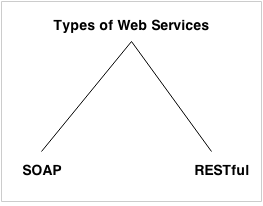
**RESTful Web Services**

# What is Web Service

A **Web Service** is can be defined by following ways:

* is a client server application or application component for communication.
* method of communication between two devices over network.
* is a software system for interoperable machine to machine communication.
* is a collection of standards or protocols for exchanging information between two devices or application.

Let's understand it by the figure given below:

As you can see in the figure, java, .net or PHP applications can communicate with other applications through web service over the network. For example, java application can interact with Java, .Net and PHP applications. So web service is a language independent way of communication.

## Types of Web Services

There are mainly two types of web services.

1. SOAP web services.
2. RESTful web services.

# Web Service Components

There are three major web service components.

1. SOAP
2. WSDL
3. UDDI

## SOAP

SOAP is an acronym for Simple Object Access Protocol.

SOAP is a XML-based protocol for accessing web services.

SOAP is a W3C recommendation for communication between applications.

SOAP is XML based, so it is platform independent and language independent. In other words, it can be used with Java, .Net or PHP language on any platform.

## WSDL

WSDL is an acronym for Web Services Description Language.

WSDL is a xml document containing information about web services such as method name, method parameter and how to access it.

WSDL is a part of UDDI. It acts as a interface between web service applications.

WSDL is pronounced as wiz-dull.

## UDDI

UDDI is an acronym for Universal Description, Discovery and Integration.

UDDI is a XML based framework for describing, discovering and integrating web services.

UDDI is a directory of web service interfaces described by WSDL, containing information about web services.

# SOAP Web Services

SOAP stands for Simple Object Access Protocol. It is a XML-based protocol for accessing web services.

SOAP is a W3C recommendation for communication between two applications.

SOAP is XML based protocol. It is platform independent and language independent. By using SOAP, you will be able to interact with other programming language applications.

## Advantages of Soap Web Services

**WS Security**: SOAP defines its own security known as WS Security.

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

## Disadvantages of Soap Web Services

**Slow**: SOAP uses XML format that must be parsed to be read. It defines many standards that must be followed while developing the SOAP applications. So it is slow and consumes more bandwidth and resource.

**WSDL dependent**: SOAP uses WSDL and doesn't have any other mechanism to discover the service.

# 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 **inherit 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. |

# Service Oriented Architecture (SOA)

Service Oriented Architecture or SOA is a design pattern. It is designed to provide services to other applications through protocol. It is a concept only and not tied to any programming language or platform.

Web services is a technology of SOA most likely.

## Service

A service is well-defined, self-contained function that represents unit of functionality. A service can exchange information from another service. It is not dependent on the state of another service.

## Service Connections

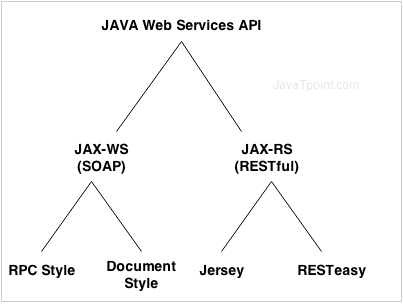
The figure given below illustrates the service oriented architecture. Service consumer sends service request to the service provider and service provider sends the service response to the service consumer. The service connection is understandable to both service consumer and service provider.

# Java Web Services Tutorial

**Java web services tutorial** provides concepts and examples of two main java web services api: JAX-WS and JAX-RS. The java web service application can be accessed by other programming languages such as .Net and PHP.

Java web service application perform communication through WSDL (Web Services Description Language). There are two ways to write java web service application code: SOAP and RESTful.

## Java Web Services API

There are two main API's defined by Java for developing web service applications since JavaEE 6.

1) **JAX-WS**: for SOAP web services. The are two ways to write JAX-WS application code: by RPC style and Document style.

