. Given below are the steps which can be taken in order to add an XPath Assertion in SoapUI:

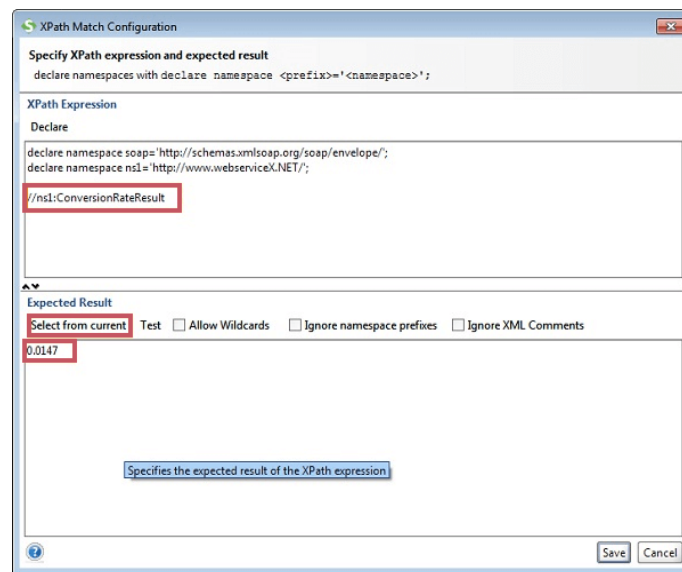
- Select Assertion Category – Property Content after clicking Add Assertion.
- Then click Add and pick Assertion Type – XPath Match. The wizard for XPath Match Configuration appears. The Namespace must be declared before an XPath may be added. An XML namespace is a set of names that are used as element and attribute names in XML documents and are identifiable by a URI reference. SoapUI XPath Assertion does the same thing.



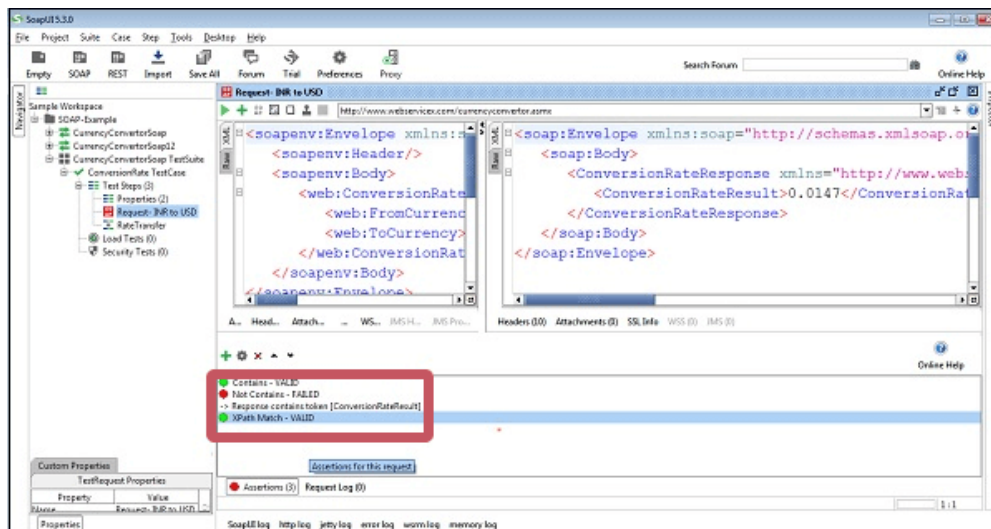
- Click the 'Declare' button to automatically declare an XML Namespace; otherwise, manually declare a namespace.
- Refer to the XPath using the newly established namespace after declaring the namespace. When you click the 'Declare' button, two namespaces with two URIs will appear. The schema URL is one of them, while the real web service URL is the other. When accessing XPath, we must use the real namespace where the web service is situated, not the schema namespace.



- Input the XPath of the XML node which needs to be checked.
//ns1:ConversionRateResult gives the value of the node enclosed between <ConversionRateResult> and </ConversionRateResult> and ns1 stand for the declared namespace pointing to 'www.webserviceX.NET'.
- After inputting the XPath, click the 'Select from current' button to take up the value from the current answer for future comparison. The user has the ability to change the value.
- Save the file.



As demonstrated in the accompanying screenshot, the added assertion will be presented.



Data Driven testing entails storing our test data, which includes both input and expected output, in an external data source such as Excel, Database, or XML file. The data source must then be iterated using the appropriate component. For data-driven testing in SoapUI, the Datasource and Datasource Loop test steps are employed.

