What is Web Services?

A web service is a set of open protocols and standards that allow data to be exchanged between different applications or systems.

Where?

Web services can be used by software programs written in a variety of programming languages and running on a variety of platforms to exchange data via computer networks such as the Internet in a similar way to inter-process communication on a single computer .

Why?

**Functions of Web Services**

* It’s possible to access it via the internet or intranet networks.
* XML messaging protocol that is standardized.
* Operating system or programming language independent.
* Using the XML standard, it is self-describing.
* A simple location approach can be used to locate it.

Which and when ?

XML and HTTP is the most fundamental web services platform. The following components are used by all typical web services:

**SOAP (Simple Object Access Protocol)**

SOAP stands for “Simple Object Access Protocol.”

It is a transport-independent messaging protocol.

SOAP is built on sending XML data in the form of SOAP Messages.

A document known as an XML document is attached to each message. Only the structure of the XML document, not the content, follows a pattern.

The best thing about Web services and SOAP is that everything is sent through HTTP, the standard web protocol.

A root element known as the element is required in every SOAP document. In an XML document, the root element is the first element. The “envelope” is separated into two halves. The header comes first, followed by the body. The routing data, or information that directs the XML document to which client it should be sent to, is contained in the header. The real message will be in the body.

**UDDI (Universal Description, Discovery, and Integration)**

UDDI is a standard for specifying, publishing and discovering a service provider’s online services.

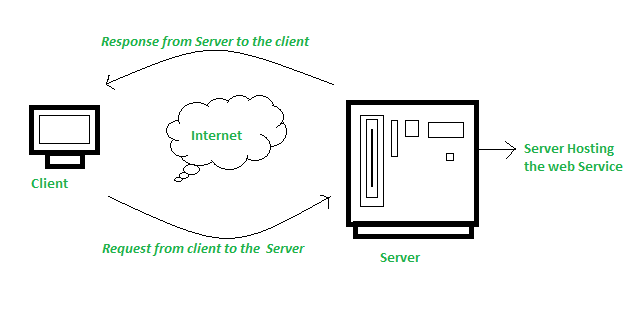
It provides a specification that aids in the hosting of data via web services. UDDI provides a repository where WSDL files can be hosted so that a client application can discover a WSDL file to learn about the various actions that a web service offers. As a result, the client application will have full access to the UDDI, which serves as a database for all WSDL files.  
The UDDI registry will hold the required information for the online service, just like a telephone directory has the name, address, and phone number of a certain individual. So that a client application may figure out where it is.

**WSDL (Web Services Description Language)**

If a web service can’t be found, it can’t be used. The client invoking the web service should be aware of the location of the web service. Second, the client application must understand what the web service does in order to invoke the correct web service. The WSDL, or Web services description language, is used to accomplish this. The WSDL file is another XML-based file that explains what the web service does to the client application. The client application will be able to understand where the web service is located and how to use it by using the WSDL document.

**How Does Web Service Work?**

The diagram depicts a very simplified version of how a web service would function. The client would use requests to send a sequence of web service calls to a server that would host the actual web service.



Remote procedure calls are what are used to make these requests. Calls to methods hosted by the relevant web service are known as Remote Procedure Calls (RPC). Example: Flipkart offers a web service that displays prices for items offered on Flipkart.com. The front end or presentation layer can be written in .Net or Java, but the web service can be communicated using either programming language.  
The data that is exchanged between the client and the server, which is XML, is the most important part of a web service design. XML (Extensible markup language) is a simple intermediate language that is understood by various programming languages. It is a counterpart to HTML. As a result, when programs communicate with one another, they do so using XML. This creates a common platform for applications written in different programming languages to communicate with one another.  
For transmitting XML data between applications, web services employ SOAP (Simple Object Access Protocol). The data is sent using standard HTTP. A SOAP message is data that is sent from the web service to the application. An XML document is all that is contained in a SOAP message. The client application that calls the web service can be created in any programming language because the content is written in XML.

# **RESTful Web Services**

REST stands for REpresentational State Transfer.

REST is an architectural style not a protocol.

## **Advantages of RESTful Web Services**

**Fast**: RESTful Web Services are fast because there is no strict specification like SOAP. It consumes less bandwidth and resource.

**Language and Platform independent**: RESTful web services can be written in any programming language and executed in any platform.

**Can use SOAP**: RESTful web services can use SOAP web services as the implementation.

**Permits different data format**: RESTful web service permits different data format such as Plain Text, HTML, XML and JSON.

# **SOAP vs REST Web Services**

There are many differences between SOAP and REST web services. The important 10 differences between SOAP and REST are given below:

|  |  |  |
| --- | --- | --- |
| **No.** | **SOAP** | **REST** |
| 1) | SOAP is a **protocol**. | REST is an **architectural style**. |
| 2) | SOAP stands for **Simple Object Access Protocol**. | REST stands for **REpresentational State Transfer**. |
| 3) | SOAP **can't use REST** because it is a protocol. | REST **can use SOAP** web services because it is a concept and can use any protocol like HTTP, SOAP. |
| 4) | SOAP **uses services interfaces to expose the business logic**. | REST **uses URI to expose business logic**. |
| 5) | **JAX-WS** is the java API for SOAP web services. | **JAX-RS** is the java API for RESTful web services. |
| 6) | SOAP **defines standards**to be strictly followed. | REST does not define too much standards like SOAP. |
| 7) | SOAP **requires more bandwidth** and resource than REST. | REST **requires less bandwidth** and resource than SOAP. |
| 8) | SOAP **defines its own security**. | RESTful web services **inherits security measures** from the underlying transport. |
| 9) | SOAP **permits XML** data format only. | REST **permits different** data format such as Plain text, HTML, XML, JSON etc. |
| 10) | SOAP is **less preferred** than REST. | REST **more preferred** than SOAP. |