

## Assignment 4.

1. Define Servlet. Give the feature of it.

→ Ans:

- Servlets offers a wide range of advantages for web developers who are building dynamic scalable and secure web application.
- Servlets is a Java based program that runs on a web server and is used to handle client requests and generate dynamic web pages.

### \* Features:

- i) Servlets are platform-independent and can be deployed on any web server that supports Servlet API.
- ii) Servlets can handle large amount of requests and can be scaled easily by deploying them.
- iii) Servlets can be reused in multiple web application.
- iv) Servlets can maintain client session, which allows the server to track user activity and maintain state across multiple requests.
- v) Servlet can implement security mechanism.

to restrict access to web resources and protect against attacks like cross site scripting and cross site request forgery.

2. Explain the Advantages of Servlet.

- Ans:
- There are servlet advantages:
  - Platform-Independent:
    - Servlet are written in Java and can be deployed on any web server that supports the Servlet API, regardless of the operating system or hardware platform.

iii) High performance:  
• Servlets are highly optimized for performance and can handle a large number of requests simultaneously. They also use minimal resources.

→

iii) Easy integration:

- Servlet can be easily integrated with other Java technologies, such as JSP, JDBC and EJB.

iv) Session management:

- Servlet provide built-in session management capability, allowing them to across web application from that maintain state across multiple users.

→

v) Deployment:

→

v) Scalability:

- Servlet provide robust security features including authentication, authorization and encryption which helps protect web application against threats.

vi) Security:

- Servlet provide robust security features including authentication, authorization and encryption which helps protect web application against threats.

vii) Scalability:

- Servlet can be easily scaled horizontally or vertically to handle increasing amount of traffic, making them ideal for large scale web application.

3. Differentiate between Servlet vs Applet

→

Ans: Servlet and Applet are

i) Components used for handling client side components used for running client requests and generating dynamic web pages.

ii) Servlets are deployed on a web server and run within web browser.

public String getServletInfo() {  
 return "myServlet";  
}

\* Example:  
import javax.servlet.\*;  
public class MyServlet extends HttpServlet {  
 public void doGet(HttpServletRequest req, HttpServletResponse res) throws  
 ServletException, IOException {  
 //Code  
 }  
}

- ii) Using GenericServlet Abstract Class.  
- To create a Servlet, you need to Create a Java class that extends the genericServlet class and overrides its service() method.

\* Example:

```
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
  
public class MyServlet extends GenericServlet {  
    public void service(HttpServletRequest req, ServletResponse res) throws  
        ServletException, IOException {  
        //Code  
    }  
}
```

- Q. What is Session Tracking, explain Types of Session Tracking with example  
Ans:  
- Session tracking is the process of maintaining information across multiple requests from a client to a web server.

- These are several ways to perform Session tracking.  
i) Cookie based session tracking.  
- It involves storing a unique session ID in a cookie on the client side, which is then sent back to the server with each request.

- ii) Using the HttpSession Class.  
- To create extends the HttpServlet Class and override its doGet() or doPost() method.

- \* The server can then use this session id to retrieve the session data associated with that particular client.

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 resources for example memory, threads, etc.

Q. Differentiate GET and POST method requests.

Ans:

GET	POST
Only limited data can be sent because data is sent in header	Large amount of data is sent because data is sent in body.

IV) GET request is not secure. POST request is secure because data is exposed in URL.

V) GET request can be bookmarked.

VI) GET request is more efficient than POST request.

VII) GET request is less efficient than POST request.

VIII) GET request is idempotent. POST request is non-idempotent.

Q. Explain three ways to implement Service with example.

Ans:

i) Using the servlet interface.

- To implement a service using this interface, we need to create a Java class that implements the Servlet interface and overrides its method.

\* Example:

```
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
  
public class MyServlet implements Servlet {  
    private ServletConfig config;
```

public void init(ServletConfig config) throws ServletException {  
 this.config = config;

...  
}

```
public void service(ServletRequest req, ServletResponse res) throws  
    ServletException, IOException {  
    ...  
}
```

return config;

```
    ...  
}
```

```
    ...  
}
```

```
    ...  
}
```

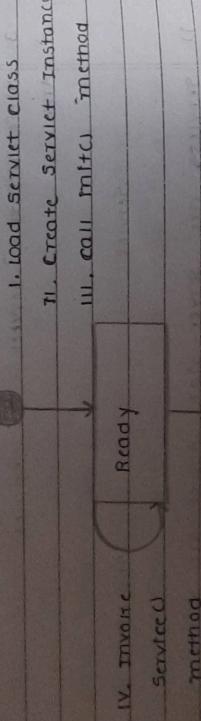
```
    ...  
}
```

return config;

```
    ...  
}
```

- iii) Servlets have access to the system's resources
- iv) Servlet typically do not have a graphical user interface
- v) Servlet generates HTML or Applet download their code.
- vi) Other types of contents and resources from the server over the network that are sent over the network to the client.

