

Date : ___ / ___ / ___

UNIT-05 (LEC-01)

• Servlet [AKTU-20-21, 21-22, 22-23]

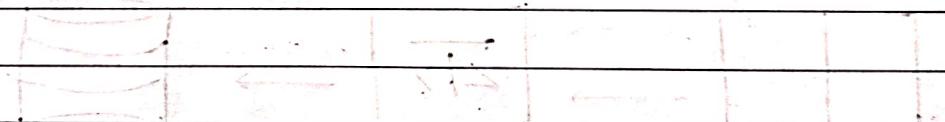
• Architecture of Servlet

• Life Cycle

• HTTP Reqⁿ

• Session Tracking

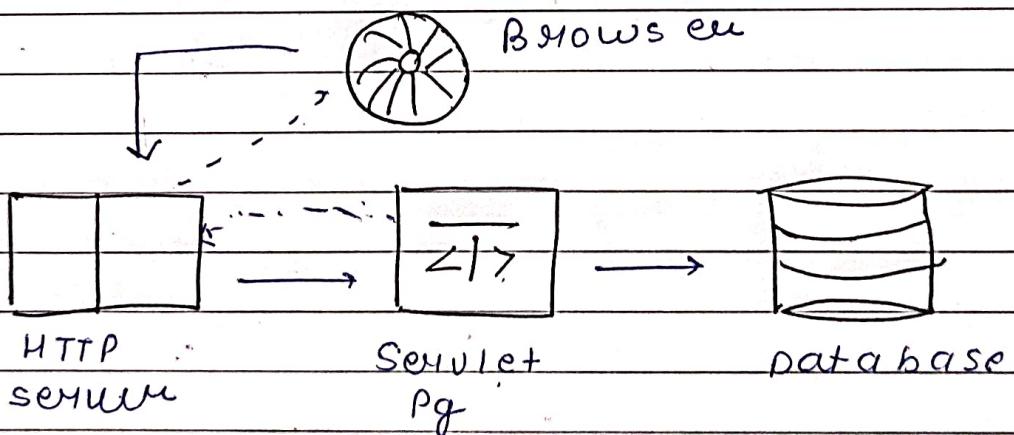
• Cookies



Servlets with its Lifecycle

[AKTU-20-21, 21-22, 22-23]

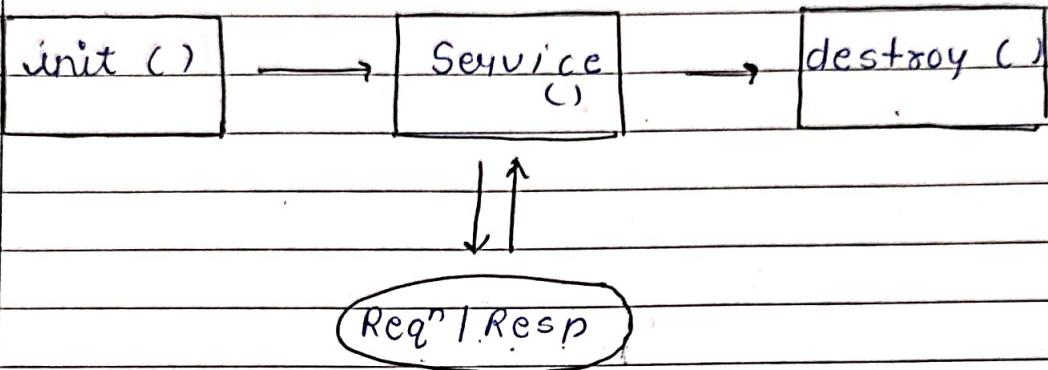
- Servlets are the Java program that run on the Java-enabled web Server or application Server.
- They are used to handle the request obtained from the web-server, process the reqn, produce the response, and then send a response to the web server.
- Servlets work on the server side.



- Servlets are helpful to create dynamic web content and it's more efficient than older CGI scripts.

Lifecycle

- init()
- Service()
- destroy()



1. init()

- The init() method is called only once
- Created only when the servlet created
- The init() method must complete successfully before the servlet can receive any request and response
- when the servlet is first requested, the container creates an instance of servlet class.
- After creation, the servlet container calls the init() method to initialize the servlet.
- This phase is used to set up resources like database connections etc

2. Service()

- The Service() method is called only after the init
- This is the main method to perform the actual task
- It check the HTTP request type
- This phase is repeated for every request

3. destroy()

- The destroy() method is called only once at the end of the life cycle of a servlet
- This method is used to release resources like closing database connection

HTTP Methods :-

Client → DB

1. Get :- To receive data.

2. Post :- To send or submit data to the server, usually to create something new.

3. Patch :- To Partially update data to the server.

4. Put :- To update existing data on the server.

5. Remove: To delete data from the Server.

Redirecting Request to other Resources

- Redirecting a request in Servlet means sending the user request to another resource to handle it.

