Name : Mohil Mokaria

Class : 6CE

Division : E

Subject : Java Experiment 3

# index.jsx :

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Mohil Mokaria</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<h1>Experiment 3</h1>

<ul>

<li><a href="<%= request.getContextPath() %>/Problem1">Problem 1</a></li>

<li><a href="<%= request.getContextPath() %>/Problem2">Problem 2</a></li>

<li><a href="<%= request.getContextPath() %>/Problem3">Problem 3</a></li>

<li><a href="<%= request.getContextPath() %>/Problem4">Problem 4</a></li>

<li><a href="<%= request.getContextPath() %>/Problem5">Problem 5</a></li>

</ul>

</body>

</html>

# Q1 - Setting Integer Header with setIntHeader.

# To set an integer value for an HTTP response header using setIntHeader.

# Create an HTTP Servlet.

# Within the doGet or doPost method: a. Obtain the HttpServletResponse object. b. Use setIntHeader to set an integer value for a specific header (e.g., "Custom-Integer-Header"). c. Send the response.

## Definition :

* Project Contains Problem1.java
* MainServlet.java : Servlet that will handle get request and respond to them with setIntHeader.

## Code :

### Problem1.java

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.io.PrintWriter;

@WebServlet("/Problem1")

public class Problem1 extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setIntHeader("Custom-Integer-Header", 1);

response.setContentType("text/html");

PrintWriter pw = response.getWriter();

pw.println("<html><head><title>Int Header</title></head><body>");

pw.println("<h1>Getting Int Header</h1><p>Header Value : " + response.getHeader("Custom-Integer-Header") + "</p>");

pw.close();

}

}



# Q2 - Setting String Header with setHeader.

# To set a string value for an HTTP response header using setHeader.

# Create an HTTP Servlet.

# Within the doGet or doPost method: a. Obtain the HttpServletResponse object. b. Use setHeader to set a string value for a specific header (e.g., "Custom-String-Header"). c. Send the response.

## Definition :

* Project Contains Problem2.java
* MainServlet.java : Servlet that will handle get request and respond to them with setHeader.

## Code :

### Problem2.java

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.io.PrintWriter;

@WebServlet("/Problem2")

public class Problem2 extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setHeader("Custom-String-Header", "This is a java EE project");

response.setContentType("text/html");

PrintWriter pw = response.getWriter();

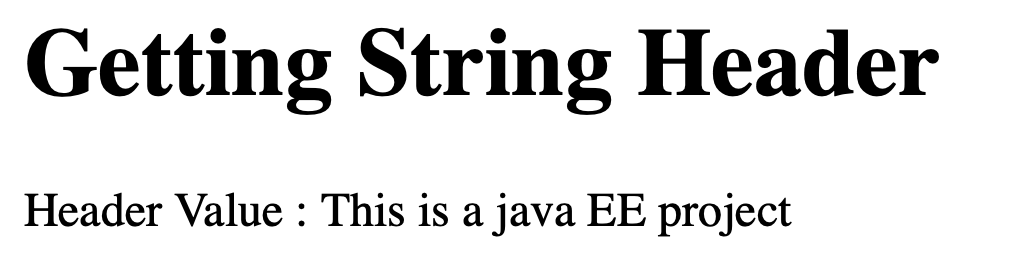
pw.println("<html><head><title>String Header</title></head><body>");

pw.println("<h1>Getting String Header</h1><p>Header Value : " + response.getHeader("Custom-String-Header") + "</p>");

pw.close();

}

}



# Q3 - Testing Content-Type Header.

# To set the Content-Type header using both setIntHeader and setHeader methods and observe the differences.

# Create an HTTP Servlet.

# Within the doGet or doPost method: a. Obtain the HttpServletResponse object. b. Use setIntHeader to attempt setting the Content-Type header with an integer value. c. Use setHeader to set the Content-Type header with a string value like "text/plain" or "application/json". d. Compare the results in the response headers using browser developer tools or an HTTP client (like cURL or Postman).

## Definition :

* Project Contains Problem3.java
* MainServlet.java : Servlet that will handle get request and respond to them with setIntHeader and setHeader both, to set content type.

## Code :

### Problem3.java

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.io.PrintWriter;

@WebServlet("/Problem3")

public class Problem3 extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

PrintWriter pw = response.getWriter();

response.setIntHeader("Content-Type", 123);

response.setHeader("Content-Type", "text/html");

pw.println("<html><head><title>Content by Header</title></head><body>");

pw.println("<h1>Getting String Header</h1><p>Into Servlet where Content-Type = text/html</p>");

pw.close();

}

}



# Q4 - **Implement Error Handling and Exception Scenarios.**

# Handle potential exceptions or errors while using setIntHeader and setHeader.

# Create an HTTP Servlet.

# Introduce scenarios such as passing invalid input or incorrect header names/values to setIntHeader and setHeader.

# Implement try-catch blocks to handle potential exceptions that may arise due to incorrect usage.

# Log or display appropriate error messages if exceptions occur.

# Definition :

* Project Contains Problem4.java
* MainServlet.java : Servlet that will handle all types of exceptions for setIntHeader.

