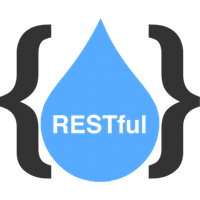
RESTful Services



# What are Web Services?

Web services are the types of internet software that uses standardized messaging protocol over the distributed environment. It integrates the web-based application using the **REST, SOAP, WSDL,** and **UDDI** over the network. For example, Java web service can communicate with .Net application.

## **Features of web Services**

* Web services are designed for application-to-application interaction.
* It should be interoperable.
* It should allow communication over the network.

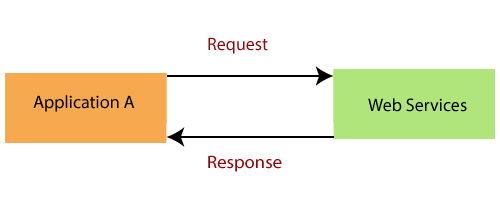
## **Components of Web Services**

The web services must be able to fulfil the following conditions:

* The web service must be accessible over the internet.
* The web service is discoverable through a common mechanism like UDDI.
* It must be interoperable over any programming language or Operating System.

## **How does data exchange between applications?**

Suppose, we have an **Application A** which create a request to access the **web services**. The web services offer a list of services. The web service process the **request** and sends the **response** to the Application A. The input to a web service is called a request, and the output from a web service is called response. The web services can be called from different platforms.



There are two popular formats for request and response **XML** and **JSON**.

**XML Format:** XML is the popular form as request and response in web services. Consider the following XML code:

1. <getDetail>
2. <id>DataStructureCourse</id>
3. </getDetail>

The code shows that user has requested to access the DataStrutureCourse. The other data exchange format is JSON. JSON is supported by wide variety of platform.

**JSON Format:** JSON (javascript object notation) is a readable format for structuring data. It is used for transiting data between server and web application.

[

1. "employee":
2. {
3. "id": 00987
4. "name":       "Jack",
5. "salary":      20000,
6. }
7. ]

To make a web service platform-independent, we make the **request** and **response** platform-independent.

Now a question arises, how does the **Application A** know the format of Request and Response?

The answer to this question is "Service Definition." Every web service offers a service definition. Service definition specifies the following:

* **Request/ Response format:** Defines the request format made by consumer and response format made by web service.
* **Request Structure:** Defines the structure of the request made by the application.
* **Response Structure:** Defines the structure of response returned by the web service.
* **Endpoint:** Defines where the services are available.(url)

# Types of Web Services

There are two types of web services:

* RESTful (**Representational State Transfer)** Web Services
* SOAP **(Simple Object Access Protocal)** Web Services

## **RESTful Web Services**

RESTful Web Services are client and server applications that communicate over the WWW. RESTful Web Services are REST Architecture based Web Services. In REST Architecture, everything is a resource.

RESTful Web Services provides communication between software applications running on different platforms and frameworks.

We can consider web services as code on demand.

A RESTful Web Service is a function or method which can be called by sending an HTTP request to a URL, and the service returns the result as the response.

In this tutorial, you will learn the basics of RSETful Web Services with suitable examples and projects.

The important methods of HTTP are:

* **GET:** It reads a resource.
* **PUT:** It updates an existing resource.
* **POST:** It creates a new resource.
* **DELETE:** It deletes the resource.

For example, if we want to perform the following actions in the social media application, we get the corresponding results.

**POST /users:** It creates a user.

**GET /users/{id}:** It retrieve the detail of one user.

**GET /users:** It retrieve the detail of all users.

**DELETE /users:** It delete all users.

**DELETE /users/{id}:** It delete a user.

**GET /users/{id}/posts/post\_id:** It retrieve the detail of a specific post.

**POST / users/{id}/ posts:** It creates a post for a user.

**GET /users/{id}/post:** Retrieve all posts for a user

HTTP also defines the following standard status code:

* **404:** RESOURCE NOT FOUND
* **200:** SUCCESS
* **201:** CREATED
* **401:** UNAUTHORIZED
* **500:** SERVER ERROR

### **RESTful Service Constraints**

* There must be a service producer and service consumer.
* The service is stateless.
* The service result must be cacheable.
* The interface is uniform and exposing resources.
* The service should assume a layered architecture.

### **Advantages of RESTful web services**

* RESTful web services are **platform-independent**.
* It can be written in any programming language and can be executed on any platform.
* It provides different data format like **JSON, text, HTML,** and **XML**.
* It is fast in comparison to SOAP because there is no strict specification like SOAP.
* These are **reusable**.
* These are **language neutral**.

Steps: -

1. Create a Spring Boot Application using Spring initializer / STS
2. Update that application using Maven Option in Eclipse
3. Create a model(class) in specific package (Student, Customer, ….)
4. Add getters and setters for the model class
5. Now create Controller in a specific package.
6. Now define methods based on type of requests (GetMapping, PostMapping, …)
7. Now Change the port no in “application.properties”(if required).
8. Now Run the application, get the url Ex: <http://localhost:9092/mapobj>

Steps for Test Service

* 1. Download and install POSTMAN / add POSTMAN extension to the google chrome
  2. Open POSTMAN, then send request based type of request

Example

[Initializing a RESTful Web Services - javatpoint](https://www.javatpoint.com/restful-web-services-example)