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

|  |  |
| --- | --- |
| JSpider Basavangudi Bangalore | Web Services – Day 1 My First Web Services |

# About Applications/ J2EE Introduction

* Every Application has its own dedicated functionality

For ex:

* Adobe Reader can only open pdf files
* Media Player can only open media files
* Every Application has its own known file extensions

For ex:

* Adobe Reader can only understand ".pdf" files
* Media Player can only understand ".mp3"/".avi" files

* Every Application is "Platform Dependent"
* "Applications acts like an interface between "User" and an "Operating System"
* "Operating System" acts like an interface between an "Applications" and "underlying Hardware"
* There are 2 types of Applications

1. Standalone application (Un-shared)
2. Web Applications (Shared)

* There are 2 Types of Standalone Applications

1. Desktop Applications
2. Mobile Applications

* Server is also a computer, but they are not dedicated to a particular user instead they are "Shared Computer". Hence Servers will not have a "Monitor / Terminal"
* “Web Servers” / “Application Services” are like any other application (Adobe Reader, Media Player, etc.,), they are also an application which runs on Operating System. As the name implies they “Serves requests to a Web Applications”

* In other words, it helps both web browser & web application to interact with each other. Hence every web application is directly under the control of webserver

# Installation of necessary Applications to work with Web Services

## Java Development Kit (JDK)

## Eclipse

## Chrome Browser

## Tomcat Web Server (Servlet Container)

## Apache Maven

## Git Client

## Tortoise Git

## Fiddler

## Postman REST Client

## XML Validator

# Our First Web Service (Producer & Consumer)

## Steps for Web Service "Producer Web Application"

1. Created the Maven Project by providing the below information

Archetype : maven-archetype-webapp

Group ID : com.jspiders

Artefact ID : googlemaps

1. Under Java Resources if "src/main/java" folder is not present then, create it
2. Added the "Glassfish Jersey & Servlet" Dependencies (given below) to pom.xml
3. Created the Java Class which hosts the Rest Web services using Jersey Framework

## Steps for Web Service "Consumer Web Application"

1. Created the Maven Project by providing the below information

Archetype : maven-archetype-webapp

Group ID : org.jspiders

Artifact ID : mywebapp

1. Under Java Resources if "src/main/java" folder is not present then, create it
2. Added the "Glassfish Jersey & Servlet" Dependencies (given below) to pom.xml
3. Created the Servlet which interact with above web application to consume the Web Services
4. Configured this Servlet in web.xml

## Maven Dependencies

* URL to get the Dependencies: <https://mvnrepository.com>
* Jersey Dependencies:
* jersey-client
* jersey-server
* jersey-container-servlet
* jersey-media-json-jackson
* jersey-common
* jersey-media-multipart
* jersey-container-servlet-core
* Servelet Dependency: javax.servlet-api