Assignment-04

Servlet Jsp

Servlet

2. What is servlet?

Ans. Servlet technology is used to create a web application (resides at server side and generates a dynamic web page).

Servlet can be described in many ways, depending on the context.

- Servlet is a technology which is used to create a web application.
- Servlet is an API that provides many interfaces and classes including documentation.
- Servlet is an interface that must be implemented for creating any Servlet.
- Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
- Servlet is a web component that is deployed on the server to create a dynamic web page.

3. How PrintWriter works?

Ans. In a servlet, PrintWriter is used to send text data (like HTML) back to the client (usually a web browser). Here's a simple explanation of how it works in the context of a servlet:

Setting Up the Servlet

- Create a Servlet: A servlet is a Java class that extends HttpServlet and overrides methods like doGet() or doPost().

Writing with PrintWriter

- Get the PrintWriter: In the servlet's doGet() or doPost() method, you get the PrintWriter from the HttpServletResponse object. This PrintWriter is used to send data back to the client.

- Write Data: Use the print() or println() methods of PrintWriter to write HTML or other text data.

4. What is servlet architecture?

Ans.Servlets are Java programs that run on a web server and handle requests from web clients (like browsers), process those requests, and send back responses. The architecture of servlets involves several key components and a lifecycle that makes this possible. Here's an easy-to-understand explanation of servlet architecture:

- Can easily manage/control the application flow.
- Suitable to implement business logic.
- Can effectively balance the load on the server side.
- Easily generate dynamic web content.
- Handle HTTP Request and Response
- Also act as an interceptor or filter for a specific group of requests.

5. What are life cycle methods of Servlet?

Ans.Servlets have a life cycle managed by the servlet container (like Tomcat). The life cycle consists of several methods that define the stages a servlet goes through from creation to destruction. Here are the main life cycle methods:

- init()
- service()
- doGet() / doPost()
- destroy()

6. What is difference between doGet and doPost()?
Ans.doGet() and doPost() are methods in a servlet that handle HTTP GET and POST requests respectively. Here's a simple explanation of their differences:

1. Purpose
doGet(): Handles HTTP GET requests.
doPost(): Handles HTTP POST requests.

2. Data Transmission
doGet(): Sends data as part of the URL (query string).
Example URL: http://example.com/servlet?name=John&age=30
Suitable for: Sending small amounts of data, bookmarking, and search queries.

doPost(): Sends data in the body of the request.
Data is not visible in the URL.
Suitable for: Sending larger amounts of data, form submissions, and sensitive information.

3. Security
doGet(): Less secure as data is visible in the URL.
Not ideal for sensitive data like passwords.
doPost(): More secure as data is not exposed in the URL.
Better for sensitive data and larger data submissions.

7. When does destroy get called?
Ans.The destroy() method in a servlet is called when the servlet is being taken out of service, and it's time to clean up resources. Here's a simple explanation of when and why destroy() gets called:

Purpose of destroy()
Cleanup: To release resources such as database connections, file handles, and background threads that were allocated during the servlet's lifecycle.

When destroy() Gets Called
- Server Shutdown: When the web server or servlet container (like Tomcat) is shutting down.
- Servlet Removal: When the servlet is being removed from service, either because it's being redeployed or the server is stopping.
- Servlet Reloading: When the servlet is being reloaded, usually due to changes in the servlet's code or configuration, the old instance is destroyed before the new instance is created.

8. What are ways to. Implements servlet ?
Ans.There are two main ways to implement a servlet in Java: by extending the HttpServlet class or by implementing the Servlet interface. Here's a simple explanation of each method:

1. Extending HttpServlet
This is the most common and straightforward way to create a servlet. You extend the HttpServlet class and override methods like doGet() and doPost() to handle HTTP GET and POST requests.

Steps to Implement
- Extend HttpServlet: Create a class that extends HttpServlet.
- Override Methods: Override doGet(), doPost(), or other relevant methods to handle HTTP requests.
- Configure Servlet: Configure the servlet in the web application deployment descriptor (web.xml) or using annotations

2. Implementing Servlet Interface
This method involves directly implementing the Servlet interface. It requires you to implement all five methods of

the interface (init(), service(), doGet(), doPost(), and destroy()).

Steps to Implement
- Implement Servlet Interface: Create a class that implements the Servlet interface.
- Implement Methods: Implement all the methods of the Servlet interface.
- Configure Servlet: Configure the servlet in the web application deployment descriptor (web.xml) or using annotations.

