**Discussion on Servlets and JSP**

Static Resource

* Exists @ the server-side before making the request
* Content doesn’t change for every request

Dynamic Resource

* Does not exist @ the server-side before the request is made
* Content may change for every request

Web Application

Any application accessed over the internet/network with the help of web browser

Servlet

It is an API of J2EE that helps a web server to serve the dynamic resources

Servlets are any other java program that accept web request via web server and generate dynamic response to be given back to the browser

Tomcat – WebServer

Eclipse

URL – to make a request

Other Web Servers –

Oracle WebLogic

**Apache Tomcat**

RedhatJBoss

GlassFish

IBM WAS

Web Browser – It is an application that helps us to interact with the web server.

It takes the request from the user, gives it to the Web server and the response given by the server is displayed by the web browser. HTML is the language which is understood by web browser and web browser knows to display the content present in HTML.

Install Tomcat 8.5/9.0

https://archive.apache.org/dist/tomcat/tomcat-8/v8.5.0/bin/

<https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.1/bin/>

Limitations of WebServer –

* It cannot interact with the DB directly
* It cannot generate dynamic response

J2EE

Build Process – WAR file (Web Archive) compressed file

WAR - .class files + other resources + dependent jar files

WAR file represents web application

Deployment Process – Process of moving the WAR file to the server deploy path

Tomcat - <Tomcat –location>/webapps

Create a web application in Eclipse – SampleApp

(Remember to check the check-box – Generate web.xml deployment descriptor)

Web URL

Web URL uniquely identifies the resource present in a network

Protocol://domain:port/path?query\_string#fragment\_id

**Protocol**-> rules (http)

**Domain** ->computername/ip address in which your web server is installed

Port -> 16-bit number which uniquely identifies a particular application on the OS. Optional

Http: 8080 Https:443

Path -> Typically refers to combination of files and folder structure @ the server-side

Query\_string -> It is a name-value string pairs which passes information to the dynamic resources such as Servlets and JSP.

Optional. Always follows ?

If there are more than one name-value string pair, they are separated by &

Fragment\_id -> internal page reference and it refers to a particular section within a web-page.

Optional. Always follows #

Ex: [http://localhost:8080/root/example/index.jsp?name=”Alex”&age=”50”#troubleshoot](http://localhost:8080/root/example/index.jsp?name=#troubleshoot)

Servlet Container/Web Container

Servlet – WebServer

All servlets that we develop are directly under the control of Servlet Container

http://localhost:8080/SampleApp/calculate - which Servlet serves this request

web.xml

All .java files – src

All non-java files –WebContent

Classes/Interfaces – javax.servlet.\*

javax.servlet.http.\*

Configure web.xml

<servlet>

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

<servlet-class>basics.HelloWorld</servlet-class>

</servlet>

<servlet-mapping>

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

<url-pattern>/hello</url-pattern>

</servlet-mapping>

Servlet - Hierarchy

Servlet, ServletConfig (Interfaces – lifecycle methods)

GenericServlet (Abstract class – service() implmn)

HttpServlet(Abstract class – service() implmn)

MyServlet(Concrete class – overriding doXXX())

Lifecycle of Servlets – init(), service() and destroy()

In the world of web applications, there are only three ways in which navigation can happen –

1. Typing URL in the browser address bar
2. Clicking on buttons
3. Clicking on hyper links

Protocol – set of rules

HTTP: Hyper Text Transfer Protocol – default port 8080

HTTPs: Hyper Text Transfer Protocol Secure – default port 443

HTML

HTTP Methods –

**Get**, Head, **Post**, Put, Delete, Connect, Option, Trace, Patch

HTTP Request – (B -> S)

1. Http Method (header)
2. URL – Requested Resource (header)
3. Form Data(if any) – (header/body)
4. Cookies (optional)

HTTP Response – (S -> B)

1. Status code (header)

100 – 199 -> Informational

200 – 299 -> Success

300 – 399 ->Redirectional

400 – 499 -> Client errors (observe URL)

500 – 599 -> Server errors

1. Response content type (MIME) – HTML, Image, Text, Video, audio(header)
2. Actual Content (body)
3. Cookies (optional)

To Do:

1. Go through all doXXX() methods in HttpServlet class
2. Compose a program to read the selected value from a group of radio buttons using a Servlet
3. Differentiate between ServletContext and ServletConfig
4. Create a Servlet which fetches Employee Data from a form and display the same through Servlet
5. Develop a custom search application
6. Create a Servlet which counts number of hits for that particular page and print the count on the browser.
7. Create Student information collection form and persist the same in the database through a servlet.