14. State the steps to add assertions in SoapUI.

On receiving a response to a request, do the following steps:

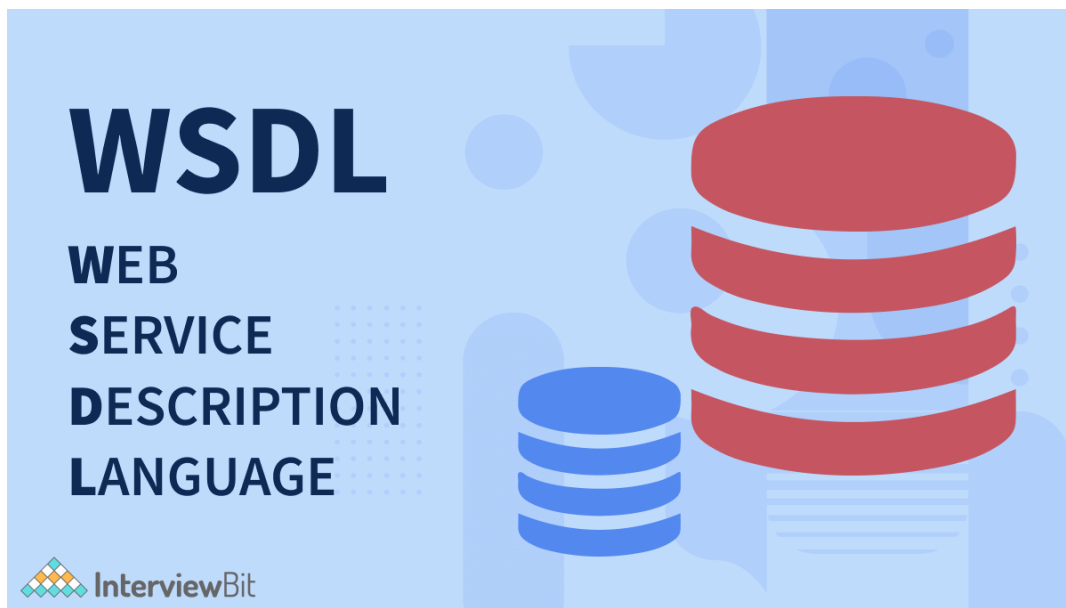
- Create a project and include the WSDL.
- Include a test suite, a test case, and a set of test steps.
- Execute the request

Now to add assertions, perform the following steps:

- Go to the top of the log tabs and click Add Assertions.
- Configure the assertions according to the data and type needed.

15. What do you understand about WSDL?

WSDL: WSDL is an acronym for **Web Services Description Language**. It is connected to SOAP. It defines how the Web service works so that you may refer to it when you make a reference to it. The Web Services Description Language (WSDL) is an XML-based interface description language for specifying a web service's capabilities. Any specific WSDL description of a web service (also referred to as a WSDL file) that offers a machine-readable explanation of how the service can be called, what parameters it expects, and what data structures it returns is also referred to by the acronym. As a result, it serves a similar purpose to a type signature in a programming language.



Services are described in the WSDL as groups of network endpoints or ports. For this reason, the WSDL definition specifies an XML format for documents. The abstract definitions of ports and messages are abstracted from their real application or instance, allowing them to be reused. A service is defined by associating a network address with a reusable binding, and a group of ports is formed by associating a network address with a reusable binding. Port types are abstract collections of supported operations, while messages are abstract descriptions of the data being exchanged. The concrete protocol and data format specifications for a specific port type comprise a reusable binding, which binds operations and messages to a specific network protocol and message format. WSDL describes the public interface to the Web service in this way.

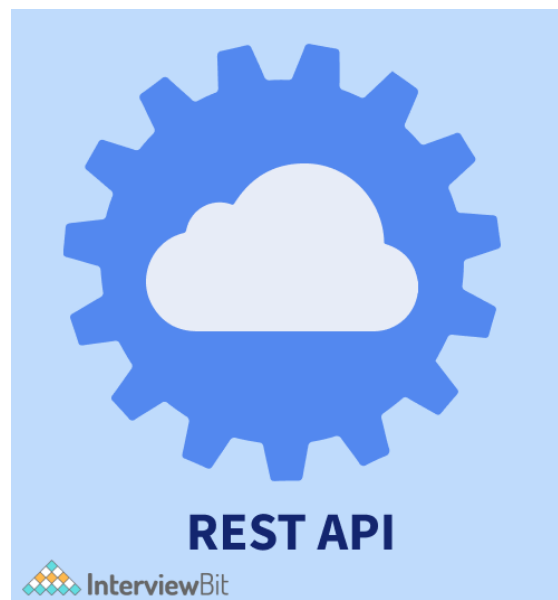
16. Differentiate between SOAP and REST.

