

Slot 2 – Slot 3

Java Serve Page

1. Giới thiệu

Java Serve Page: file có đuôi mở rộng, các file jsp đặt trong thư mục Web pages

JSP: Java Server Page

Server (back end): jsp , servlet

Các thẻ:

Ở đầu tệp (pageEncoding="utf-8" – nếu tiếng Việt)

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

Mã JSP (Scriptlets):

```
<% code java %>
```

Ví dụ viết vào văn bản:

```
<% out.println("Welcome to JSP world"); %>
```

```
<% out.write("How are you?\n"); %>
```

Khai báo (Declarations):

```
<%! int x=1; %>
```

Biểu thức (Expressions):

```
<%= expression %>
```

Chú thích (comment): đâu là chú thích??

1. <!-- đây là chú thích -->
2. <p>chú ý hay nhầm lẫn : <!-- <%= name %> --></p>
3. <%--hidden comment --%>
4. <%// hidden comment %>

No.	Đối tượng & Mô tả
1	request Đây là đối tượng HttpServletRequest được liên kết với request.
2	response Đây là đối tượng HttpServletResponse được liên kết với response tới client.
3	out Đây là đối tượng PrintWriter được sử dụng để gửi dữ liệu tới client.
4	session Đây là đối tượng HttpSession được liên kết với request.
5	application Đây là đối tượng ServletContext được liên kết với application context.
6	config Đây là đối tượng ServletConfig được liên kết với page.
7	pageContext Điều này đóng gói việc sử dụng các tính năng cụ thể của server như JspWriters với hiệu suất cao hơn.
8	page Đơn giản là một từ đồng nghĩa với this , được sử dụng để gọi các phương thức được định nghĩa bởi lớp serv được biên dịch.
9	exception Đối tượng exception cho phép các dữ liệu ngoại lệ (Exception) được truy cập bằng JSP được chỉ định.

2. Thẻ điều hướng:

Có 3 loại thẻ directive trong JSP:

Thẻ điều hướng Directive	Mô tả
<% @ page ... %>	Định nghĩa một thuộc tính page-dependent (phụ thuộc trang), như ngôn ngữ scripting, trang lỗi và các yêu cầu bộ đệm
<% @ include ... %>	Include một file trong suốt giai đoạn biên dịch
<% @ taglib ... %>	Khai báo một thư viện thẻ, chứa các action tùy biến, được sử dụng trong trang đó

2.1 page directive

Page directive được sử dụng để cung cấp các chỉ lệnh tới container mà liên quan tới JSP page hiện tại. Ta có thể mã hóa page directive bất cứ đâu trong JSP page.

Cú pháp:

```
<% @ page attribute="value" %>
```

Ví dụ:

```
<% @ page buffer="none" %>
```

```
<% @ page buffer="8kb" %>
<% @ page contentType="text/xml" %>
<% @ page contentType="text/html" %>
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<% @ page import="java.sql.*" %>
<% @page import="java.util.Date, java.text.SimpleDateFormat"%>
```

2.2. include Directive trong JSP được sử dụng để bao một file trong suốt giai đoạn biên dịch. Ta có thể mã hóa include directive bất cứ đâu trong JSP page.

```
<% @ include file="relative url" >
```

Ví dụ:

```
<% @include file="header.jsp" %>
<% @include file="logo.html" %>
<% @include file="content.jsp" %>
```

Java Server Pages API cho phép định nghĩa các Custom Tags trong JSP, mà giống như các thẻ HTML hoặc XML và một thư viện thẻ (tag library) là một tập hợp các thẻ được định nghĩa bởi người sử dụng mà triển khai các hành vi tùy biến.