2) **JAX-RS**: for RESTful web services. There are mainly 2 implementation currently in use for creating JAX-RS application: Jersey and RESTeasy.

# JAX-WS tutorialJAX-WS Tutorial

**JAX-WS tutorial** is provides concepts and examples of JAX-WS API. This JAX-WS tutorial is designed for beginners and professionals.

There are two ways to develop JAX-WS example.

1. RPC style
2. Document style

# Difference between RPC and Document web services

There are many differences between RPC and Document web services. The important differences between RPC and Document are given below:

## RPC Style

1) RPC style web services use method name and parameters to generate XML structure.

2) The generated WSDL is **difficult to be validated** against schema.

3) In RPC style, SOAP **message is sent as many elements**.

4) RPC style message is **tightly coupled**.

5) In RPC style, SOAP message **keeps the operation name**.

6) In RPC style, parameters are sent as **discrete values**.

## Document Style

1) Document style web services **can be validated against predefined schema**.

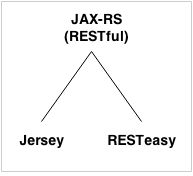
2) In document style, SOAP message is **sent as a single document**.

3) Document style message is **loosely coupled**.

4) In Document style, SOAP message **loses the operation name**.

5) In Document style, parameters are sent in **XML format**.

# JAX-RS Tutorial

**JAX-RS tutorial** is provides concepts and examples of JAX-RS API. This JAX-RS tutorial is designed for beginners and professionals.

There are two main implementation of JAX-RS API.

1. Jersey
2. RESTEasy

## JAX-RS Annotations

The **javax.ws.rs** package contains JAX-RS annotations.

|  |  |
| --- | --- |
| **Annotation** | **Description** |
| Path | It identifies the URI path. It can be specified on class or method. |
| PathParam | represents the parameter of the URI path. |
| GET | specifies method responds to GET request. |
| POST | specifies method responds to POST request. |
| PUT | specifies method responds to PUT request. |
| HEAD | specifies method responds to HEAD request. |
| DELETE | specifies method responds to DELETE request. |
| OPTIONS | specifies method responds to OPTIONS request. |
| FormParam | represents the parameter of the form. |
| QueryParam | represents the parameter of the query string of an URL. |
| HeaderParam | represents the parameter of the header. |
| CookieParam | represents the parameter of the cookie. |
| Produces | defines media type for the response such as XML, PLAIN, JSON etc. It defines the media type that the methods of a resource class or MessageBodyWriter can produce. |
| Consumes | It defines the media type that the methods of a resource class or MessageBodyReader can produce. |

# RESTful JAX-RS File Download Example

We can download text files, image files, pdf files, excel files in java by JAX-RS API. To do so we need to write few lines of code only. Here, we are using jersey implementation for developing JAX-RS file download examples.

You need to specify different content type to download different files. The @Produces annotation is used to specify the type of file content.

1. **@Produces("text/plain")**: for downloading text file.
2. **@Produces("image/png")**: for downloading png image file.
3. **@Produces("application/pdf")**: for downloading PDF file.
4. **@Produces("application/vnd.ms-excel")**: for downloading excel file.
5. **@Produces("application/msword")**: for downloading ms word file.

# RESTful JAX-RS File Upload Example

Like download in previous page, we can easily upload a file such as image file, pdf file, excel file, text file etc.

The @FormDataParam("file") annotation is used to mention file parameter in the service class. The @Consumes(MediaType.MULTIPART\_FORM\_DATA) is used to provide information of the file upload.

To upload file using JAX-RS API, we are using jersey implementation.

To upload file through jersey implementation, you need to provide extra configuration entry in web.xml file.

**<init-param>**

**<param-name>**jersey.config.server.provider.classnames**</param-name>**

**<param-value>**org.glassfish.jersey.filter.LoggingFilter;

     org.glassfish.jersey.media.multipart.MultiPartFeature**</param-value>**

**</init-param>**