Url URL

Web.xml

Request

Server

Client

Method type

Data

* Data

web.xml

Response

**Response**

* Status type
* Content type
* Actual Data

**Servlet:-**

It is a API which helps to establish the connection between front end and java programme.

After connection this API also convert frontend data to java understandable object and java object to frontend understandable response.

**Server-:**

Server is simple computer where deploy (run or excute) our project

**Components of Servlet Architecture :-**

To a servlet project we need a client or front pages which will send the request to the server and on the basis of request a suitable response is obtained from server side to client side.

**Request:-**Client sends request to the server request object content 3 things url, method type and data.

**URL:**- On basis of the url specific servlet programming executing.

**Method type:-** It controls specific method to be executed from servlet class.

**Data:**-**It is user input information from frontend to server side.**

**Note:-**In request object it is mandatory tag it will contain method type and URL and the data depends on situation.

**Response:-**Server side programme communicates with client side by sending a response. Response contain 3 things Status code, Content Type and actual content out of this three it is mandatory that response object will content all this three piece of information. For each request response should be generate.

**Status Code:-**It is set of predefined number which gives us the information about the status of our request.

Example:404 data not found,500 internal server error,200 susses.

**Deployment Descriptor / web.xml / web configuration file:-**

This file is responsible for mapping between URL and servlet classes .For manual mapping of url and servlet classes we need follow following structure of tags in web.xml.

<servlet>

<servlet-name>first</servlet-name>

<servlet-class>data.UserData</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>first</servlet-name>

<url-pattern>/user</url-pattern>

* Inside servlet tag we providing information of servlet class by providing fully qualified name of class inside servlet class tag.
* Inside servlet mapping tag we provide the information of url inside url pattern tag, Url is achieved by writing the action name from the from prefixed with forward slash ( / ).
* Servlet tag and servlet mapping are linked with each other by the help of servlet-name tag.

Important tag from Pom.xml:-

**<dependencies> -:**It contains multiple dependency tag.

**<dependency> -:**This tag is used to import new libraries in the project.

**<plugin> -:** plugin tag are responsible for project structure or arch-types(Architecture type).

Project:

Name,Email,Adress,age

**Servlet Life Cycle:-**

1. Servlet class is loaded-> The classloader is responsible to load the servlet class. The servlet class is loaded when the first request for the servlet is received by the web container.
2. Servlet instance is created-> The web container creates the instance of a servlet after loading the servlet class. The servlet instance is created only once in the servlet life cycle.
3. init method is invoked-> The web container calls the init method only once after creating the servlet instance. The init method is used to initialize the servlet. It is the life cycle method of the javax.servlet.Servlet interface. Syntax of the init method is given below:

**public** **void** init(ServletConfig config) **throws** ServletException

1. service method is invoked-> The web container calls the service method each time when request for the servlet is received. If servlet is not initialized, it follows the first three steps as described above then calls the service method. If servlet is initialized, it calls the service method. Notice that servlet is initialized only once. The syntax of the service method of the Servlet interface is given below.
2. destroy method is invoked-> The web container calls the destroy method before removing the servlet instance from the service. It gives the servlet an opportunity to clean up any resource for example memory, thread etc. The syntax of the destroy method of the Servlet interface is given below.

There are 3 types of servlet:

* Servlet Interface
* Generic Servlet
* Https Servlet

Servlet Interface:-

It is Interface .its present in javax.servlet

|  |  |
| --- | --- |
| **Method** | **Description** |
| **public void init(ServletConfig config)** | initializes the servlet. It is the life cycle method of servlet and invoked by the web container only once. |
| **public void service(ServletRequest request,ServletResponse response)** | provides response for the incoming request. It is invoked at each request by the web container. |
| **public void destroy()** | is invoked only once and indicates that servlet is being destroyed. |
| **public ServletConfig getServletConfig()** | returns the object of ServletConfig. |
| **public String getServletInfo()** | returns information about servlet such as writer, copyright, version etc. |

**GenericServlet:-**

**GenericServlet** class implements **Servlet**, **ServletConfig** and **Serializable** interfaces. It provides the implementation of all the methods of these interfaces except the service method.

GenericServlet class can handle any type of request so it is protocol-independent.

**HttpServlet:-**

The HttpServlet class extends the GenericServlet class and implements Serializable interface. It provides http specific methods such as doGet, doPost, doHead, doTrace etc.

**Request Dispatcher:-**

It is responsible for dispathichg the request to another resource it may be html,servlet or jsp.

**Forward()method:-**



**Inclde method:-**



RequestDispatcher rd=request.getRequestDispatcher("servlet2");

rd.forward(request, response);

**Welcome File and Welcome file List:-**

**Deployment Descriptor:-**

* **File that contains configuration of your java web application**
* **It inside in the WEB-INF folder.**

**Session Tracking:-**

* Session Tracking is way to maintain state (data) of an user.
* It is also known as State Management.
* Session Tracking Technique:

1. Cookie
2. Hidden From Field
3. URL Rewriting
4. HttpSession

**Cookies:-**

* Cookies are the textual information which are stored in key value pair format to the client’s browser during multiple request.
* It is nothing when ever client request to server for some information he response generated then another time same request for that machine send request+data(cookies) send to the server response generated previously what will store that it is nothing session storage some time storing the client details till server running when server shutdown cookies also destroy.
* **How to use cookie in java:-**
* In order to use cookies in java there is cookie class in java in javax.servlet.http package.
* To make cookie just create a object of cookie class and pass name (key) and its value.’