2.3. taglib directive khai báo rằng: JSP page sử dụng một tập hợp các Custom Tags, thông báo vị trí của thư viện, và cung cấp một phương thức để nhận diện các Custom Tags đó trong JSP page.

```
<% @ taglib uri="uri" prefix="prefixOfTag" >
```

Exercise:

Exercise 1:

Truy cập từ file jsp có form sau: Khi nhập vào bán kính thì tính ra diện tích của hình tròn

Bài tập 1

enter radius:

Có 3 cách:

Cách 1: access to page via url: /ex1.jsp (dựng form + xử lý backend đều dùng jsp)

Cách 2: access to page via url: /cal (làm hoàn toàn bằng servlet)

Cách 3: access to page via url: /cal (dựng form dùng jsp, xử lý backend dùng servlet)

Note: Muốn chạy dự án từ file jsp thì ta cấu hình như sau: mở file web.xml sau đó thêm cài đặt:

```
<welcome-file-list>
    <welcome-file>
        circle1.jsp
    </welcome-file>
</welcome-file-list>
```

Cách 1: access to page via url: /circle1.jsp (dựng form + xử lý backend đều dùng jsp)

Tạo file Circle.jsp trong thư mục WEB Page

```
10 <html>
11 <head>
12     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
13     <title>Circle Area Calculation</title>
14 </head>
15 <body>
16     <h2>Calculate Circle Area</h2>
17
18     <form action="Circle.jsp" method="post">
19         <table>
20             <tr>
21                 <td>Enter radius:</td>
22                 <td><input type="text" name="radius"
23                     value="<%= request.getParameter("radius") != null ? request.getParameter("radius") : "" %>"/>
24                 </td>
25             </tr>
26             <tr>
27                 <td><input type="submit" value="SUBMIT"></td>
28             </tr>
29         </table>
30     </form>
```

Xử lý backend:

```
32 <%
33 String radiusStr = request.getParameter("radius");
34 if (radiusStr != null) {
35     try {
36         double radius = Double.parseDouble(radiusStr);
37         if (radius < 0) {
38             out.println("<p style='color:red;'>"
39                 + "Error: Radius cannot be negative. Please enter a positive value.</p>");
40         } else {
41             double area = Math.PI * radius * radius;
42             // Định dạng diện tích với 2 chữ số sau dấu phẩy
43             DecimalFormat df = new DecimalFormat("#.##");
44             String formattedArea = df.format(area);
45             out.println("<p>Area of the circle with radius " + radius + " is: " + formattedArea + "</p>");
46         }
47     } catch (NumberFormatException e) {
48         out.println("<p style='color:red;'>Error: Invalid input. Please enter a valid number.</p>");
49     }
50 }
51 %>
```

Đề định dạng giá trị diện tích có 2 chữ số sau dấu phẩy thập phân, sử dụng thêm thư viện

<% @page import="java.text.DecimalFormat"%>

```
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <%@page import="java.text.DecimalFormat"%>
```

Cách 2: access to page via url: /cal (dùng form dùng jsp, xử lý backend dùng servlet)

Các bước chính:

1. Tạo JSP để hiển thị form nhập bán kính và kết quả (ví dụ: circle2.jsp).
2. Tạo Servlet để xử lý logic (ví dụ: CircleServlet.java).

Bước 1:

```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6 <title>Circle Area Calculation 2</title>
7 </head>
8 <body>
9 <h2>Calculate Circle Area 2</h2>
10
11 <form action="cal" method="post">
12 <table>
13 <tr>
14 <td>Enter radius:</td>
15 <td>
16 <input type="text" name="radius"
17 value="<%= request.getAttribute("radius") != null ? request.getAttribute("radius") : "" %>"/>
18 </td>
19 </tr>
20 <tr>
21 <td colspan="2"><input type="submit" value="Submit"></td>
22 </tr>
23 </table>
24 </form>
```

Bước 2: Tạo Servlet để xử lý logic

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String radiusStr = request.getParameter(string: "radius");
    String message = null;
    double area = 0;

    try {
        double radius = Double.parseDouble(s: radiusStr);

        if (radius < 0) {
            message = "Error: Radius cannot be negative. Please enter a positive value.";
        } else {
            area = Math.PI * radius * radius;
            DecimalFormat df = new DecimalFormat(pattern: "#.##");
            request.setAttribute(string: "formattedArea", o: df.format(number: area));
        }
    } catch (NumberFormatException e) {
        message = "Error: Invalid input. Please enter a valid number.";
    }

    // Đặt các thông báo hoặc kết quả vào request attribute
    request.setAttribute(string: "message", o: message);
    request.setAttribute(string: "radius", o: radiusStr);

