QAClickAcademy.com

Webservices/REST API Testing Interview Questions

1. What are Webservices ?

**Web Service is an application or business logic that is accessible using standard Internet protocols via standardized XML messaging system**

2. What are REST API’s?  
 **REST". Representational State Transfer.  
 We will use standard HTTp Methods to access the resources They are GET, POST, DELETE, and PUT**

1. What is WSDL?  
   **WSDL definition describes how to access a web service and what operations it will perform.  
   WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet**
2. What is Endpoint?  
   **It is the Ip address of the Server where Webservice is running**
3. What methods does REST Support?  
   **GET, POST, DELETE, and PUT**
4. What is SOAP UI tool?  
   SoapUI**, is the world leading Open Source Functional Testing**tool**for API Testing**
5. How to access properties from Testcase level in SoapUI?

**We can pass to the request Parmaters as ${#TestCase#PropertyName},  
If it is Groovy Script Step then TestRunner.TestCase.getPropertyValue(PropertyName)  
From Script Assertion window it can be accessed by Context.getTestCase.getPropertyValue(PropertyName)**

1. **Mention what is the general format for reading the custom or default property value?**

**The general format for reading the custom or default property value is ${#levelname#key}**

1. **What is Property Transfer Step in SoapUI ?  
   Property Transfer enables ut to transfer the values from one API Response to other API Request**
2. **What king of Scripting SoapUI Tool supports**?  
   **Groovy Scripting. Java, Java Script**
3. **How many kind of Assertions we have in Soap UI Tool?**

**Xpath Match , Contains, Script Assertion, Xquery Match,Http Status codes etc**

1. **What are Script** **Assertion window objects**?

**Log, context, messageExchange**

13.**TestRunner Usage in SoapUI**?

With **the help of TestRunner variable we can have control over the TestCases Testsuites and Project level Methods and access properties**

14 **What king of inputs and outputs does REST API Support**?  
 It supports both XML and Json

**15. What is Mocking**?

**Mock services are a great way of getting testing early into the picture of a**

**service-oriented project. Once the WSDL of the web service is ready, you can**

**simulate the service implementation and start testing the consumer applications**

All the jars in Lib directory of SoapUI

Single jar present in bin directory

Jcommander jar from Web

TesTNG jar from Web

* **WebServices  
  Web Service is an application or business logic that is accessible using standard Internet protocols via standardized XML messaging system.**
* **Web Services allows you to expose the functionality of your existing code over the network. Once it is exposed on the network, other application can use the functionality of your program**
* **Because all communication is in XML, web services are not tied to any one operating system or programming language**
* **Example:Java can talk with Perl; Windows applications can talk with Unix applications.**
* **There are specific standards to exchange the information between Webservices**

**SOAP is an XML-based protocol for exchanging information between Webservices.**

* **SOAP is acronym for Simple Object Access Protocol**
* **SOAP requests are sent via an HTTP request and SOAP responses are returned within the content of the HTTP response**
* **The basic Web services communication platform is XML + HTTP**.
* **Scenario:**
* **Application “A” bundles Employee information into a Soap Message and sends to - WebService “B” over HTTP-request**
* **Web Service “B” unpacks the SOAP request and converts it into a command that the Dontnet application can understand**
* **Next, the Web Service B packages up the response into another SOAP message, which it sends back to the Application “A” in response to its HTTP request.-response**
* **The Application “A” unpacks the SOAP message to obtain the results of the account registration process**

**Soap Protocol**

SOAP is an XML-based protocol for exchanging information between Webservices

SOAP is acronym for Simple Object Access Protocol

SOAP is a communication protocol

SOAP provides data transport for Web services

SOAP enables client applications to easily connect to remote services and invoke remote methods.

A SOAP message is an ordinary XML document containing the following elements.

**Envelope:** ( Mandatory )  
The SOAP envelope indicates the start and the end of the message so that the receiver knows when an entire message has been received. The SOAP envelope solves the problem of knowing when you're done receiving a message and are ready to process it.

* Every SOAP message has a root Envelope element.
* Envelope element is mandatory part of SOAP Message.

**Header**: ( Optional )

Contains any optional attributes of the message used in processing the message, either at an intermediary point or at the ultimate end point.  
Headers are intended to add new features and functionality

**Body**: ( Mandatory )

Contains the XML data comprising the message being sent.  
The SOAP body is a mandatory element which contains the application-defined XML data being exchanged in the SOAP message. The body must be contained within the envelope and must follow any headers that might be defined for the message.

