|  |
| --- |
| | Praveen D |

|  |  |
| --- | --- |
| JSpider Basavangudi Bangalore | Web Services - Introduction |

# Web Services Introduction

* Servlets, JSP along with JDBC helps to develop Web Application
* However "Web Services" a.k.a "Application Services" helps web applications to interact with any other applications (Mobile Apps / Desktop Applications / Web Applications)
* Web Services, as name implies, they are the services available in "Web". They take the request from Applications, via any communication protocol such as HTTP & generate Response
* Web Services, is a concept, are independent of
* Platform
* Programing Languages &
* Applications
* Web Services take Web-applications to the Next Level!!!
* Unlike Web Applications, Web Services do not provide the user with a GUI. In other words Web services are meant for "Inter-System Communication / Application-to-Application Interactions but not for Users"
* Hence Web Services do not require the use of browsers or HTML
* Few real time examples of Web Services
* Mobile Apps (ex: Gmail App) communicate with their corresponding Web Applications ex: Gmail) using Web Services
* Uber / OLA Mobile Apps/Web Application interact with Google Maps using Web Services
* BookMyShow Web Application / Mobile Application interact with PVR Web Application using Web Services

* Web Service has two participants:

1. Service Provider / Producer
2. Service Consumer / Requester

* Service Provider exposes the "Service on Web" & Service Consumer uses this "Web Service"
* Service Provider should be a Web Application; however Service Consumer can be Desktop/Mobile Application or Web Application
* There are 2 types of Web Services
* RESTful Web Services
* SOAP Web Services / XML Web Services
* Java provides 2 API's to develop these web services

1. "JAX-RS" API for developing "RESTful Web Services"
2. "JAX-WS" API for developing "SOAP Web Services"

* "javax.ws.rs.\*" is the package representation of JAX-RS API

"javax.jws.\*" is the package representation of JAX-WS API

# Why Web Services / Advantages of Web Services

## Web Services Interoperability (WS-I)

* Web Services are "Application, Platform and Technology Independent"
* Ex: VB / .NET applications can talk to Java Web Services and vice versa. Thus helping the organisations to use existing applications which are developed using other technologies (may be legacy)
* Also Web Services helps/allows different applications to share data among themselves
* Ex: Uber / OLA and Google Maps shares the data among each other

## 2. Loosely Coupled

* + - Each application is independent of one another. Hence changes done to one application will not impact the "unrelated areas"

Ex: Changes / Modifications done at Uber / OLA will not impact Google Maps

## 3. No need of re-inventing the wheel

* + - Web Services reduces the software development time
    - This helps the other business partners to quickly develop application and start doing business
    - This helps business to save time and money by cutting development time

Ex: Uber / OLA can make use of Google Maps

## 4. Business Opportunity

* Web Services will open the door for new business opportunities by making it easy to connect with partners

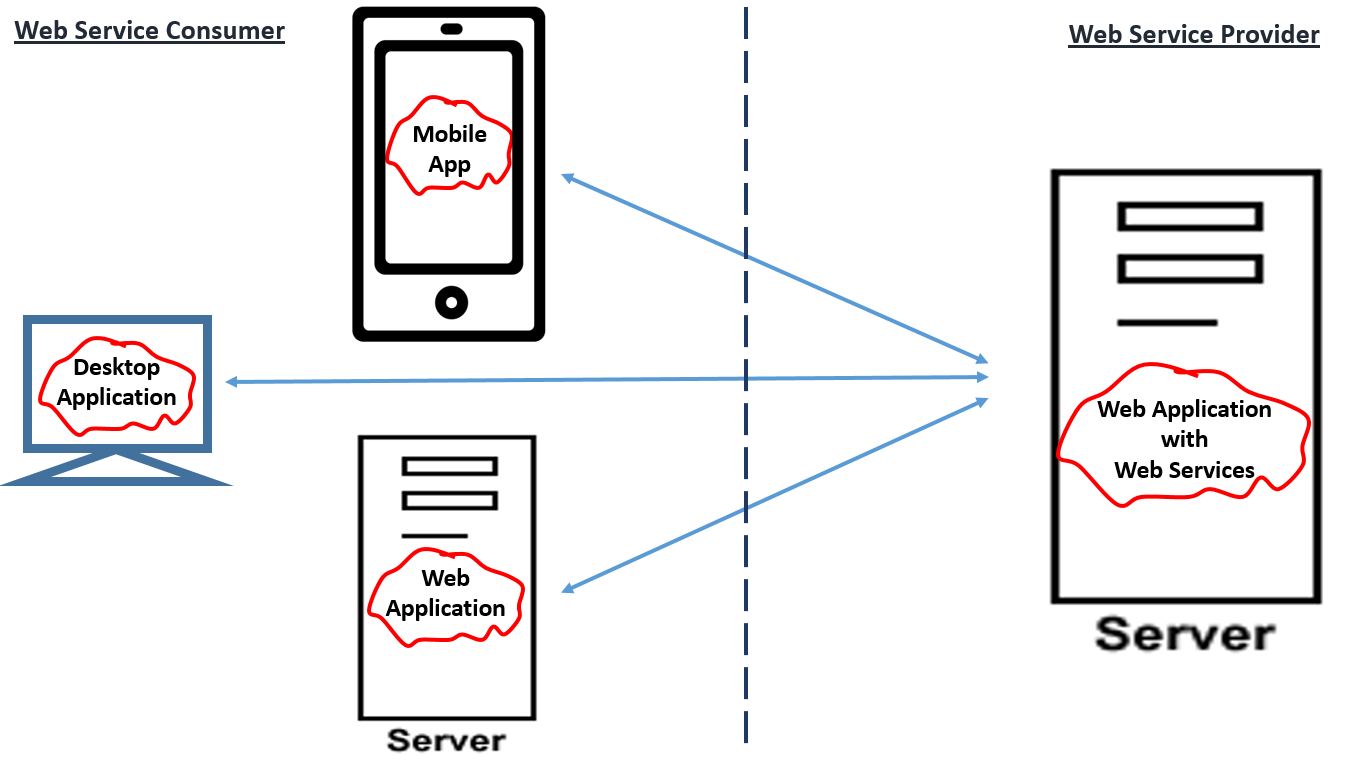
Ex: Dominos can get the order from Food Panda / Swiggy along with getting orders from its own site

## 5. Service Reuse

* Web Services takes code reuse a step further
* A specific service within the organisation is only coded once and used over and over again by other applications

Ex: An organization can have a "Single Payment Gateway service" which helps other web applications of that organization to interact with "Payment Gateways"

# Web Services Architecture



**Note:**

* Web Service Producer MUST be a Web Application
* However Web Service Consumer can be
* Web Application OR
* Desktop Application OR
* Mobile Application
* Web Application can “Produce” some web services as well as “Consume” some other web services. In other words, web application can have both “Producers and Consumers”