Advanced Programming

Week 13

Servlets

- Setting up Apache Tomcat
- RequestDispacher in Servlet
- Session Tracking

Tilahun Melak(PhD)

November, 2025

Objectives

At the end of this lecture, students will be able to:

- Explain how to install and configure the Apache Tomcat server by setting environment variables and managing the server using startup and shutdown commands.
- Explain the purpose and functionality of the RequestDispatcher interface.
- Demonstrate how it enables servlet collaboration through request forwarding and content inclusion.
- Explain the concept of a session in web applications and describe how session tracking maintains user state in stateless HTTP communication.

Setting up Apache Tomcat Server

1. Install Apache Tomcat in your computer

2. Define environment variable JAVA_HOME

   - JAVA_HOME should point to the directory containing your Java installation
   - Eg. C:\Program Files\Java\jdk1.7.0_10

3. Define Environment variable CATALINA_HOME

   - CATALINA_HOME should point to the directory that contains Tomcat
   - Eg. C:\Program Files\Apache Software Foundation\Apache Tomcat 8.0.3

4. To start the Tomcat server, use

   CATALINA_HOME\bin\startup.bat

5. To stop the Tomcat server, use

   CATALINA_HOME\bin\shutdown.bat

Oracle. (2023). Setting up Apache Tomcat server environment. In Java Platform, Standard Edition documentation.

RequestDispatcher in Servlet

- The RequestDispatcher interface provides the facility of dispatching the request to another resource it may be html, servlet or jsp. This interface can also be used to include the content of another resource. It is one of the way of servlet collaboration.
- There are two methods defined in the RequestDispatcher interface.

Oracle. (2023). RequestDispatcher interface (Jakarta Servlet API documentation). In Java Platform, Enterprise Edition documentation.

RequestDispatcher in Servlet…

- **Methods of RequestDispatcher interface**

The RequestDispatcher interface provides two methods. They are:

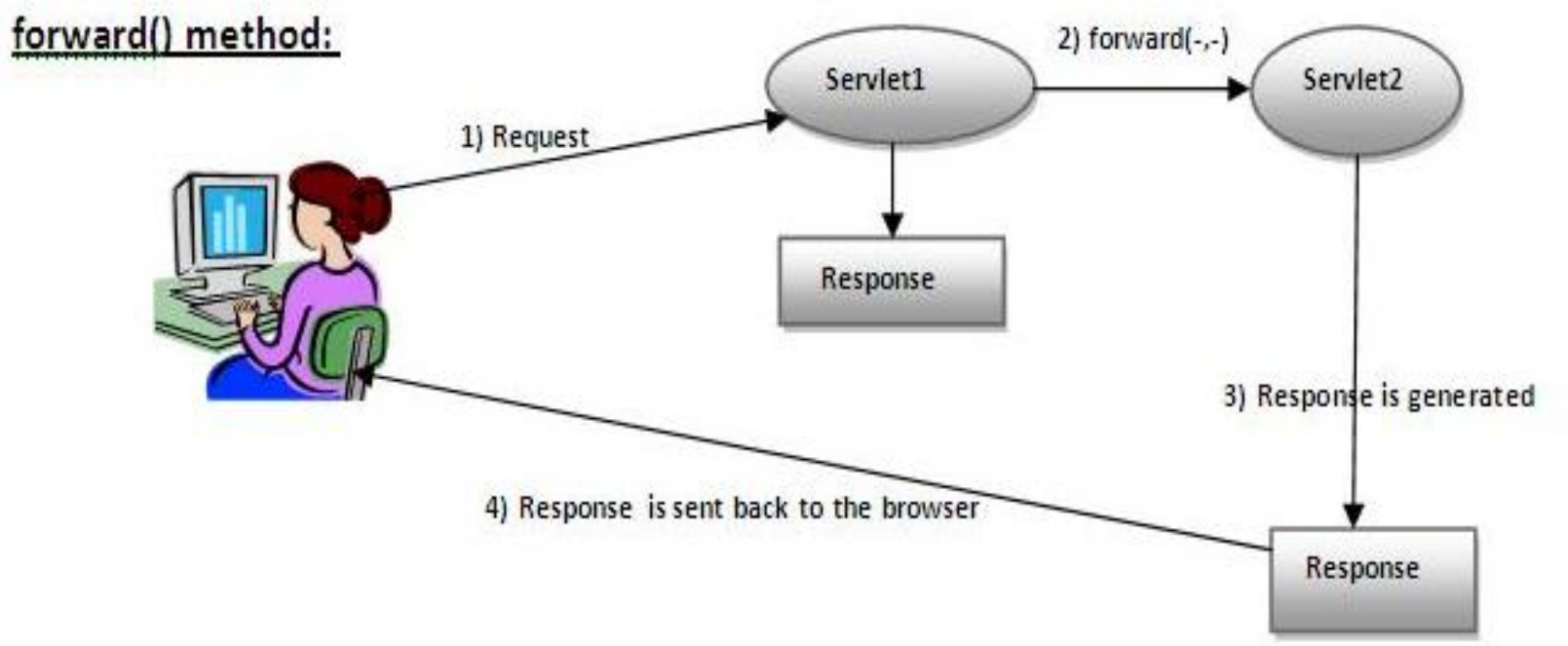
1. **public void forward(ServletRequest request,ServletResponse response)throws ServletException,java.io.IOException:**

Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.

2. **public void include(ServletRequest request,ServletResponse response)throws ServletException,java.io.IOException:** Includes the content of a resource (servlet, JSP page, or HTML file) in the response.

Oracle. (2023). RequestDispatcher interface (Jakarta Servlet API documentation). In Java Platform, Enterprise Edition documentation.

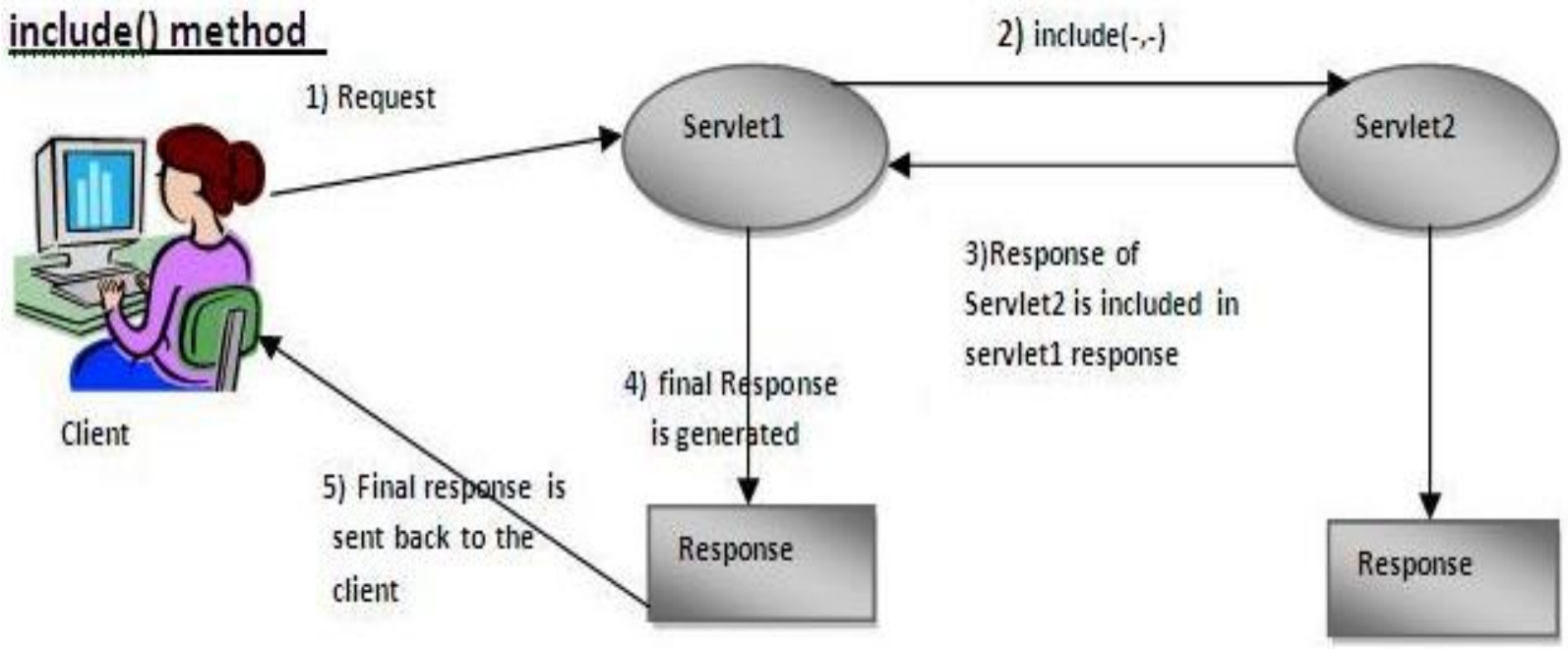
RequestDispatcher in Servlet…



- As you see in the above figure, response of second servlet is sent to the client. Response of the first servlet is not displayed to the user.

Oracle. (2023). RequestDispatcher interface (Jakarta Servlet API documentation). In Java Platform, Enterprise Edition documentation.

RequestDispatcher in Servlet…



- As you can see in the above figure, response of second servlet is included in the response of the first servlet that is being sent to the client.

Oracle. (2023). RequestDispatcher interface (Jakarta Servlet API documentation). In Java Platform, Enterprise Edition documentation.

RequestDispatcher in Servlet…

How to get the object of RequestDispatcher

- The getRequestDispatcher() method of ServletRequest interface returns the object of RequestDispatcher.
- Syntax:

```
public RequestDispatcher getRequestDispatcher(String resource);
```

Example:

```
RequestDispatcher rd=request.getRequestDispatcher("servlet2");
//servlet2 is the url-pattern of the second servlet

rd.forward(request,response);//method may be include or forward
```

Oracle. (2023). RequestDispatcher interface (Jakarta Servlet API documentation). In Java Platform, Enterprise Edition documentation.