Before differentiating between SOAP and REST, let us first look at what these are:

- **SOAP:** A SOAP API is an XML document receiver that is also expected to react with an XML document. The XML document provided should include all parameters that the receiver must be able to respond to. Due to their difficulty in integrating across enterprises, SOAP APIs have recently lost favour.



- **REST:** REST is an architectural methodology for scalable web applications that stands for Representational State Transfer. A "REST-ful" web service is one that places less emphasis on rigid formatting and instead uses JSON (JavaScript Object Notation) formatted data in message bodies rather than XML, while there is still a lot of XML in a RESTful web service.



The key differences between SOAP and REST are as follows:



SOAP	REST
SOAP (Simple Object Access Protocol) cannot make use of REST because of the fact that SOAP is just a protocol while REST is an architectural pattern.	It is possible for REST (Representational State Transfer) to make use of the underlying protocol used for the transport since REST is an architectural pattern.
SOAP stands for Simple Object Access Protocol. SOAP was created with a set of guidelines in mind. It includes a WSDL file that contains the needed information on what the web service does as well as the web service's location.	REST is an architectural approach. A web service can only be called RESTful if it complies with the constraints: Client-Server, Stateless, Cacheable, Layered System, and Uniform Interface.
The only XML format is supported with SOAP since it can work only using XML data.	REST can support various data formats, for instance, HTML, XML, JSON, etc. with the most common format being JSON (JavaScript Object Notation).
SOAP exposes its functionality to client applications through service interfaces. The WSDL file in SOAP supplies the client with the essential information to determine what services the web service can provide.	To access the components of a device, REST uses Uniform Resource Identifiers (URIs). For example, if an object representing an employee's data is hosted on http://docs.google.com , then the examples of the URIs that can be used to retrieve it. <ul style="list-style-type: none">• http://docs.google.com• http://docs.google.com

Soap UI Interview Questions for Experienced

17. Can we use SOAPUI to do API testing? If yes, then state the steps to do the same.

Yes, we can use SOAPUI to do API testing. API testing includes both unit and integration testing of APIs. API testing is similar to software testing, however, it focuses on APIs. It assesses the API's functionality, dependability, performance, and security.

The SoapUI programme allows us to test SOAP and REST web services using the following steps:

- Using SoapUI to test restful web services requires first developing test cases, after which the following actions can be followed.
- The next step is to make a REST Request. By using the "Add to TestCase" button (found on the top left of the message editor) or the "Add to TestCase" action from the Request Popup Menu, TestStep can be added to a new or existing TestCase.
- After that, you must include Assertions in order to validate the response message.

18. How can we use SoapUI to test web services?

A web service is a network-based service that allows other systems to communicate with it via a defined protocol. Using SoapUI to test restful web services requires first developing test cases, after which the same actions can be followed or the same steps can be followed, which are followed while testing any other API using SoapUI. Please refer to this [link for the steps](#) that need to be taken.

19. Can we do automation for the test cases using SoapUI? If yes, then how?

Yes, we can do automation for the test cases using SoapUI with the help of scripting. Scripts are used by SoapUI as follows:

- With the help of Groovy Script Test Step in a Test Case.
- The opening or closing of Project.
- A Pre or post-execution of a Test Case or a Test Suite is used to do continuous tasks such as initialization or cleaning up memory.
- Initializing or the stoppage of a MockService for initialization or memory clean up along with a lot of other related repetitive tasks.
- SoapUI Pro provides us with scripting support for REST and SOAP tests using a single click.

20. How can we create and import Test Suite in SoapUI?

In order to create a Test Suite in SoapUI, we can use the menu which pop-ups in the project - "New Test Suite" or a particular button on top of the array of Test Suites. In order to import Test Suite in SoapUI, one of the following options can be chosen:

- **In File Menu -> Import Project option:** For adding an existing project into the current workspace.
- **In File Menu -> Import Remote Project option:** For importing a remote project into the current workspace.
- **In File Menu -> Import Preferences option:** For importing SoapUI settings from another setting - file
- **In Icons Toolbar -> Import option:** For importing an existing SoapUI project into the current workspace.

21. In SoapUI, how can we do the reading of data from excel spreadsheet using a Groovy Script?

The DataSource and DataSource Loop functionalities are not present in the unpaid version of SoapUI. For the DataSource and DataSource Loop functionality, one can use the Ready API licensed version. One more way to read data from excel using the Groovy Script in SoapUI is as follows:

- A library can be imported to SOAPUI for the management of the excel file - JXL.jar. We can download it from the following website- http://sourceforge.net/projects/jexcelapi/files/jexcelapi/2.6.12/jexcelapi_2_6_12.zip/
- After that, the unzipping of the JAR file needs to be done followed by the copying of the JXL.jar file to the SOAPUI installation folder.