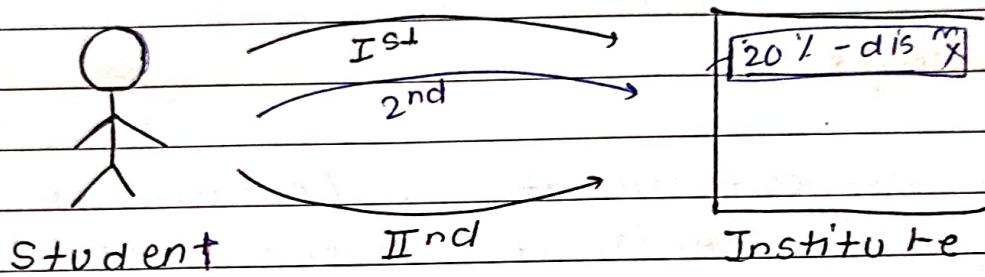
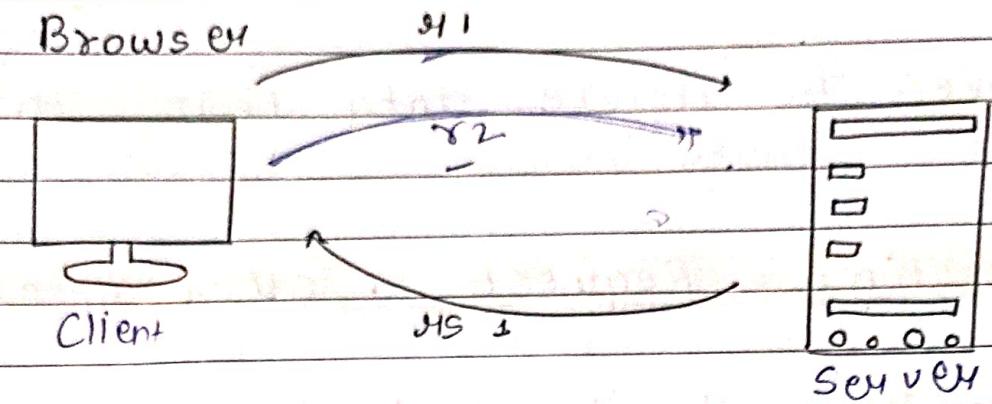
Two-ways :-

1. Using RequestDispatcher (server-side)
2. Using sendRedirect (client-side)

Session Tracking :-

- Session Tracking in Java Servlets is the mechanism to maintain the state of a user interacting with a web application. Since HTTP is a stateless
- Session Tracking is a way to maintain state of an user. It is also known as Session Management.
- HTTP is stateless that means each request is considered as the new reqn.
- To recognize the user.

Date : 1/1



Session Tracking Techniques

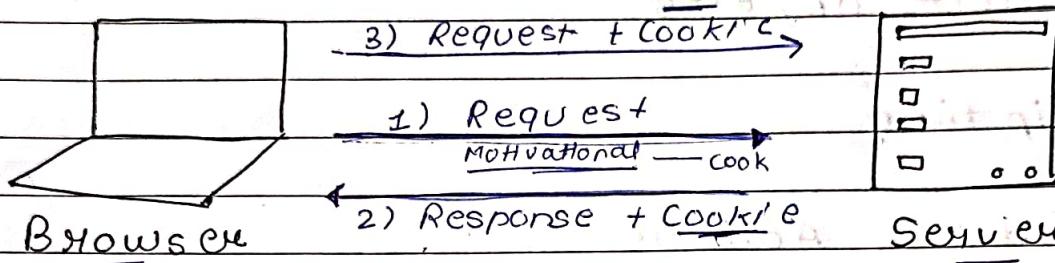
1. Cookies
2. Hidden Form Field
3. URL Rewriting
4. HttpSession

Cookies

- A Cookies is a small piece of information stored on the client.
- Session

Date: ___ / ___ / ___

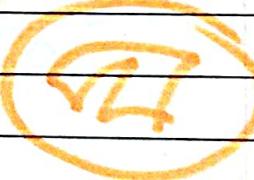
- A Cookie has a name, a single value and optional attributes.
- In cookies technique, we add cookie with response from the Service.



