Servlets and JSP

Review-

- Overview of Web Application
- HTTP Methods and Responses

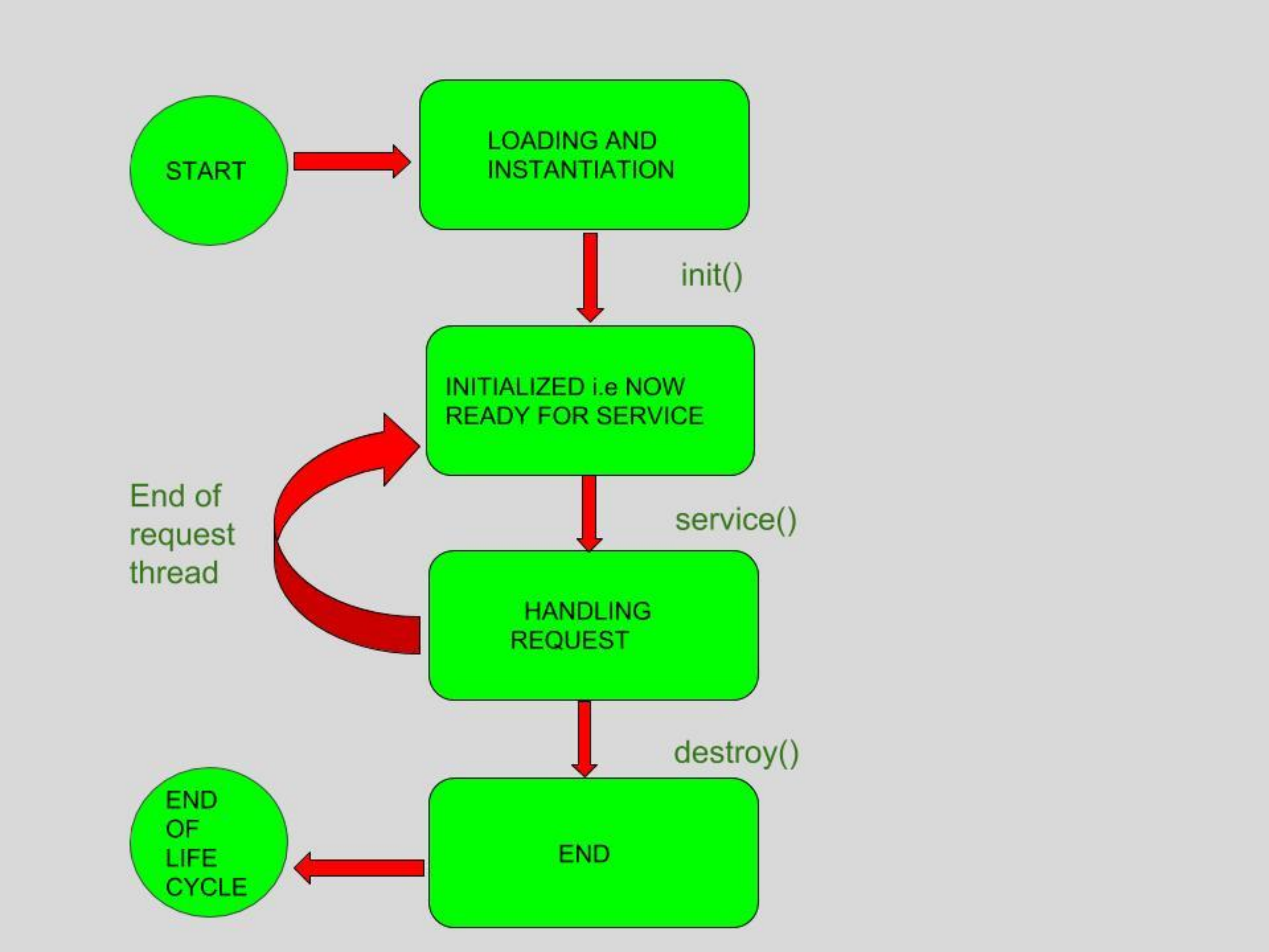
Outline

- Overview of Web Application
- HTTP Methods and Responses
- Lifecycle of Web Servlets
- Writing Servlet programs with Servlet APIs
- Reading and Processing Forms
- Handling GET/POST Requests
- Database connectivity through servlets
- Cookies and Sessions

Life Cycle Overview

- Initialization
    - Servlet container loads and initializes the servlet when the first request is received.
    - init() method is called during initialization.
- Handling Requests
    - Servlet handles multiple requests concurrently through multithreading.
    - service() method is invoked for each request, delegating to doGet() or doPost() based on the request type.
- Termina ion
    - Servlet container calls destroy() method when shutting down or reloading the web application.
    - Cleanup operations are performed in the destory() method.

