Servlet

In Java, a **Servlet** is a **Java class used to handle HTTP requests and generate responses**, typically used in web applications. It runs on a **Servlet container** (like Apache Tomcat) inside a web server.

🔹 What Does a Servlet Do?

A servlet acts as a **middle layer between a client (like a web browser) and the server**:

- It **receives a request** (usually HTTP GET or POST),
- **Processes the request** (e.g., interacts with a database, performs logic),
- **Generates a response** (often HTML, JSON, etc.), and
- **Sends the response back** to the client.

By Eslam Khder

Servlet

Why Are Servlets Important?

Servlets are foundational to Java web development. Technologies like:

- **JSP** (JavaServer Pages),
- **Spring MVC**,
- **JavaServer Faces (JSF)**,

...are built on top of servlets.

By Eslam Khder

Servlet

the context of Java Servlets, a **session** is a way to maintain state (data) across multiple HTTP requests from the same client.

By Eslam Khder

Servlet

🔹 Why do we need sessions?

HTTP is a **stateless protocol**, meaning each request from a client to the server is independent and the server does not remember previous requests. However, many web applications need to remember information about users across multiple requests — for example:

- User login status
- Shopping cart contents
- User preferences during a browsing session

This is where **sessions** come in.

By Eslam Khder

Servlet

🔹 What is a Session in Servlet?

A **session** is an object managed by the servlet container that stores data about a user's interaction with a web application across multiple requests.

Servlets use the `HttpSession` interface to create and manage sessions.

By Eslam Khder

Servlet

- **How does session work?**

1. When a user visits a web application for the first time, the server creates an **HttpSession** object.
2. The server sends back a **session ID** (usually in a cookie called `JSESSIONID`) to the client.
3. On subsequent requests, the client sends this session ID back to the server.
4. The server uses this session ID to retrieve the stored session data.

Servlet

- **Summary:**
- A **session** lets you store user-specific data between HTTP requests.
- It's identified by a **session ID**.
- Implemented with the `HttpSession` object in servlets.

By Eslam Khder

Servlet

🍪 What Are Cookies?

Cookies are small pieces of data (text) that a web server sends to a client's browser. The browser stores these cookies and sends them back to the server with each subsequent request to the same server.

By Eslam Khder

Servlet

🍪 Why Are Cookies Used?

Cookies help **maintain state** and **store information** about the user across multiple HTTP requests because HTTP itself is stateless.

Common uses include:

- Keeping a user logged in
- Tracking user preferences
- Storing session IDs
- Tracking user behavior for analytics

By Eslam Khder

Servlet

🍪 Cookies vs Sessions

- **Cookies** are stored **on the client (browser)**.
- **Sessions** are stored **on the server**, and sessions usually rely on cookies to store a session ID that links the client to the server-side session.

By Eslam Khder

Servlet

🍪 Summary

- Cookies store small amounts of data **on the client**.
- They help remember info across HTTP requests.
- Sessions often use cookies internally to track users.

By Eslam Khder

Servlet

✅ Session vs Cookie in Servlet/JSP

Feature	Session	Cookie
Storage Location	Stored on the **server side**	Stored on the **client side** (in the browser)
Data Storage	Stores objects (Java objects, strings, etc.)	Stores only **string data**
Security	More secure (data not visible to client)	Less secure (can be viewed and modified by client)
Size Limit	No strict limit (depends on server memory)	Limited (usually ~4KB per cookie)
Lifetime	Ends when session times out or user logs out	Can persist after the browser is closed (if set to do so)
Creation	Created via `HttpSession session = request.getSession();`	Created via `Cookie c = new Cookie("name", "value");`
Usage Example	`session.setAttribute("user", userObj);`	`response.addCookie(c);`
Access	`session.getAttribute("user");`	`request.getCookies();`
Best Use Case	Storing sensitive or large data like user login info	Storing small preferences (e.g., language, theme)

EraaSoft

By Eslam Khder