26/02/24

Java Servlet Page (JSP)

Q. What is JSP? (Java Servlet Page)

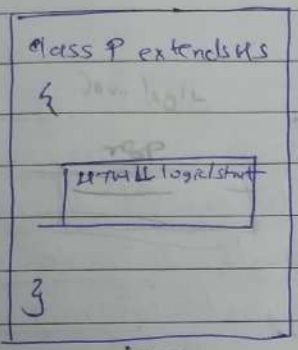
→ - Previously in Servlet we used to write java logic as well as response which we want in HTML format

- In servlet, we use to write HTML Logic / statement with the help of (sendRedirect) in that requestDispatcher ~~or~~ PrintWritter (we are writing HTML statements which we want to display)

- In Servlet, when we were try to write both Java logic and HTML logic / statement we were facing some challenges.

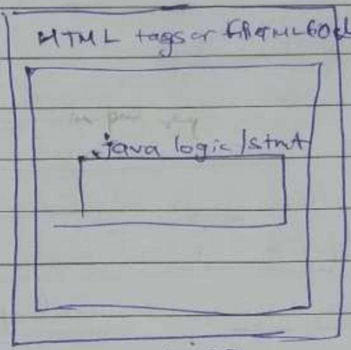
challenges [illegible] formatting with HTML doc

- To avoid all these we have solution which is 'JSP' (Java Servlet Page)



.java

When we have plain servlet



.jsp

when we have JSP

JSP → HTML doc

- In JSP Page basically we write HTML code and as per requirement we can write Java logic within that HTML code.

- Whenever we want some operations to be performed related to Java Buisness Logic (controller type / backend side logic), we can write Java Code inside HTML code with help of tags

- To display something on browser, displaying it with help of servlet is not recommended, it is better to display it through JSP File.

- From the Servlet, we transfer the data to JSP Page and ask JSP Page to display it

~~location of JSP file in Servlet Maven Proj~~

* Where JSP files will be stored in Servlet Maven Project?

→ - JSP files will be stored in Servlet Maven Project in webapp folder (src → Main → webapp)

*(when to use HTML or JSP) w.r.t user response

- When we want to design response for user without any java logic it is recommended to use HTML (Static response → HTML)
- When we want to design response for user by using java logic it is recommended to use JSP (dynamic response → JSP)

7/2/24

JSP Tags

- These are special constructs (theese is build karne ke use main aate hai) used in JSP files to dynamically generate content or web pages (in JSP file)
- These tags allows (java ke) developers to embed Java code, define variables and methods, include external resources, control page flow and interact with Java Beans. (concept used for communicating with external servers)
- JSP tags are categorized into different types
  1. Declaration Tags.
  2. Expression Tags
  3. Scriptlet Tags

* Explaination for:

(expression)
1. Declaration Tag: - These are enclose within <%! %>
- It is used to declare variables & methods
- Declarations are placed outside the service () (can also be doPost() & doGet())
(& inside the servlet class)

* Explaination: - It means declaration inside '<%! %>' are placed outside of any method & within servlet class, meaning, they are directly within class body and outside of any method.

- Declaration tags are defined at class level making them accessible throughout the servlet class.

(printing purpose)
2. Expression Tag: - These are enclosed within <%= %>
- It is used to evaluate expression and output result directly to servlet output
- It automatically converts result to String.

evaluate - expression & directly to servlet o/p

used to display o/p on server side
3. Scriplet Tag: - These are enclosed within <% %>
- It is used to insert Java Code directly into Servlet which allaos dynamic content generation.

9|2|24

JSP Directives

- JSP Directives are special instructions that provides essential information to JSP Container about how to process corresponding JSP Page.
- These directives are enclosed within <%@ %>
- These directives are processed when JSP Page is translated into Servlet by JSP Container

1. Page Directive (<%@page %>)
- These directive ~~are~~ is used to provide instruction to JSP JSP Container about the characteristics of 'Generated Servlet', such as Error Handling, Content Type, Language and more...

(html header mein rehta hai?)

- eg: <%@ page language = "java" contentType = "text/html"; charset = UTF-8" pageEncoding = "UTF-8" %>

2. Include Directive (<%@include %>)
- These directive is used to include the content of another file (JSP, HTML, text) into the current JSP Page at translation time.
  (JSP to Servlet)
- It is a 'static include' Meaning the content of included file is ~~must~~ Merged into Main JSP file during translation.
- Eg: <%@ include file = "Animal.jsp" %>

3. Taglib Directive (<%@taglib %>)
  (whether it is attribute or keyword)
- These directive is used to define and specify a tag library for use in the JSP.
- It declares the custom tags used in JSP

11/2/24

JSP Life Cycle

- JSP is extension of Servlet
- ~~In JSP application~~
- In JSP application, the lifecycle refers to sequence of steps that occur from the time that JSP is requested by a client to the time the response is sent back to client
- Below are key stages in the life cycle of JSP
(same as ~~that~~ to that of servlet life cycle)

1. ~~Compilation~~/Translation:- When a client sends a request for a JSP page, the JSP Container translates the JSP File into Servlet
- The translation process involves converting JSP elements such as HTML and JSP tags into Java Equivalent code to create a servlet class.
- The resulting servlet class implements HTTPServlet interface and contains the logic to handle request & generate response.
([illegible] concrete class & interface with same name?)
(command prompt java run)

2. Compilation:- After translation phase, the e.g. generated servlet class is compiled into byte code by Javac Compiler
(Jvm mein Java compiler hota hai jo source code ko byte code)
- The compiled servlet class is then loaded by class (present in JRE) loader & stored in memory for execution
(we also have JIT compiler (converts byte code into binary code))

(instance of servlet class is created)
3. Initialization:- Upon loading servlet class, Servlet container initializes the servlet instance by calling init()
- The init() is called only once during Servlet's ~~life~~ life cycle / life Time

4. Request Handling: - When client sends request to JSP Page the Servlet container invokes service() to handle request.
- The service() processes the request including retrieving parameters, executing Java Code and generating dynamic content to send back to the client.

5. Destroy: - When a servlet container when it will decide decides to ~~remove~~ remove servlet instance, typically during application shutdown, it calls destroy()
- The destroy() allows servlet to perform cleanup tasks ~~task~~, release resources
- After destroy() is called, the servlet instance is eligible for Garbage Collection.

6. Garbage Collection: - Once the servlet instance is no longer in use and there are no more references to it, Java Garbage Collector can now reclaimed memory occupied by Servlet Instance

What:
Provided by:
why it is provided:

- JSP Objects

(keyword: Developer provided meaning)

- JSP objects are predefined Java objects provided by JSP Container to facilitate dynamic ~~g~~content generation within JSP
- These objects represent various aspects of ~~J~~ JSP environment, including HTTP request, response ~~response,~~ session & more - -
- JSP objects allows developers to interact with Servlet Environment seamlessly without needing to Manually handle Servlet API objects.

[illegible] : ~~http~~ connection, statement, preparedstatement etc)
JDBC objects

- These objects are typically allessed directly within JSP Pagesusing predefined names. and they provide convinient methods and properties to perform common task such as retrieving request parameters , Managing session Managing , attributes , generating dynamic content and controlling flow of application.

- Egs:- of JSP objects:
  request, response, session , config , page etc.