Lifecycle Contd..



Writing Servlet Programs with Servlet APIs

- Import Statements:
    - Import necessary classes from the javax.servlet package.
- Extending HttpServlet:
    - Servlet classes extend HttpServlet to handle HTTP requests.
- Overriding Methods:
    - Override doGet() or doPost() methods to handle specific request types.

Handling Requests and Sending Responses

- Request and Response Objects:
    - Use HttpServletRequest to retrieve client data.
    - Use HttpServletResponse to send data back to the client.

Reading and Processing Forms

- HTML Forms:
  - HTML forms allow users to submit data to a server.
  - Elements like text fields, checkboxes, radio buttons, and buttons are used in forms.
- Servlet Form Handling:
  - Form Submission: Forms are submitted to the server using the POST or GET method.
  - Processing Form Data in Servlets: Retrieve form data using request.getParameter() method.
  - Example: Demonstrate a servlet that reads and processes form data.

Handling GET/POST Requests

- GET vs POST:
  - GET requests append data to the URL.
  - POST requests send data in the request body.
- HttpServlet Methods:
  - doGet() Method:
    - Invoked for HTTP GET requests.
    - Used to retrieve data from the server.
  - doPost() Method:
    - Invoked for HTTP POST requests.
    - Used to submit data to the server.

Database Connectivity Through Servlets

- Use JDBC to establish a connection to the database.
- Load JDBC driver, create a connection, and obtain a statement.
- Use PreparedStatement to execute SQL queries.
- Secure way to handle parameters in queries.

Cookies and Sessions

- Cookies
    - Small pieces of data stored on the client's browser.
    - Retaining user-specific information between requests
- Session
    - HttpSession interface is used for session management.
    - Allows the server to store and retrieve user-specific information

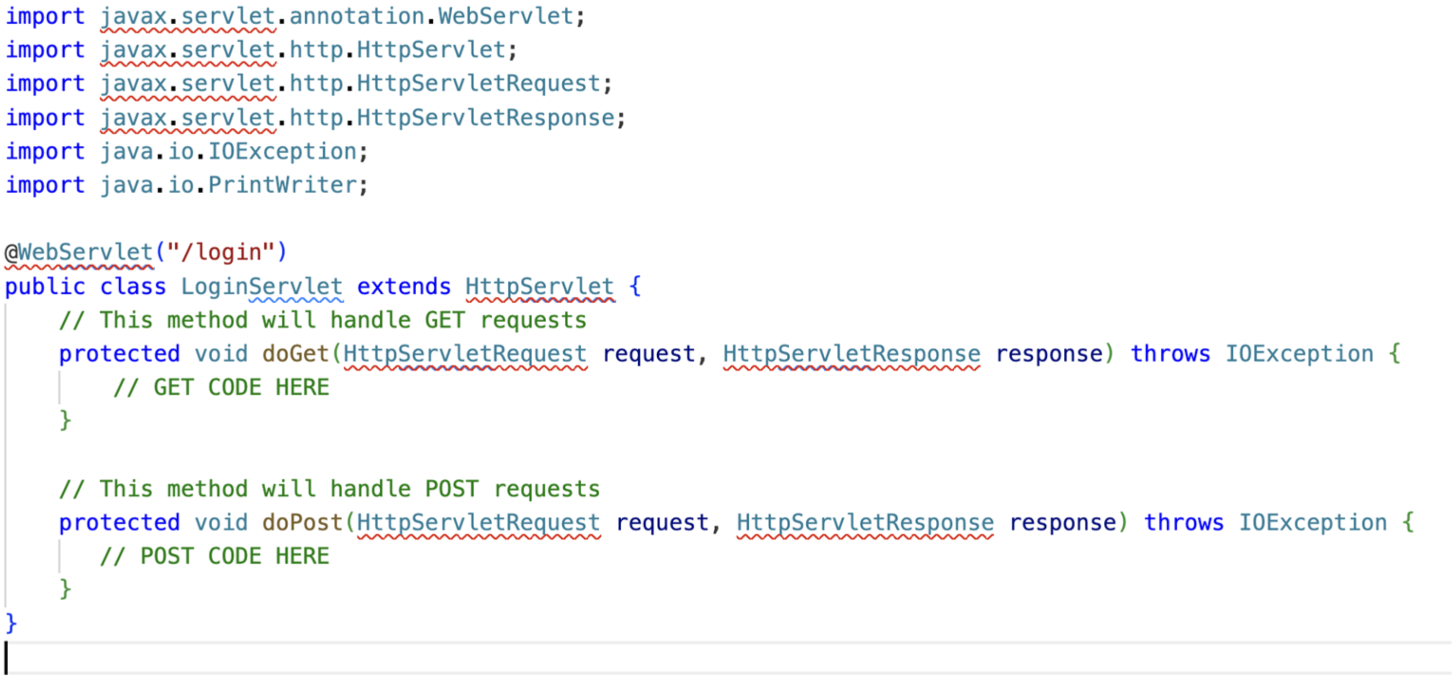
Handling Cookies and Sessions in Servlets