Types of cookie :-

1. Non-Persistent cookie → Valid for Single Session
2. Persistent cookie → Valid for multiple Session.

SUBSCRIBE

JOIN TELEGRAM 

Date : ___ / ___ / ___

UNIT-05 (LEC-2)

JSP AKTU-20-21, 21-22, 22-23

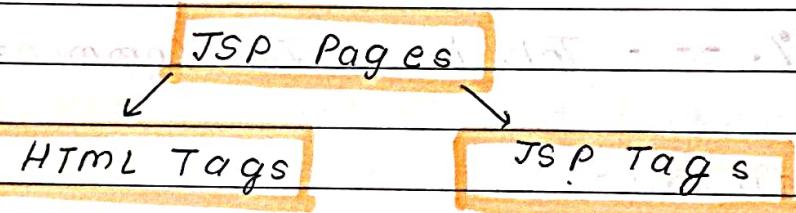
- Implicit Objects
- Scripting
- Standard Actions
- Directives
- Custom tag Libraries

SUBSCRIBE 

JOIN TELEGRAM 

JSP (Java Server Page) [AKTU-20-21, 21-22, 22-23]

- JSP is a serverside technology that is used for creating web application. It is used to create dynamic web content.
- It supports both Scripting and element based dynamic content
- JSP is an advanced version of Servlet Technology



- JSP tags are reusable than pages even faster
- JSP and CGI purpose is same but JSP offers several advantages in comparison with CGI
- Java Code can be inserted in HTML Page or both. JSP allows embedding Java Code in HTML Pages.

Features in JSP :

- Coding in JSP is Easy
- Easy to Use and Learn
- It Does Not Require Advanced knowledge of Java

Tags and Elements in JSP :-

1. JSP Comment :-

- JSP comment is to document the code
- JSP comment is used to note some parts of JSP pages to make it clearer and easier to maintain

<%-- This is a JSP Comment ---%>

2. Expression :-

- Basic Scripting elements in JSP
- expression is used to insert value directly to the output

<% = Expression %>

3. Scriptlet tag :-

- Insert any plain Java code inside a scriptlet

<% Java-Code %>

4. Declaration tag :-

We can declare static members, instance variable and methods inside declaration tag.

<%! declaration %>



Qn JSP is an extension of Servlets not replacement Justify? What problems of Servlets technology can JSP suggest to solve? [AKTU-20-21]

Sol. JSP is not a replacement of Servlets but extension of Servlet, as coding decreases more than half

- In JSP, static code and dynamic code are separated

- JSP needs no compilation by the programmer

Problem of Servlet technology

Solved by JSP :

- Difficult to code
- It cannot be integrated with
- It does not manage Cookies
- Do not all reading, and sending HTML headers

Implicit Object :

- Implicit Objects are a set of Java Objects that the JSP Container make available to developers on each Pages.
- The JSP request is an implicit Object which is provided by HttpServletRequest.
- JSP request Object is created by the web container for each request of the client.

Total Nine Implicit objects as follows :

1. Request
2. Response
3. Config

Date: ___ / ___ / ___

- 4 Application
- 5 Session
- 6 Page Context
- 7 Page Object
- 8 Exception
- 9 out

Standard Actions :-

- In JSP, Standard actions are special JSP tags that are used to perform common tasks like Forwarding request, including files or interacting with JavaBeans.

1. <jsp:include>

- To include another JSP

Ex:- <jsp:include page="header.jsp" />

2. <jsp:forward>

- Forward the request to another JSP

Ex:- <jsp:forward page="nextPage.jsp" />

3. <jsp:useBean>

- Creates or retrieves a JavaBean object.

Ex:- <jsp:useBean id="" class="" scope="" />

Date : ___ / ___ / ___

4. `<jsp: setProperty>`

- Sets properties of a Java bean

Ex :- `<jsp: setProperty name="" Prop="" Value="" />`

5. `<jsp: getProperty>`

- Retrieves a property value from a Java Bean

Ex :- `<jsp: getProperty name="" Property="" />`

6. `<jsp: param>`

- Passes parameters to ~~another~~ included page or a forwarded resource

Ex :- `<jsp: param name="" Value="" />`

Directives :-

- JSP Directives are instructions to the JSP Container that affect the entire JSP page, like how it is translated into a Servlet.

- These directives are not used.

directly in servlets because servlets do not go through a translation process.

3-types of directives :-

1. Page directive <%@ page attribute = " " %>
2. Include directive <%@ include file = " " %>
3. Taglib directive <%@ taglib uri = " " %>

Custom Tag Libraries :-

- Custom Tag Libraries are primarily a feature of JSP and are not directly used in Servlets.
- We are going to create a custom tag that prints the current date and time.
- Custom tag libraries are collection of user-defined tags encapsulating reusable functionality.

Ex :- <%@ custom : welcome name = " " %>

How Tag Libraries work :

1. Tag Handler classes:

Java classes that define the behaviour of custom tags

2. Tag Library Description:

XML files that maps custom tags to their handler classes

3. Usage in JSP:

Custom tags are used in JSP with `<%@ taglib %>` directives

Custom Tags in JSP Imp. Points:

- Declarative, like HTML
- High, encapsulated in tag
- writing plain Java code

SUBSCRIBE 

JOIN TELEGRAM 