9. What are http methods?
Ans.HTTP methods are the actions or verbs that indicate what you want to do with a resource on the server when making a request. They define the operation to be performed and are an essential part of the HTTP protocol used for communication between clients (such as web browsers) and servers. Here are the main HTTP methods:

1. GET
- Purpose: This method retrieves information from the given server using a given URI. GET request can retrieve the data. It can not apply other effects on the data.

2. POST
- Purpose: The POST request sends the data to the server. For example, file upload, customer information, etc. using the HTML forms.

3. PUT
- Purpose: The PUT method is used to replace all the current representations of the target resource with the uploaded content.

4. DELETE
- Purpose:The DELETE method is used to remove all the current representations of the target resource, which is given by URI.

5. HEAD
- Purpose:This method is the same as the GET method. It is used to transfer the status line and header section only.

6. OPTIONS
- Purpose: Describe the communication options for the target resource.

10. What http method used by Servlet?
Ans.Ans.In the context of servlets (which are Java programs that run on a server and handle requests from clients, typically web browsers), HTTP methods are ways of indicating the desired action to be performed on the server. Here are the most common HTTP methods:

1. GET
- Purpose: Retrieve data from the server.

2. POST
- Purpose: Send data to the server to create/update resources.

3. PUT
- Purpose: Update a resource on the server.

4. DELETE
- Purpose: Remove a resource from the server.

5. HEAD
- Purpose: Retrieve the headers from the server without the body.

6. OPTIONS

- Purpose: Describe the communication options for the target resource.

11. What is difference between Generic Servlet and HttpServlet?
Ans.GenericServlet
Purpose: GenericServlet is a general-purpose servlet class provided by Java. It is protocol-independent, which means it can be used with any protocol (not just HTTP).

Methods: It defines the basic methods that all servlets must implement, such as service(), init(), and destroy(). It doesn't have built-in support for handling specific HTTP methods like GET or POST.

Usage: You would use GenericServlet if you were dealing with non-HTTP protocols, although this is quite rare. Most web applications use HTTP, so HttpServlet is more commonly used.

HttpServlet
Purpose: HttpServlet is specifically designed for HTTP protocol, which is the protocol used by the web.

Methods: It extends GenericServlet and adds methods for handling HTTP-specific tasks, such as doGet(), doPost(), doPut(), doDelete(), and doHead().

Usage: This is the most commonly used servlet for web applications because it directly supports HTTP requests and responses.

12. What is the use of RequestDispatcher Interface?
Ans.The RequestDispatcher interface in Java's Servlet API is used to forward a request from one servlet to another resource (which could be another servlet, a JSP file, or an HTML file) on the server, or to include the content of another resource in the response.

13. What is the difference between ServletConfig and ServletContext?
Ans.ServletConfig and ServletContext, both are objects created at the time of servlet initialization and used to provide some initial parameters or configuration information to the servlet. But, the difference lies in the fact that information shared by ServletConfig is for a specific servlet, while information shared by ServletContext is available for all servlets in the web application.

ServletConfig:
- ServletConfig is an object containing some initial parameters or configuration information created by Servlet Container and passed to the servlet during initialization.
- ServletConfig is for a particular servlet, that means one should store servlet specific information in web.xml and retrieve them using this object.

ServletContext:

- ServletContext is the object created by Servlet Container to share initial parameters or configuration information to the whole application.

14.What do you mean by InterServlet communication?
Ans.A process where two or more servlets communicates with each other to process the client request.
A servlet can forward the request to another servlet to process the client request.
A servlet can include the output of another servlet to process the client request.

- Inter-servlet communication using Request Dispatcher
- A Request Dispatcher:
- Is an object of the javax.servlet.RequestDispatcher interface that allows inter-servlet communication.
- Object is used to include the content of another servlet.
- Object is used to forward the request to another servlet.

15. What is significance of web.xml?
Ans.web. xml defines mappings between URL paths and the servlets that handle requests with those paths. The web server uses this configuration to identify the servlet to handle a given request and call the class method that corresponds to the request method.

16. Explain Web Container?
Ans.A web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access-rights. A web container handles requests to servlets, Jakarta Server Pages (JSP) files, and other types of files that include server-side code.

17. What do you mean by the Servlet Chaining?
Ans.Taking the request from a browser window and processing that request by using multiple servlets as a chain is called servlet chaining. In servlet chaining, we perform communication between servlet programs to process the request given by a client.

18. Why do we use sendredirect() method?
Ans.The sendRedirect() method of HttpServletResponse interface can be used to redirect response to another resource, it may be servlet, jsp or html file.

It accepts relative as well as absolute URL.