4. Describe the life cycle of a service  
→ Ans:  
→ There are 5 stages of service life cycle



- i) 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.
- ii) 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.
- iii) init method is invoked.
- The web container calls the init method only once after creating the servlet instance.
  - cc) The init method is used to initialize the servlet.
- iv) Service method is invoked.
- The web container calls the service method each time when request for the servlet is received.
  - If the servlet is not initialized the first three steps are described above them calls the service method.
- v) destroy method is invoked.
- If servlet is initialized it calls the service method.
  - Only once.

\* Example

```
// Create session
HttpSession s = request.getSession();
String name = (String)s.getAttribute("username");
String sessionId = s.getId();
```

- It involves appending a unique session ID to the URL in the response that the server sends to the client.
- The client then includes this session ID in subsequent requests to the server.
- The server can then use this session ID to retrieve data.

\* Example:

```
// Create session ID
String sessionId = request.getSession().getId();
```

// Append to URL  
Response.sendRedirect('http://exam.com/myapp/' + page.jsp; sessionId);

III) Hidden form field

- It involves including a hidden field in an HTML form that stores the unique session ID.
- When the client submits the form, the session ID sent back to the server along with form data.

\* Example:

```
// Create a hidden form field
out.println("<form method='post' action='form.jsp'>");  
out.println("<input type='hidden' name='SID' value='");  
out.println(session.getId());  
out.println("</input>");  
out.println("<input type='text' name='username'>");  
out.println("<input type='submit' value='Submit'>");  
out.println("</form>");
```

8. How HTTP service handles client request?

- Ans.
- When an HTTP service receives a client request, it goes through a series of steps to handle the client request.

\* Steps of handling client request

- i) The client sends an HTTP request to the server specifying the URL of the resources it wants to access.

v) The container receives the request and maps it to the appropriate Servlet based on the URL.

Ans:

vi) The Servlet Container creates a new HttpServletRequest object to encapsulate all the Request data and a new HttpServletResponse object to represent the response that will be sent back to the client.

vii) The Servlet's service() method examines the request and determines the appropriate action to take. This involves reading the request parameters, headers and cookies and performing some business logic based on the data.

viii) The servlet generates the response by setting the appropriate headers, cookies and response body data and then sends the response back to the client using the HttpServletResponse object.

ix) If the response includes any cookies, the Servlet Container adds them to the set cookie header of the response.

x) The Servlet Container class then connects to the client and the request-response cycle is complete.

q. List Interface and Class available in Servlet API.

Ans:

\* Interfaces:  
- Servlet - HttpSession  
- ServletConfig - HttpSessionBindingListener  
- ServletContext - HttpSessionListener  
- Filter - ServletContextListener  
- FilterConfig - ServletRequest  
- FilterInputStream - ServletResponse

\* Classes:  
- GenericServlet  
- HttpServlet  
- HttpServletDoGet  
- HttpServletRequestWrapper  
- HttpServletResponseWrapper  
- ServletInputStream  
- ServletOutputStream

Ans:  
i) Describe web container and its working.

A web container, also known as a Servlet container or a web server, is a component of a web server that provides the environment in which Java Servlet and JSP can be run.

\* warning: Standard working of JSP

i) Recieve Requests:



The web container receives requests from clients.

ii. Explain web.xml file with its attributes.

Ans:

i) MAP Requests:  
The web container maps the requests to the URL.

- The web container maps the requests to the URL based on the URL pattern specified in the web.xml file.

ii) Create object:  
The web container creates a new instance of the Servlet or JSP to handle the object.

iii) Initialize object:  
The web container initializes the Servlet or JSP by calling init() method.

iv) Invoke method:  
Calls the appropriate methods of the Servlet or JSP such as doGet(), doPost() or service() to process the request.

v) Generate response:  
by setting headers, cookies and body content.

vi) Servlet:  
defines Servlet that handles request from Client.

vii) Send response:  
sends the response back to Client.

viii) Destroy objects:  
No longer needed it calls destroy() method.

ii) Explain web.xml file with its attributes.

- The web.xml file is a deployment descriptor for Java Web applications that provides information to the web container about the application's configuration, Servlets, JSP and security settings.

iii) web-app:  
- This is root element of the web.xml file and defines the version of the web application and its name.

iv) display-name:  
- This element specifies the name of the web application that is displayed in the browser.

v) description:  
- provides a description of the web application.

vi) Servlet:  
it specifies the name of the Servlet class, URL pattern, and initialization param.

vii) Servlet-mapping:  
maps Servlet to URL pattern.

Page  
Date

Page  
Date

i) Go to <https://www.oracle.com/java/technologies/>

Java - Servlet - San- downloads.html.

ii) Accept the licence agreement and select the appropriate version for OS.

iii) Download and install above selected version for OS.

iv) Once the download is complete, extract the contents of the archive to a suitable location on your system.

v) Set the environment variables required for ISDK by adding the ISDK installation direct

try to your System's path variable

vi) Verify that the installation is complete by opening cmd and typing

javac -version

vii) Now you can use to develop and deploy Java services and ISDK.

12. List installation steps of Java Servlet Development Kit.

→ Ans :  
To install Java Servlet development Kit (JSRDK)

\* Steps :  
1. Download Java Server API  
2. Extract Java Server API  
3. Install Java Server API  
4. Set Environment Variables  
5. Verify Java Server API Installation