- Creating Cookies:
  - Use Cookie class to create cookies and add them to the response.
- Managing Sessions:
  - Use HttpSession to store and retrieve session data.

Creating Cookie

```
public class CookieServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
                Cookie cookie = new Cookie("username", "JohnDoe");          // Create a new cookie
                cookie.setMaxAge(24 * 60 * 60);  // Set the maximum age to 24 hours (24 * 60 * 60 seconds)
                response.addCookie(cookie);       // Add the cookie to the response
                response.setContentType("text/html");       // Set response content type
                response.getWriter().println("Cookie has been set with name 'username' and value
          'JohnDoe'.");    // Write a response to the client
    }
}
```

Servlet: Example

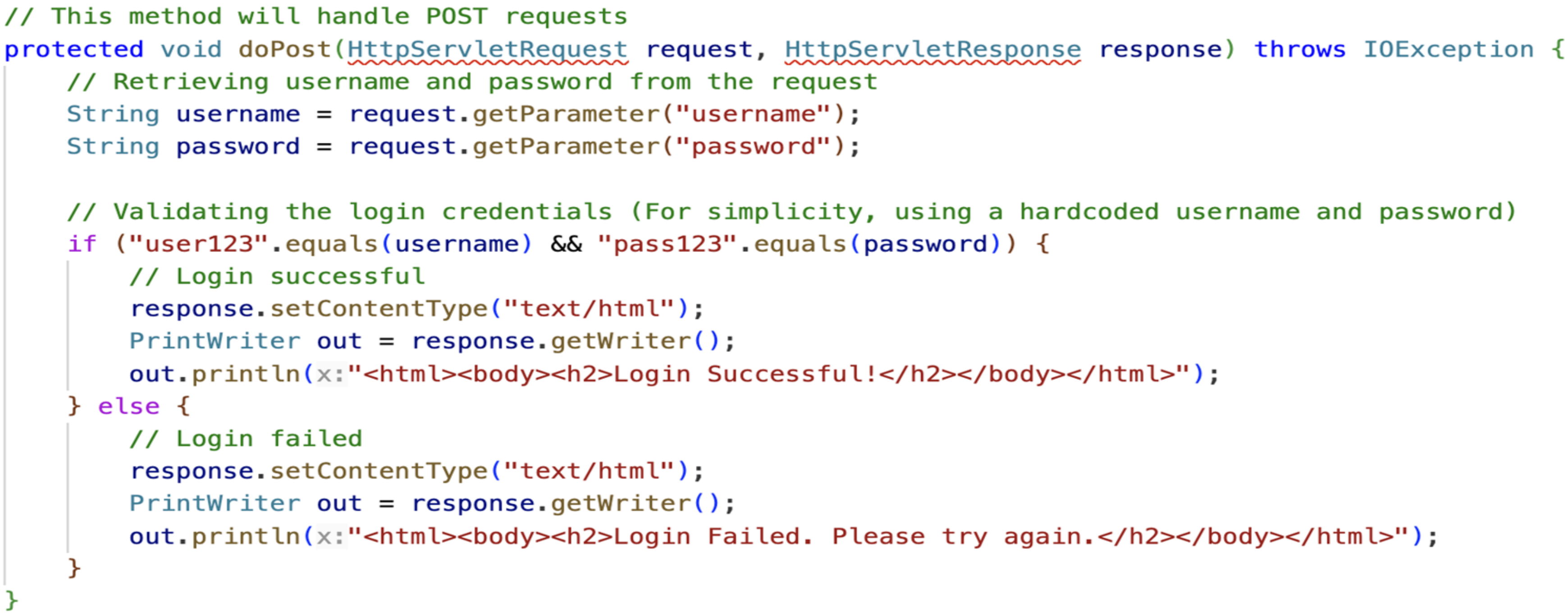


```
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/login")
public class LoginServlet extends HttpServlet {
    // This method will handle GET requests
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        // GET CODE HERE
    }

    // This method will handle POST requests
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {
       // POST CODE HERE
    }
}
```