22. Write a Groovy Script to save request and response in SoapUI?

A Groovy Script for saving a request and response in SoapUI is as follows:

```
def projectName = context.currentStep.testCase.testSuite.project.name //getting the name of the project
def projectDate = new Date().format( 'yyyyMMdd' ) //getting the current date
def stringDate = projectDate.toString() //converting the projectDate into string type
def projectTestSuite = context.currentStep.testCase.testSuite.name //getting the Test Suite name
def projectTestCase = context.currentStep.testCase.name //getting the Test Case name of the Test Suite
def filePath = 'C:/SoapUI Projects/TS0.1/TestEvidence/'+projectName+'_'+stringDate+'/'
File file = new File(filePath)
if (!file.exists()) file.mkdirs() //creating the destination folder
fos = new FileOutputStream(filePath+ testStepResult.testStep.label + '.txt', true)
pw = new PrintWriter(fos)
testStepResult.writeTo(pw)
pw.close()
fos.close()
```

A groovy event handler saves requests and responses for each test step, while a script below captures the requests and responses. 'TestRunListener.AfterStep' is the event handler.

23. How to stop a Groovy Script in SoapUI?

For stopping a Groovy Script in SoapUI, we can use: `testRunner.cancel()` . It will stop the execution of the running Groovy Script. If we want to fail it, we can use

`testRunner.fail()` or try pressing the combination Ctrl+Z or the Escape key.

24. How can we set a property value in SoapUI using Groovy Script?

We can set a property value in SoapUI using Groovy Script using the following steps:

- Right-click on the Test steps node.
- From the context menu, select the Add Step and Properties options.
- Click OK after entering the desired property name.
- To add a property, go to the properties screen and click the properties with the groovy Script(1) icon.
- Click the OK button after entering the desired property name.
- If you like, you can give the property any default value.
- Following that, a Groovy Script test step should be introduced after the property step.

25. Using Groovy Script, get the test case name in SoapUI.

There are two options to achieve the above in SoapUI. They are as follows:

- Method One: In Groovy code, put the name of the testCase being run by Run Test Case (The one it directs to) into a testCase level property variable in it:

```
testRunner.testCase.setPropertyValue("Here", testRunner.testCase.getName());
```

- Method Two:

```
${=testRunner.testCase.name}
```

26. How can we use SoapUI to import an XSD (XML Schema Definition) file?

An XSD file should be imported immediately when integrating WSDL from a distant source. If you are using a local file to load the WSDL, make sure the.xsd file is in the correct location as specified in the WSDL.

The import statement in WSDL imports an XML namespace from a specified file. The namespace, not the declarations in the namespace, is imported. The element includes the "schemaLocation" attribute, which is optional. Because a namespace can be imported from the same file or registered in another method, the attribute is optional.

27. Which service in SoapUI helps us to call the SoapUI mock service from Java?

The SOAP Service Mocking facility in SoapUI using Groovy or configuring SOAP service mocking in SoapUI pro helps us to call the SoapUI mock service from Java.

28. Can we classify SOAP as a RPC (Remote Procedure Call)?

Yes, Soap is, in fact, an RPC (Remote Procedure Call). Remote Procedure Call (RPC) is a protocol that allows one program to request a service from another programme on a network without needing to know the network's specifics. Remote procedure calls in SOAP are essentially client-server interactions over HTTP with SOAP encoding rules for the request and response. From a high-level perspective, XML RPC and SOAP are extremely comparable protocols. Both use XML to encapsulate web service method requests and responses. XML RPC, on the other hand, can be thought of as a subset of SOAP capabilities. In terms of protocol, XML RPC can only make requests over an HTTP(s) connection, whereas SOAP is usually used over HTTP(s) but can also be used over other protocols.

29. What are the options to debug Groovy Scripts in SoapUI?

The options for debugging Groovy Scripts in SoapUI are as follows:

- For enabling the debugger when we run ReadyAPI using the shell script (.bat or .sh file), we can use the groovy parameter: `readyapi.bat groovy`
- For enabling the debugger by default, we can perform the following steps in Windows:

Firstly, the vmoptions file needs to be opened and then we need to add the line given below in it:

```
-agentlib:jdwp=transport=dt_shmem,suspend=n,server=y
```

The first step is followed by the saving of the file.