SendRedirect in servlet

- 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

Oracle. (2023). HttpServletResponse interface: sendRedirect method (Jakarta Servlet API documentation). In Java Platform, Enterprise Edition documentation

Difference between forward() and sendRedirect() method

- There are many differences between the **forward()** method of RequestDispatcher and **sendRedirect()** method of HttpServletResponse interface. They are given below:

Oracle. (2023). HttpServletResponse interface: sendRedirect method (Jakarta Servlet API documentation). In Java Platform, Enterprise Edition documentation

Difference between forward() and sendRedirect() method

forward() method	sendRedirect() method
The forward() method works at server side.	The sendRedirect() method works at client side.
It sends the same request and response objects to another servlet.	It always sends a new request.
It can work within the server only.	It can be used within and outside the server.
Example: request.getRequestDispacher("servlet2").forward(request,response);	Example: response.sendRedirect("servlet2");

Oracle. (2023). HttpServletResponse interface: sendRedirect method (Jakarta Servlet API documentation). In Java Platform, Enterprise Edition documentation

- **Syntax of sendRedirect() method**

```
public void sendRedirect(String URL)throws IOException;
```

- **Example**

```
response.sendRedirect("http://www.google.com");
```

Oracle. (2023). HttpServletResponse interface: sendRedirect method (Jakarta Servlet API documentation). In Java Platform, Enterprise Edition documentation

