**What are Web Services?**

Web services are web components that transfers data between client and server. Client sends a web request to the server and the server then responds to client. This response will differ based on the web service request type.

**Web services in SoapUI:**

SoapUI is designed for validating web services easily.

Let’s look at an example: A flight ticker booking application that runs in City 1 and is being accessed from City 2 to book a ticket. A user enters all the information such as boarding point, destination point, date of journey etc, and then as soon as the “Book Now” button is clicked, the web service from City 1 gets invoked and it passes all the information that is entered to the application server that processes the user request. The Reservation application will then send a response to the User’s request.

Most of the online payment transactions are processed through web services only because of the enhanced security this method offers. An input parameter will be sent to the payment gateway website and which would be processed subsequently. An acknowledgement will be sent to the client regarding payment status finally.

All these activities can be seen through SoapUI request and response screens. SoapUI helps us to evaluate these web services.

**Now let’s see the important components of the web services.** They are,

* **WSDL** – Web Service Description Language
* **SOAP** – Simple Object Access Protocol
* **UDDI** – Universal Description, Discovery and Integration
* **RDF** – Resource Description Framework

**#1. WSDL (Web Services Description Language):**

A WSDL is a document that should be written using XML. This document describes the following details about the web service:

* Origin of the web service
* Header information
* Port type
* Input and output messages

Each of the above information is represented as a tag in the WSDL file, such as:

1. **<types>** – XML Schema data types
2. **<message>** – the actual request and response data being communicated
3. **<portType>**– the target / end points where the actual web service is hosted to perform the operation
4. **<binding>**– the protocol information is given for the data format
5. **<definitions>**– the parent tag for the above mentioned tags

**Now let’s look at a sample WSDL file:**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/webservices-1.jpg)

Your WSDL file should follow the [W3C standard](http://www.w3.org/standards/) as above. Through web services we can convert into web based application. Web services are constructed on top of XML, HTTP, TCP / IP, Java, HTML and so on. Since web services are XML based language so that we can have these applications as local, distributed and web based environments.

**Role of WSDL:**

Validating web services using SoapUI is easy and is only possible with WSDL document because to configure web services in SoapUI, WSDL document is mandatory. If the WSDL document is not valid, SoapUI will throw an exception immediately. Now let us look at **UDDI**component.

**#2. UDDI (Universal Description, Discovery and Integration):**

This is a global repository where we can search the web services spread over the globe. In order to get or search web services just visit <http://uddi.xml.org/> web site. Here you can also register your own web service and make it available to global users.

UDDI is the place where the WSDL is described in detail. This will communicate through the SOAP protocol which will be explored later in this tutorial. Say for example, if you wish to advertise your products to the global customers you could create a web service and host it through UDDI. This can now be accessed by global users and from there the business could be established.

**#3. SOAP (Simple Access Object Protocol):**

Generally, it uses XML based data to interact with web applications.

**Here are some points to remember:**

* SOAP is language and platform independent as it is written by using XML.
* It creates the platform to communicate with the applications that are running in different operating systems using different technologies.
* Most of the Internet applications interact with each other over Remote Procedure Calls that use [DCOM](http://en.wikipedia.org/wiki/Distributed_Component_Object_Model) (Distributed Component) and [CORBA](http://www.corba.org/) (Common Broker Architecture)
* These technologies are different than the HTTP.

RPC (Remote procedure calls) are sometimes blocked by firewalls and proxy servers. To overcome these issues, SOAP was designed. There are some standard rules to be followed while building SOAP requests.

**Let’s take a look at sample SOAP document.**

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/webservices-2.jpg)

------------

As you can see, a **SOAP document must contain the following elements:**

1. **Envelope element** is the top most tag which identifies the XML document as a SOAP message.
2. Followed by Envelope element, you see the**header element** that has header information.
3. The **Body element** specifies the call and response information.
4. Finally, you have a **Fault element** which contains errors and status information.

The above said elements should be declared with default namespace for the SOAP envelope.

Generally, a protocol is a set of standard rules that transfers the data between two regions in the Internet over the web services. There are many protocols that are used in the Internet applications. They are, Transmission Control Protocol (TCP) which serves as a packet between two connections. Internet Protocol (IP) that sends and receives the messages between two destinations.

**Let us see some other important protocols:**

* Hyper Text Transfer Protocol (HTTP)
* File Transfer Protocol (FTP)
* Border Gateway Protocol (BGP) and
* Dynamic Host Configuration Protocol (DHCP)

These protocols are used according to the requirements.

**#4. RDF (Resource Description Framework):**

RDF contains the description of the web resources such as title, author, content, and copyright information. This framework was designed so that computers can be read and understood easily by the web.

RDF is completely written by using XML language.

RDF data can be transferred between different types of computers using different operating systems and programming languages. Generally, RDF uses Uniform Resource Identifiers (URIs) on the web and it describes the resources along with the property and property values.

Take a look at the **sample RDF document** to understand better:

|  |  |  |
| --- | --- | --- |
| 1 | <? xml version="1.0"?> | |
| 2 | <RDF> |

|  |  |  |
| --- | --- | --- |
| 3 | <Descriptionabout="http://www.softwaretestinghelp.com/rdf"> | |
| 4 | <author> Wilfred R. Myers </author> |

|  |  |  |
| --- | --- | --- |
| 5 | <homepage>http://www. softwaretestinghelp.com</homepage> | |
| 6 | </Description> |

|  |  |
| --- | --- |
| 7 | </RDF> |

**What is XML?**

XML (e**X**tensible **M**arkup **L**anguage) is a mark-up language that is used for storing, sharing and formatting data. In general, an XML document is built by the tags. Let us see the sample XML content for a user’s personal information.

**<Firstname>**Joel **</Firstname>**  
**<Lastname>** King **</Lastname>**  
**<Address>** 1432 Valley Drive **</Address>**  
**<City>** New York **</City>**  
**<Country>** United States **</Country>**  
**<Zipcode>**19714 **</Zipcode>**

**Meaning of “*eXtensible*”**and**“M*arkup*”:**

In the above sample, **First name, Last name, Address** etc. are enclosed by less than (<) and greater than (>) symbols. These labels are known as tags and the one with forward slash (/) along with the text, that is called closing tag. Tags are also called as mark-ups. These are customized as needed. This customization is not possible in other mark-up languages like SGML, HTML and so on. This is why XML is an **extensible**language.

XML focuses on the data for storing, sharing and exchanging as required, and HTML deals with the **format**of the data like applying colours, adding images, changing fonts, styles and so on.

XML and HTML can be used together in applications. For example, if you take a book, there will be textual data and graphical representation formatted. Hypothetically, XML can handle storing actual data and HTML applies the format for the content. Thereby the text book could have information as well as attractive images and colours.

**How XML works with SoapUI?**

As XML is a common language on Internet, it can be integrated with SoapUI because web services are mostly written in the form of XML. Also, if we pass XML input parameter to the web service, the response itself will be in the form of XML. SOAPUI can configure these web services

**Conclusion:**

So far in this tutorial, we took a look at:

* Web services and its several components like WSDL, UDDI, RDF SOAP
* Importance of WSDL document and its body of content
* XML and its usages in SoapUI

………………………………………………………………………………………………………………………………………………………………