**Fault:** ( Optional )

An optional Fault element that provides information about errors that occurred while processing the message  
When an error occurs during processing, the response to a SOAP message is a SOAP fault element in the body of the message, and the fault is returned to the sender of the SOAP message

Calculator Add Operation Soap Request

**<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:tem="http://tempuri.org/">**

**<soap:Header/>**

**<soap:Body>**

**<tem:Add>**

**<tem:a>7</tem:a>**

**<tem:b>6</tem:b>**

**</tem:Add>**

**</soap:Body>**

**</soap:Envelope>**

**Web Services**

What is a Web Service:

Web Service - service available over the web –

-enables communication between applications over the web

-provides a standard protocol/format for communication

Why we use it:

- platform independent communication

- using web services two different applications (implementation) can talk to each other and exchange data/information

How Web Services works?

SERVICE PROVIDER (Server) is the one who creates the web service and sends the response

Service Consumer (Client) is the one who sends the request to Server

In order to communicate between client and server there should be Medium and Format

Medium – HTTP/INTERNET

Format – XML/JSON

2 types of implementing web services -

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

SOAP –

Medium – HTTP (POST)

Format – XML

REST –

Medium – HTTP (POST, GET, PUT, DELETE,…)

Format – XML, JSON, TEXT

WSDL – Web Service Description Language is an XML Interface provided by Service Provider as Web Service

UDDI – Universal Description, Discovery and Integration is an XML based standard for publishing and finding web services.

Properties can be accessed at following levels:

Project - ${[#Project](https://www.youtube.com/results?search_query=%23Project)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

TestSuite - ${[#TestSuite](https://www.youtube.com/results?search_query=%23TestSuite)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

TestCase - ${[#TestCase](https://www.youtube.com/results?search_query=%23TestCase)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

TestStep - ${TestStepName[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

System - ${[#System](https://www.youtube.com/results?search_query=%23System)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

Env - ${[#Env](https://www.youtube.com/results?search_query=%23Env)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

Global - ${[#Global](https://www.youtube.com/results?search_query=%23Global)[#PropertyName](https://www.youtube.com/results?search_query=%23PropertyName)}

Groovy Script for properties –

log.info"Hello World..."

def prjname = context.expand ('${TS\_Properties#Name}')

log.info ("Properties name is...." + prjname);

def tsname = context.expand ('${#TestSuite#Name}')

log.info ("Properties name is...." + tsname);

def tcname = context.expand('${#TestCase#Name}')

log.info ("Properties name is...." + tcname);

def tspname = context.expand ('${TS\_Properties#Name}')

log.info ("Properties name is...." + tspname);

Types of assertions –

* Contains – verify if the specified text contains in the response?
* Not Contains – verify if the specified text does not contain in the response?
* SOAP Response – verify if it is valid SOAP response or not?
* Valid HTTP Response – verify if the response has valid HTTP response something like 200?
* Response SLA – Verify if the response time within in the boundary limit or not?
* Sensitive Information Exposure – Verify if any sensitive information is leaked in response or not?
* Matching XPath attribute value using expected result
* Using xpath for node verification by using exists(//ns:name) with expected result as true
* Using xpath for unique verification by using count(//ns:age) with expected result as 1
* Using select from content and applying wildcards

Properties in SOAP UI –

* TestCase level properties access ${#TestCase#age}
* TestSuite level properties access ${#TestSuite#name}
* Project level properties access ${#Project#id}
* Properties TestStep ${Properties#dept}
* Properties upload from an external file (Create an notepad file with .properties extension and load it from (Project or TestSuite or TestCase or teststep)
* Property Transfer teststep

Property transfer can be done using property transfer step by –

Declaring the name space (ex: declare namespace ns='http://sample.com/reservation/guest/types')

Providing the xpath of the request containing text of it (ex: //ns:getEmployeeDetailsResponse/ns:return/ns:name/text()

Also providing the name space for the request of other action with xpath

* Accessing custom properties

Propery transder can be achived using command at test step level as below

${addEmployee#Request#//typ:addEmployee/typ:name/text()}

Groovy Scripting using SOAP UI –

REST API –

import groovy.json.JsonSlurpler

def response = messageExchange.response.responseContent

def json1=new JsonSlurpler().parseText(response)

assert context.getTestCase().getPropertyValue("place id") == json1.results[4].id

using XML file -

import com.eviware.sopaui.support.XmlHolder

def response = messageExchange.response.responseContent

def holder = new XmlHolder(response)

holder.getNodeValue("//........../....../......")

Note: If we need to validate only response validation then script assertion, if not we need to update the record or taking the output from one response and give it as input in another request then Groovy script is good option

photos.photo[?(@.id==”12345”)].server

in the expected value, put something like [45445454]