SERVLETS

Servlets:

Servlets is a specification for developing web applications with Java programming language.

Web Application:

A web application is a distributed application which runs on browser & server.

Distributed Application:

An application that is installed on one computer and runs on many computers is called distributed application.

Application:

An application is a program in which we interact with on the desktop.

Browser:

A browser is a software that executes web pages containing text, image, graphics, animation, audio, video, .. etc.,

List of Browsers:

1) Internet Explorer

2) Netscape Navigator

3) Neo Planet

4) Hot Java

5) Eye Browse

6) Cyber Dog

7) America Online

8) Mozilla Firefox

9) Mosaic

10) Opera

11) Mac Web

12) Win Web

13) Lynx

14) Google Chrome

Browser is called as web client.

Client:

A client is a software that sends the request to server to get the response.

Server:

A server is a software that receives request from the client, process the request, constructs the response and sends the response back to a client.

Types of Servers:

1) Web Servers

2) Application Servers

1) Web Server:

A web server is a server which contains only web container

List of web servers:

1) Tomcat web server

2) Pramathi web server

3) iPlanet web server

4) Java web server

5) Personal web server

6) Jetty web server

7) Resin web server .. etc.,

2) Application Server:

An application server is a server which contains both web container & EJB container.

EJB stands for Enterprise Java Beans

List of Application Servers:

1) WebLogic Application Server

2) WebSphere Application Server

3) JBoss Application Server

4) Sun One Application Server

5) Glass Fish Application Server .. etc.,

Specification:

A specification is a set of rules & guidelines that are used to develop applications & environments.

Servlets specification used by vendors to develop web containers.

Web container is a part web server & application server.

Servlets specification used by Java programmers to develop web applications.

There are two types of web applications:

1) Static Web Applications

2) Dynamic Web Applications

1) Static Web Applications:

A web application that is already prepared and placed in server is known as static web application.

Static web applications are common to all users.

Static web application resides in server & runs in browser.

The server sends the program to a browser whenever request comes to a static web application.

Static web applications can be developed by using HTML, CSS, Java Script, Applets, .. etc.,

HTML stands for Hyper Text Markup Language.

CSS stands for Cascading Style Sheets.

Applets are Java programs.

2) Dynamic Web Applications:

A web application that is prepared dynamically whenever request comes to a server is known as dynamic web application.

Dynamic web applications are specific to user.

Dynamic web application resides in server & runs in server only.

The server executes the program & sends the output to a browser whenever request comes to a dynamic web application.

Dynamic web applications can be developed by using Servlets, JSP, Struts, JSF, Spring MVC, PHP, CGI, ASP, .. etc.,

JSP stands for Java Server Pages.

JSF stands for Java Server Faces.

MVC stands for Model View Controller

PHP stands for Personal Home Page

CGI stands for Common Gateway Interface

ASP stands for Active Server Pages

Servlets, JSP, Struts, JSF & Spring MVC are Java programs.

CGI Vs Servlets

1) CGI is a specification for developing web applications with C, C++,Perl, .. etc.,

1) Servlets is a specification for developing web applications with Java programming language only.

2) CGI Based web server creates a new process for every request.

2) Servlets based web server creates a new process for first request only.

Servlets to do not have main() method because servlet runs under server.

There are life cycle methods to run servlet under server.

Life Cycle Methods:

1) init() method

2) service() method

3) destroy() method

init() method called by web container whenever first request comes to a servlet.

service() method called by web container for every request.

destroy() method called by web container whenever servlet instance is removed from web container.

Servlet instance is removed from web container whenever web application is undeployed from server or server shuts down.

The above life cycle methods are the part of javax.servlet.Servlet interface.

Every servlet must implements javax.servlet.Servlet interface to derive life cycle.

Every servlet class must be public because it should be accessible to web container to create an object to call life cycle methods.

Steps to develop web application:

1) Create a web application folder structure.

2) Create and compile web application source code(Servlet Program).

3) Write deployment descriptor(web.xml file)

4) Create a WAR(Web ARchive) file

5) Start the Web Server or Application Server

6) Deploy a WAR file on Server

javax.servlet.Servlet

Methods:

public abstract void init(ServletConfig) throws ServletException;

public abstract ServletConfig getServletConfig();

public abstract void service(ServletRequest, ServletResponse)

throws ServletException, IOException;

public abstract String getServletInfo();

public abstract void destroy();

Servlets API:

1) javax.servlet package

2) javax.servlet.http package

javax.servlet package

Classes	Interfaces
1) GenericServlet	1) Servlet
2) ServletInputStream	2) ServletRequest
3) ServletOutputStream	3) ServletResponse
4) ServletException	4) ServletConfig
5) UnavailableException	6) ServletContext
	7) RequestDispatcher
	8) SingleThreadModel

javax.servlet.http package

Classes	Interfaces
1) HttpServlet	1) HttpServletRequest
2) Cookie	2) HttpServletResponse
	3) HttpSession

Servlets API is a part of servlet-api.jar file in a tomcat server.

servlet-api.jar file location:

C:\Program Files\Apache Software Foundation\Tomcat 9.0\lib

FirstServlet.java

```
import java.io.*;
import javax.servlet.*;
public class FirstServlet extends GenericServlet
{
 public void service(ServletRequest req, ServletResponse res)
 {
  try{
  PrintWriter pw=res.getWriter();
  pw.println("Welcome to Servlets");
  }catch(Exception e)
  {
   System.err.println(e);
  }
```

```
 }
}
```

web.xml

```
<web-app>
<servlet>
<servlet-name>demo</servlet-name>
<servlet-class>FirstServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>demo</servlet-name>
<url-pattern>/test</url-pattern>
</servlet-mapping>
</web-app>
```


By

Mr. Venkatesh Mansani

Naresh i Technologies