The following requirements must be satisfied for debugging:

- For Groovy script debugging, having a SoapUI Pro license is a must.
- The debugging of the scripts can be done only in the SoapUI Groovy Script test step. Event handler scripts, setup, teardown, and report scripts, as well as Script assertions, data source and data, sink scripts, and virtual service scripts, are not supported for debugging.
- We can only debug Groovy scripts and that too is only one script at a time.
- For installing the binary package: The binary package, unlike the ReadyAPI installation package, does not include the essential Java files. For debugging, ReadyAPI leverages the Java JDK's tools.jar and attach.dll files. You cannot use the debugger without these files, which are not included in the typical JRE installation. To enable Groovy debugging in a binary installation, you can follow the below-given instructions:
- Firstly, the installation of JDK must be done on your local computer or on a computer to which you have admin rights.
 - Windows users can refer to this link for the installation of JDK: [Link](#)
 - For mac Operating System users, this link can be referred to for the installation of JDK: [Link](#)
 - For Linux Operating System users, this link can be referred to for the installation of JDK: [Link](#)
- Use a Java JDK for getting the tools.jar file. This file can be found in the lib folder of the JDK installation folder. This is followed by copying this file to the lib folder in the ReadyAPI installation folder.
- Use a Java JDK for getting the attach.dll file. This file can be found in the JRE bin folder in the JDK installation folder. The copying of this file needs to be done to the bin folder in the ReadyAPI installation folder.

30. What do you understand by API Testing? Why is it needed?

API testing is a sort of software testing that involves explicitly testing APIs as well as part of integration testing to see if the API satisfies the application's expectations in terms of functionality, stability, performance, and security. Our main focus in API testing will be on the software architecture's business logic layer. Any software system with various APIs can be subjected to API testing. The appearance and feel of the application will not be a focus of API testing. API testing differs significantly from GUI testing. API testing involves the following:

- Unit testing
- Integration testing
- Functional testing
- and various other types of testing

API testing is required to determine whether an API meets user expectations in terms of functionality, dependability, performance, and security.

31. Write a few differences for the following: SoapUI vs Postman.

The following table brings out the differences between SoapUI vs Postman:



SOAP UI	POSTMAN
SoapUI is used for testing protocols like SOAP, GraphQL, REST, etc.	Only REST APIs can be tested using Postman.
Data-Driven Testing can be done using SoapUI.	Data-Driven Testing cannot be done using Postman.
Automation testing can be done using SoapUI.	Automation testing cannot be done using Postman. Only Manual and exploratory Testing are supported by Postman.
SoapUI is an API tool.	Postman is a Google Chrome Extension.
SoapUI supports a command-line interface.	Postman does not have a command-line interface.
Groovy Script is SoapUI's scripting language.	JavaScript is Postman's scripting language.

Conclusion:

Being one of the most amazing web service testing applications out there, SoapUI is being used by some of the really famous companies like Bank of Montreal, ARGO, Maximus Inc., etc. to test their web services. Therefore, having a thorough knowledge about SoapUI and having good practice with some of the most common SoapUI interview questions is a must for any budding talent of today who is wanting to take a job for testing services.

References and Resources:

- [SoapUI documentation](#)
- [Github: SoapUI source code](#)
- [Automation Testing Interview Questions](#)
- [Manual Testing Interview Questions](#)
- [Top 20 Testing Tools in 2022](#)
- [REST API Interview Questions](#)
- [API Testing Interview Questions](#)

Links to More Interview Questions

[C Interview Questions](#)

[Php Interview Questions](#)

[C Sharp Interview Questions](#)

[Web Api Interview Questions](#)

[Hibernate Interview Questions](#)

[Node Js Interview Questions](#)

[Cpp Interview Questions](#)

[Oops Interview Questions](#)

[Devops Interview Questions](#)

[Machine Learning Interview Questions](#)

[Docker Interview Questions](#)

[Mysql Interview Questions](#)

[Css Interview Questions](#)

[Laravel Interview Questions](#)

[Asp Net Interview Questions](#)

[Django Interview Questions](#)

[Dot Net Interview Questions](#)

[Kubernetes Interview Questions](#)

[Operating System Interview Questions](#)

[React Native Interview Questions](#)

[Aws Interview Questions](#)

[Git Interview Questions](#)

[Java 8 Interview Questions](#)

[Mongodb Interview Questions](#)

[Dbms Interview Questions](#)

[Spring Boot Interview Questions](#)

[Power Bi Interview Questions](#)

[Pl Sql Interview Questions](#)

[Tableau Interview Questions](#)

[Linux Interview Questions](#)

[Ansible Interview Questions](#)

[Java Interview Questions](#)

[Jenkins Interview Questions](#)