Fiddler Demo

Reading the Form Data –

getParameter() – returns String

getParameterNames() – returns enum of string

getParameterValues() – returns arrays of string

Example for all the above three discussed.

Differences between get and post

|  |  |
| --- | --- |
| **GET** | **POST** |
| 1. Data gets exposed in the URL | Data doesn’t get exposed in the URL |
| 1. Used to get the data(retrieve) | Used to post the data to the server |
| 1. Not secure | Much secured |
| 1. Data transfer limited | Data is unlimited |
| 1. Default | Method=”post” mandatory |
| 1. May have empty body | Body usually is not empty |
|  |  |

Servlet Redirection –

1. Happens @ browser-side
2. URL changes
3. Redirects makes request to contain “GET”, so always doGet() is invoked
4. Slower in operation
5. It has more than one request-response cycle
6. Always invoke sendRedirect() method on response object

Browser --------🡪/RedirectDemo (S1)

<---------

Google.com (S2)

Servlet Forward

1. Happens at server-side
2. URL doesn’t change
3. It has only one request/response cycle
4. Faster in operation
5. Works only for internal URL
6. It involves corresponding doXXX() method @ the destination

Servlet include

Inclusion on static and dynamic resources is possible using RequestDispatcher object

RequestDispatcher r = request.getRequestDispatcher(url);

r.include(request,response);

JSP – Java Server Pages

Its an API of J2EE that accepts a web request and gives a dynamic response

All JSP files - .jsp -> .java

Welcome.jsp -> Welcome.java ->Welcome.class -> response

<http://localhost:8080/SampleApp/Welcome.jsp>

<http://localhost:8080/SampleApp/login/Welcome.jsp>

JSP – it is protocol dependent only (only http/https) – indirectly extends HttpServlet

Why JSP?

Separates business logic from presentation logic

Web Designers/UXUI Designer/Engineer<-> Developers

Lifecycle of JSP

Translation – Instantiation ->Init -> service -> destroy

jspInit()

\_jspService() – cannot override this

jspDestroy()

YourJSPFile extends HttpJspBase (immediate super class)

Tags (5)

1. Declaration tag <%! %> - outside \_jspService(),anywhere
2. Scriptlet tag <% %> - any amount of valid java code, BL, inside \_jspService()
3. Expression tag<%= %> - no semi-colon to be used
4. Action tag <jsp:action\_name attributes>

Forward, Include, UseBean

1. Directive tag <%@directive\_name attributes %>

Include, Taglib, Page

Scope – (explicit or implicit objects)

Page(default)

Request

Session

Application

Implicit Objects (9)

out –JSPWriter - page

request – HttpServletRequest- request

response – HttpServletResponse - request

session – HttpSession - session

config – ServletConfig - page

application – ServletContext - application

page – Object - page

pageContext – PageContext - page

exception – Throwable– page

**SOP2R2ACE**

There are two ways to access a jsp file

1. Typing the URL on the browser – direct access
2. Entries in web.xml

<servlet>

<servlet-name>MyJSP</servlet-name>

**Difference Between Servlets & JSP**

|  |  |
| --- | --- |
| **Servlets** | **JSP** |
| Servlets are protocol Independent | By default JSP is protocol dependent  i.e it handles ONLY HTTP & HTTPS protocols. |
| In Case of Servlets , Business Logic is tightly coupled with presentation Logic i.e “HTML inside Java” | In Case of JSP Business Logic is kept separated From Presentation Logic i.e “Java inside the HTML” |
| In case of Servlets, Already compiled “.class” files are used to generate the response | In case of JSP, it gets converted into Servlet, it gets compiled & the compiled “.class “ file is used to generate the response. |
| Hence time to take generate the response for 1st Request is Less (as compared to JSP) | But in case of JSP , time to take generate the response for 1st Request is More. |
| Servlets are accessed ONLY by using the configured URL present in web.xml/annotation | JSP can be accessed   * Either by using the file name directly in the url * Using the configured URL present in web.xml |
| Servlet extends either “GenericServlet “ or “HttpServlet” | In case of JSP, “Translated servlet extends HttpJspBase “ which in turn extends “HttpServlet” |
| In Case of Servlets (Which extends HttpServlet, Container invokes corespondingdoXXX (HSR, HSR) method depending on the http method present in the request. | Container invokes \_jspService (HSR, HSR) method irrespectinve of the Http Method present in the request. |
| Servlets has “\*.java” file extension | JSP has “\*. jsp” file extension |
| Implicitly objects are Not available we have to explicitly define them | Implicit objects are Available. |