## Code :

### Problem4.java

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

@WebServlet("/Problem4")

public class Problem4 extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

try {

// Scenario 1: Using setIntHeader with a non-integer value

response.setIntHeader("Content-Type", 123); // This should not throw an exception

// Scenario 2: Using setIntHeader with a non-numeric value (should throw an exception)

response.setIntHeader("Invalid-Header", Integer.parseInt("NotAnInteger"));

// Scenario 3: Using setHeader with invalid header name or value (should not throw an exception)

response.setHeader("Invalid-Header-Name", "text/plain");

response.setHeader("Content-Type", "Invalid-Header-Value");

// Writing some content to the response

response.getWriter().println("No exceptions occurred!");

} catch (NumberFormatException e) {

// Handle NumberFormatException (e.g., when trying to parse a non-numeric value)

response.getWriter().println("NumberFormatException occurred: " + e.getMessage());

} catch (IllegalArgumentException e) {

// Handle IllegalArgumentException (e.g., when using an invalid header name or value)

response.getWriter().println("IllegalArgumentException occurred: " + e.getMessage());

} catch (Exception e) {

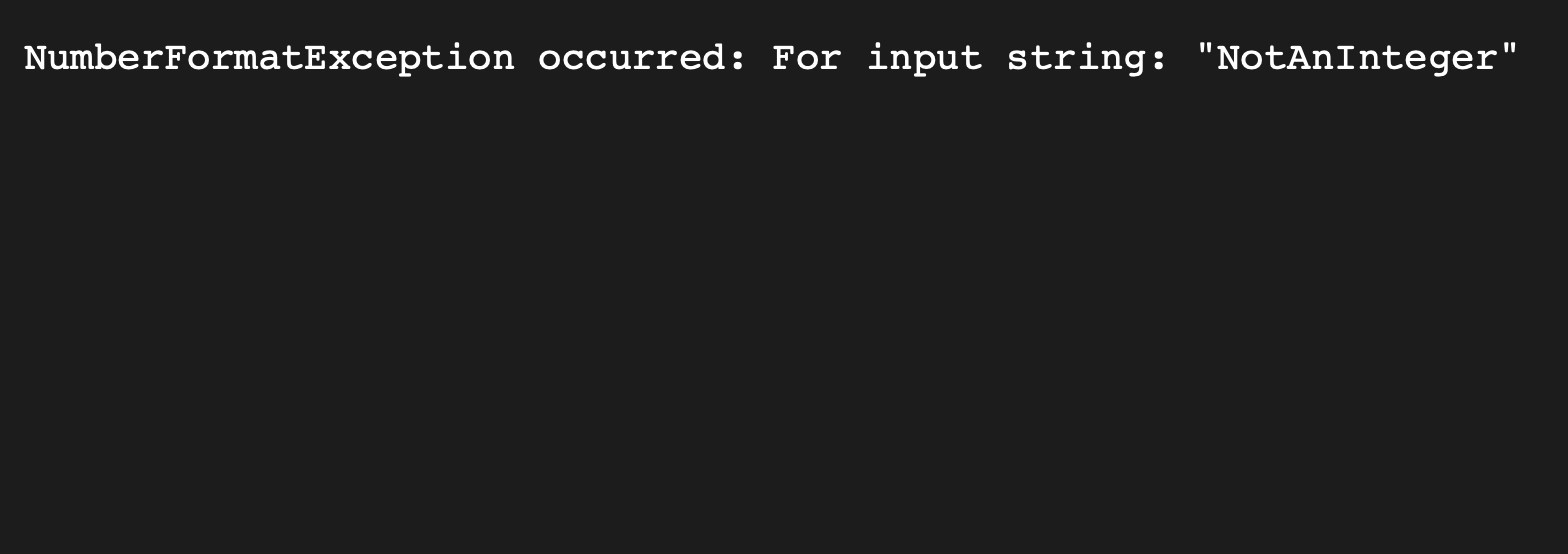
// Handle other exceptions

response.getWriter().println("An unexpected exception occurred: " + e.getMessage());

}

}

}



# Q5 - **Impact on Response Size and Performance**

# Measure and compare the impact on response size and performance when setting headers using setIntHeader and setHeader.

# Create an HTTP Servlet that generates a sizable response (JSP page, HTML content).

# Measure the response size and performance (time taken to generate and send the response) with headers set using setIntHeader.

# Repeat the measurement with headers set using setHeader.

# Analyze the differences in response size and performance between the two approaches.

# Definition :

* Project Contains Problem5.java
* MainServlet.java : Servlet that will send a response of huge size, to measure impact and at last setting header and intheader.

## Code :

### Problem5.java

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

@WebServlet("/Problem5")

public class Problem5 extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// Generate sizable content (e.g., HTML content)

StringBuilder content = new StringBuilder("<html><body>");

for (int i = 0; i < 10; i++) {

content.append("<p>This is a line of content.</p>");

}

content.append("</body></html>");

// Measure performance with setIntHeader

long startTimeSetIntHeader = System.currentTimeMillis();

try {

response.setIntHeader("Content-Type", 123);

response.getWriter().write(content.toString());

} catch (IOException e) {

e.printStackTrace();

}

long endTimeSetIntHeader = System.currentTimeMillis();

long responseSizeSetIntHeader = response.getBufferSize();

// Reset response to clear previous headers and content

response.reset();

// Measure performance with setHeader

long startTimeSetHeader = System.currentTimeMillis();

try {

response.setHeader("Content-Type", "text/html");

response.getWriter().write(content.toString());

} catch (IOException e) {

e.printStackTrace();

}

long endTimeSetHeader = System.currentTimeMillis();

long responseSizeSetHeader = response.getBufferSize();

// Compare and analyze the results

long timeTakenSetIntHeader = endTimeSetIntHeader - startTimeSetIntHeader;

long timeTakenSetHeader = endTimeSetHeader - startTimeSetHeader;

System.out.println("Response Size (setIntHeader): " + responseSizeSetIntHeader + " bytes");

System.out.println("Time Taken (setIntHeader): " + timeTakenSetIntHeader + " milliseconds");

System.out.println("Response Size (setHeader): " + responseSizeSetHeader + " bytes");

System.out.println("Time Taken (setHeader): " + timeTakenSetHeader + " milliseconds");

}

}