Session Tracking in Servlets…

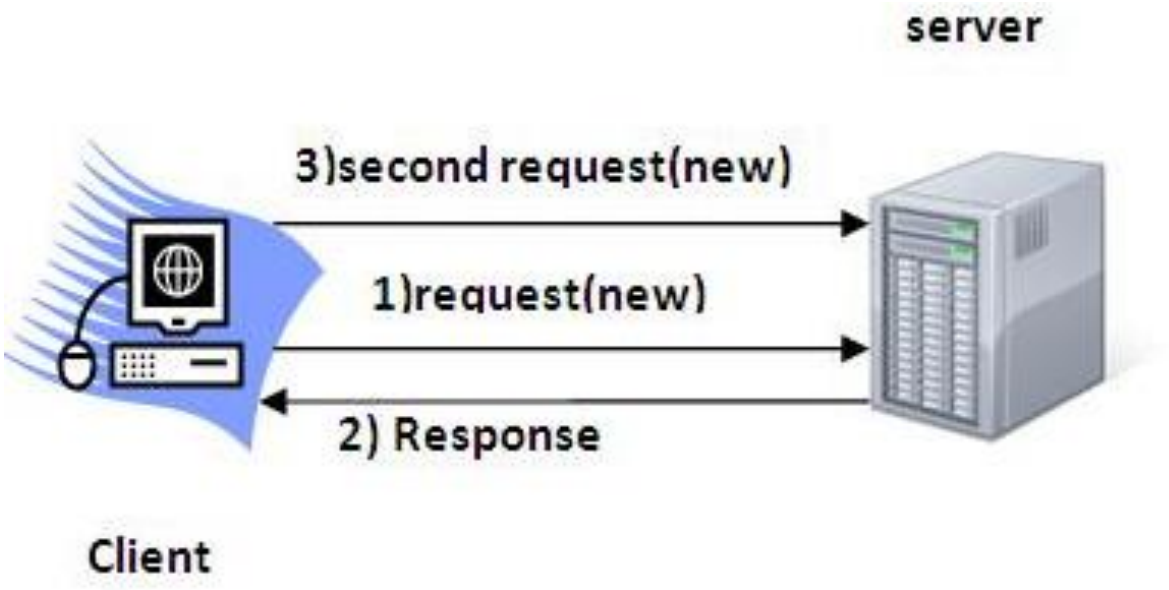
What is a Session?

- **Session** simply means a particular interval of time.
- **Session Tracking** is a way to maintain state (data) of a user. It is also known as **session management** in servlet.
- Http protocol is a stateless so we need to maintain state using session tracking techniques. Each time user requests to the server, server treats the request as the new request. So we need to maintain the state of a user to recognize to particular user.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

- HTTP is stateless that means each request is considered as the new request. It is shown in the figure given below:



Why use Session Tracking?

- **To recognize the user** :It is used to recognize the particular user.

Oracle. (n.d.). *Session Tracking in Servlets*. In *Java Servlet technology* documentation. Oracle.

Session Tracking in Servlets…

Session Tracking Techniques

- There are four techniques used in Session tracking:

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

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

1. **Cookie**

- A **cookie** is a small piece of information that is persisted between the multiple client requests.
- A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

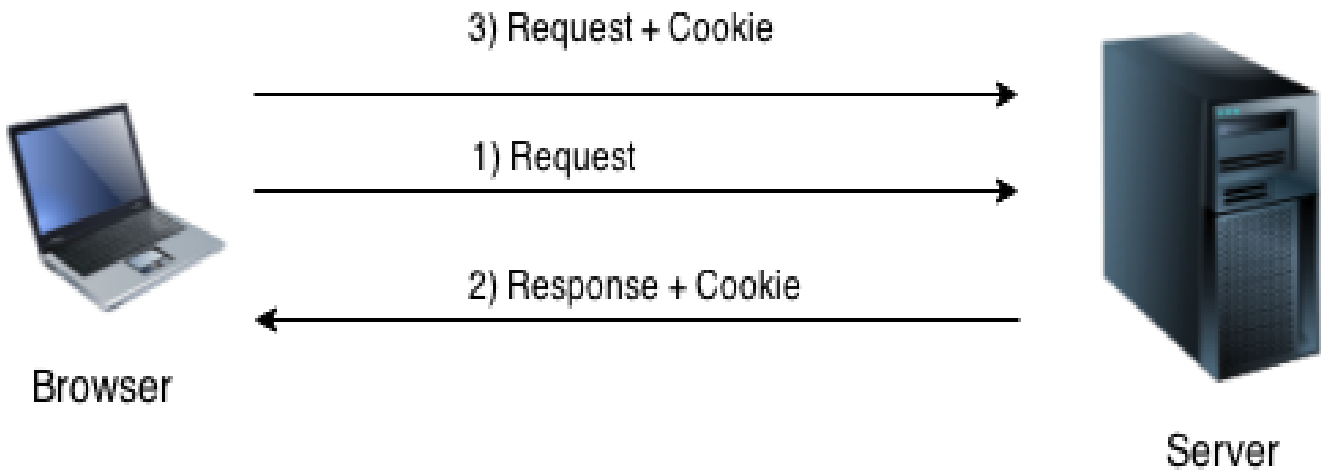
How Cookie works

- By default, each request is considered as a new request. In cookies technique, we add cookie with response from the servlet. So cookie is stored in the cache of the browser. After that if request is sent by the user, cookie is added with request by default. Thus, we recognize the user as the old user.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

1. **Cookie**



Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

1. **Cookie**

Types of Cookie

There are 2 types of cookies in servlets.