It works at client side because it uses the url bar of the browser to make another request. So, it can work inside and outside the server.

19. What Servlet filters?
Ans.Servlet filters are components in the Java Servlet API that allow you to intercept and modify requests and responses in a web application. They provide a powerful way to handle various common tasks in a centralized manner before the request reaches a servlet or after the response leaves a servlet.

20. When to use Servlet filter?
Ans.Servlet Filters are Java classes that can be used in Servlet Programming for the following purposes –

- To intercept requests from a client before they access a resource at back end.
-
- To manipulate responses from server before they are sent back to the client.

Jsp

21. How do we translate JSP?
Ans.JSP (JavaServer Pages) is a server-side technology that allows you to embed Java code within HTML pages. When a JSP page is requested, the web server (such as Apache Tomcat) translates the JSP code into a servlet, which is then executed by the Java Virtual Machine (JVM).

22. What is JSP ?
Ans.JSP (JavaServer Pages) is a server-side technology that allows you to create dynamic web pages that interact with databases, perform server-side logic, and respond to user input. It's a Java-based technology that enables you to embed Java code within HTML pages, making it a powerful tool for building web applications.

23. Why do you use JSP?
Ans.It is used to create dynamic web content. JSP consists of both HTML tags and JSP tags. In this, JSP tags are used to insert JAVA code into HTML pages.

24. What are implicit object in JSP?
Ans.In JSP (JavaServer Pages), implicit objects are pre-defined objects that are automatically available to a JSP page without the need to declare or instantiate them. These objects provide access to various aspects of the JSP environment, such as the request, response, and session.

Here are the some implicit objects in JSP:

request: Represents the HTTP request object, which contains information about the client's request.
response: Represents the HTTP response object, which is used to send the response back to the client.
out: Represents the output stream, which is used to write data to the response.
session: Represents the HTTP session object, which stores information about the user's session.
application: Represents the servlet context, which provides access to the application's configuration and resources.
config: Represents the servlet configuration, which provides access to the servlet's initialization parameters.
page: Represents the current JSP page, which provides access to the page's properties and methods.
pageContext: Represents the page context, which provides access to the page's context and its implicit objects.
exception: Represents the exception object, which is used to handle exceptions in the JSP page.

25. What are scriptlet in JSP?
Ans.In JSP (JavaServer Pages), a scriptlet is a block of Java code that is embedded directly into a JSP page. Scriptlets are used to execute Java code on the server-side, allowing you to perform dynamic operations, manipulate data, and interact with the JSP environment.

26. What are directive?
Ans.In JSP (JavaServer Pages), a directive is a special instruction that is used to convey information to the JSP engine about how to process the JSP page. Directives are used to control the behavior of the JSP engine, specify page settings, and import external resources.

There are three types of directives in JSP:
page directive: Used to specify page-level settings, such as the language, contentType, and errorPage.

include directive: Used to include a file or a JSP page at the current location.

taglib directive: Used to import a custom tag library and make its tags available for use in the JSP page.

27. How to execute Java code in Jsp?
Ans.In JSP (JavaServer Pages), you can execute Java code in several ways:
1. Scriptlets
Scriptlets are blocks of Java code that are embedded directly into a JSP page. They are enclosed within <% and %>

tags.

```
<%
  // Java code goes here
  int x = 10;
  out.println("Hello, World!");
%>
```

2. Expressions:
Expressions are used to evaluate a Java expression and output its result. They are enclosed within <%= and %> tags.

```
<%= "Hello, " + request.getParameter("username") %>
```

3. Declarations

Declarations are used to declare variables and methods that can be used throughout the JSP page. They are enclosed within <%! and %> tags.

```
<%! int x = 10; %>
```

4. JavaBeans

JavaBeans are reusable Java classes that can be used to encapsulate complex logic and data. You can use JavaBeans in JSP pages to execute Java code.

28. Why jsp is in when Servlet can do everything what jsp can do?
Ans.We use because JSPs allow for a clear separation of concerns between presentation logic (HTML, CSS, JavaScript) and business logic (Java code). This makes it easier to maintain and update the presentation layer without affecting the underlying business logic.

29. What is lifecycle phases of JSP?
Ans.The JSP lifecycle involves translation, compilation, loading, initialization, request processing, and destruction similar to a servlet lifecycle. The document provides an introduction to Java Server Pages (JSP) which is a server-side technology used to create dynamic web pages.

30. What are the method used here?
Ans. 1. jspInit()
2. _jspService()
3. jspDestroy()
4. jspInit(ServletConfig config)
5. jspDestroy(ServletConfig config)