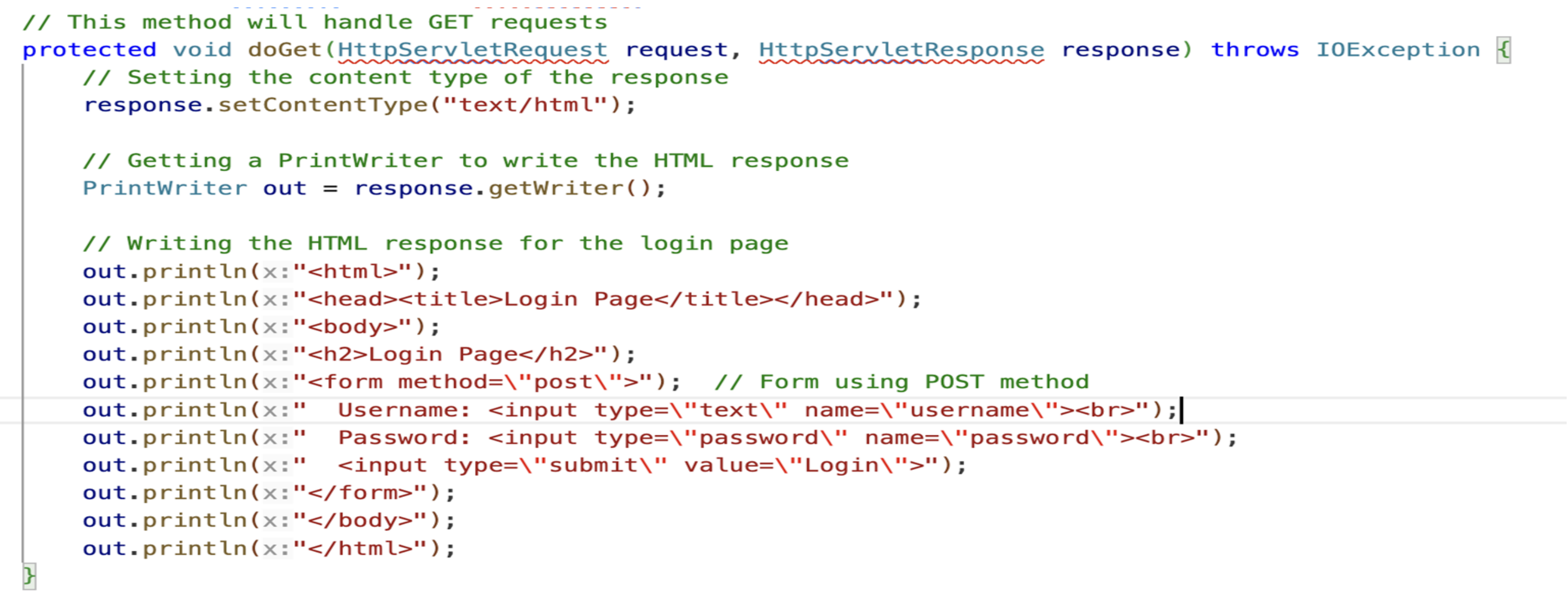
doPost()



```
// This method will handle POST requests
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {
    // Retrieving username and password from the request
    String username = request.getParameter("username");
    String password = request.getParameter("password");

    // Validating the login credentials (For simplicity, using a hardcoded username and password)
    if ("user123".equals(username) && "pass123".equals(password)) {
        // Login successful
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println(x:"<html><body><h2>Login Successful!</h2></body></html>");
    } else {
        // Login failed
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println(x:"<html><body><h2>Login Failed. Please try again.</h2></body></html>");
    }
}
```

doGet()



```
// This method will handle GET requests
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
    // Setting the content type of the response
    response.setContentType("text/html");

    // Getting a PrintWriter to write the HTML response
    PrintWriter out = response.getWriter();

    // Writing the HTML response for the login page
    out.println(x:"<html>");
    out.println(x:"<head><title>Login Page</title></head>");
    out.println(x:"<body>");
    out.println(x:"<h2>Login Page</h2>");
    out.println(x:"<form method=\"post\">");  // Form using POST method
    out.println(x:"  Username: <input type=\"text\" name=\"username\"><br>");
    out.println(x:"  Password: <input type=\"password\" name=\"password\"><br>");
    out.println(x:"  <input type=\"submit\" value=\"Login\">");
    out.println(x:"</form>");
    out.println(x:"</body>");
    out.println(x:"</html>");
}
```

Thank You!

Session and Cookies