1. Non-persistent cookie
2. Persistent cookie

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

1. **Cookie**

- **Non-persistent cookie**

It is **valid for single session** only. It is removed each time when user closes the browser.

- **Persistent cookie**

It is **valid for multiple session** . It is not removed each time when user closes the browser. It is removed only if user logout or signout.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

Advantage of Cookies

1. Simplest technique of maintaining the state.
2. Cookies are maintained at client side.

Disadvantage of Cookies

1. It will not work if cookie is disabled from the browser.
2. Only textual information can be set in Cookie object.

Note: Gmail uses cookie technique for login. If you disable the cookie, gmail won't work.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

Cookie class

- **javax.servlet.http.Cookie** class provides the functionality of using cookies. It provides a lot of useful methods for cookies.

Constructor of Cookie class

Constructor	Description
Cookie()	constructs a cookie.
Cookie(String name, String value)	constructs a cookie with a specified name and value

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

2. Hidden Form Field

- In case of Hidden Form Field **a hidden (invisible) textfield** is used for maintaining the state of an user.
- In such case, we store the information in the hidden field and get it from another servlet. This approach is better if we have to submit form in all the pages and we don't want to depend on the browser.

Let's see the code to store value in hidden field.

```
<input type="hidden" name="uname" value="admin">
```

- Here, uname is the hidden field name and admin is the hidden field value.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

2. Hidden Form Field

Advantage of Hidden Form Field

- It will always work whether cookie is disabled or not.

Disadvantage of Hidden Form Field:

1. It is maintained at server side.
2. Extra form submission is required on each pages.
3. Only textual information can be used.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

3) URL Rewriting

- In URL rewriting, we append a token or identifier to the URL of the next Servlet or the next resource. We can send parameter name/value pairs using the following format:

```
url?name1=value1&name2=value2&??
```

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

3) URL Rewriting

- A name and a value is separated using an equal = sign, a parameter name/value pair is separated from another parameter using the ampersand(&). When the user clicks the hyperlink, the parameter name/value pairs will be passed to the server. From a Servlet, we can use getParameter() method to obtain a parameter value.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

3) URL Rewriting

Advantage of URL Rewriting

- It will always work whether cookie is disabled or not (browser independent).
- Extra form submission is not required on each pages.

Disadvantage of URL Rewriting

- It will work only with links.
- It can send Only textual information.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

