**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**

How do you test webservice?

**With WSDL - web service description language**

* **describes how to access webservice and**
* **what are the operations needed to be performed**
* **SOAP enables client applications to easily connect to**
* **remote services and invoke remote methods**
* **invoke particular method based on the input parameters**
* **SOAP request**
* **SOAP response**
* **Envelope tag (mandatory)**
* **<envelope>**
* **</envelope>**
* **Header (optional)**
* **oath token**
* **Body (mandatory)**
* **Fault(Optional)**
* **if the webservice is not invoked we can use fault tag**
* **when wan error occurs during processing**
* **Using soap request we need to hit the endpoint**
* **the request gets processed and gives back the response**
* **Main task**
* **Is to create proper test suite we need to protocols with validations,assertions**
* **have proper validations for test cases**
* **===========================================================================================|**
* **Assertions**
* **1) Contains Assertions**
* **2) Not contains assertions**
* **3) Soap Assertion**
* **4) HTTP status code assertion**
* **5) Response status time ( should be quick) -**
* **6) senitive information - Security assertion**
* **Xpath Assertions**
* **with the help of xpath we can travel to a particular node in the XML (age,id are nodes)**
* **Xpath = //path to traverse to node - accurate xpath from parent node**
* **Department Xpath - Direct node value**
* **exists(//ns:age) - checks whether node exist**
* **how to validate only one instance is available**

**count(//ns:age)**

* **//ns:return -- displays all the nodes**
* **If we others values also to be accepted then we should Wildcards**

**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>**

Xml is used to tag data

Selenium