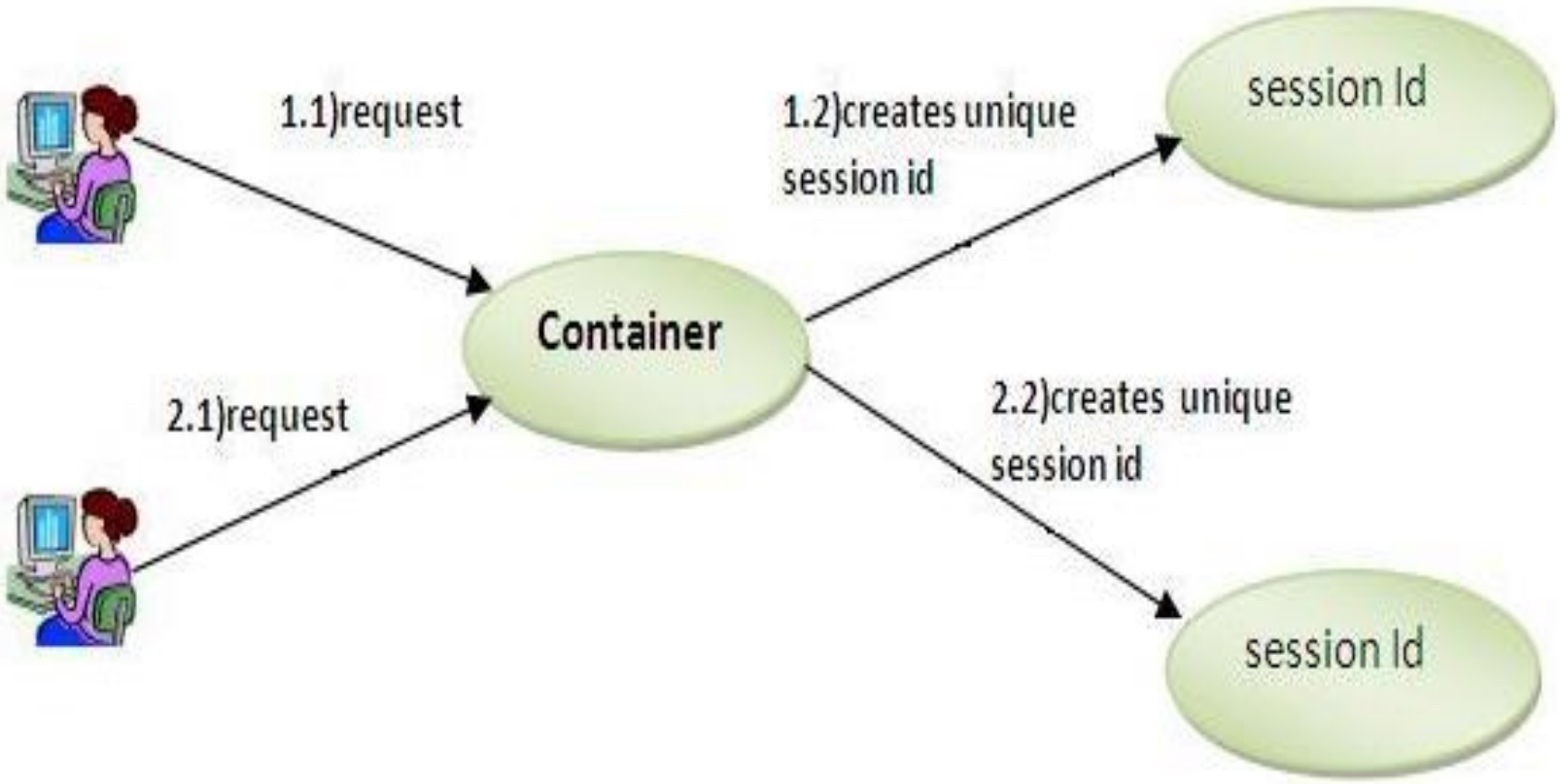
4) HttpSession interface

- In such case, container creates a session id for each user. The container uses this id to identify the particular user.
- An object of HttpSession can be used to perform two tasks:
    1. bind objects
    2. view and manipulate information about a session, such as the session identifier, creation time, and last accessed time.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

4) HttpSession interface



Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

4) HttpSession interface

How to get the HttpSession object ?

- The HttpServletRequest interface provides two methods to get the object of HttpSession:

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

4) HttpSession interface

1. **public HttpSession getSession():**Returns the current session associated with this request, or if the request does not have a session, creates one.
2. **public HttpSession getSession(boolean create):**Returns the current HttpSession associated with this request or, if there is no current session and create is true, returns a new session.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Session Tracking in Servlets…

4) HttpSession interface

Commonly used methods of HttpSession interface

- **public String getId():**Returns a string containing the unique identifier value.
- **public long getCreationTime():**Returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.
- **public long getLastAccessedTime():**Returns the last time the client sent a request associated with this session, as the number of milliseconds since midnight January 1, 1970 GMT.
- **public void invalidate():**Invalidates this session then unbinds any objects bound to it.

Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.

Summary

- In today's lecture we have discussed about;
  - Installing and configuring the Apache Tomcat server
  - Understanding the role of the RequestDispatcher interface in servlet collaboration
  - Explaining how session tracking preserves user state in web applications using stateless HTTP.

References

- Oracle. (2023). Setting up Apache Tomcat server environment. In Java Platform, Standard Edition documentation.
- Oracle. (2023). RequestDispatcher interface (Jakarta Servlet API documentation). In Java Platform, Enterprise Edition documentation.
- Oracle. (2023). HttpServletResponse interface: sendRedirect method (Jakarta Servlet API documentation). In Java Platform, Enterprise Edition documentation.
- Oracle. (n.d.). Session Tracking in Servlets. In Java Servlet technology documentation. Oracle.