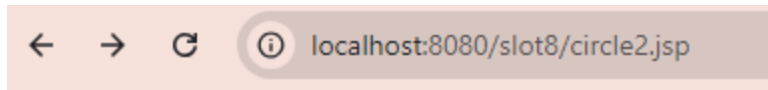
    // Chuyển hướng đến trang JSP để hiển thị kết quả
    RequestDispatcher dispatcher = request.getRequestDispatcher(string: "circle2.jsp");
    dispatcher.forward(s: request, s: response);
}
```

Hiển thị lại kết quả ở trên trang jsp, tại file circle2.jsp:

```
<%
String message = (String) request.getAttribute("message");
String formattedArea = (String) request.getAttribute("formattedArea");

if (message != null) {
    out.println("<p style='color:red;'>" + message + "</p>");
} else if (formattedArea != null) {
    out.println("<p>Area of the circle: " + formattedArea + "</p>");
}
%>
```

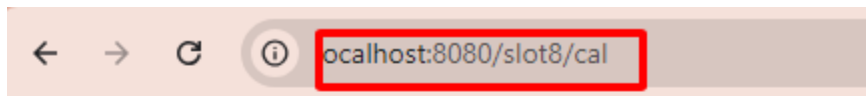
Khi chạy: do để nhiều file jsp trong cùng 1 project thì ta sẽ chạy file circle2.jsp từ trình duyệt



Calculate Circle Area 2

Enter radius:

Sau đó sẽ gửi yêu cầu đến servlet tại cal



Calculate Circle Area 2

Enter radius:

Error: Radius cannot be negative. Please enter a positive value.

Cách 3: Access to page via url: /cal3 (làm hoàn toàn bằng servlet)

Tạo file servlet Circle3Servlet.java

Sửa hàm doGet để điều hướng đến hàm doPost

```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    //processRequest(request, response);
    // Redirect to POST method for form handling
    doPost(request, response);
}
```

Xóa phương thức processRequest và thay thế bằng logic cụ thể trong phương thức doGet và doPost.

Hàm doPost:

```

23 // Xử lý khi người dùng gửi form qua phương thức POST
24 @Override
25 protected void doPost(HttpServletRequest request, HttpServletResponse response)
26     throws ServletException, IOException {
27     response.setContentType("text/html;charset=UTF-8");
28     String radiusStr = request.getParameter("radius");
29     String message = null;
30     double area = 0;
31     try (PrintWriter out = response.getWriter()) {
32         out.println("<!DOCTYPE html>");
33         out.println("<html>");
34         out.println("<head>");
35         out.println("<title>Circle Area Calculation</title>");
36         out.println("</head>");
37         out.println("<body>");
38         out.println("<h2>Circle Area Calculation (Servlet)</h2>");
39
40         // Kiểm tra xem radiusStr có null hoặc rỗng hay không
41         if (radiusStr == null || radiusStr.trim().isEmpty()) {
42             message = "Error: Please enter a radius value.";
43             out.println("<p style='color:red;'>" + message + "</p>");
44         } else {
45             try {
46                 double radius = Double.parseDouble(radiusStr);
47                 if (radius < 0) {
48                     message = "Error: Radius cannot be negative. Please enter a positive value.";
49                     out.println("<p style='color:red;'>" + message + "</p>");
50                 } else {
51                     area = Math.PI * radius * radius;
52                     DecimalFormat df = new DecimalFormat("###");
53                     out.println("<p>Area of the circle: " + df.format(area) + "</p>");
54                 }
55             } catch (NumberFormatException e) {
56                 message = "Error: Invalid input. Please enter a valid number.";
57                 out.println("<p style='color:red;'>" + message + "</p>");
58             }
59         }
60
61         // Form để nhập lại bán kính
62         out.println("<form action='cal3' method='post'>");
63         out.println("<table>");
64         out.println("<tr><td>Enter radius:</td>");
65         out.println("<td><input type='text' name='radius' value='" + (radiusStr != null ? radiusStr : "") + "'/></td></tr>");
66         out.println("<tr><td><input type='submit' value='Submit'></td></tr>");
67         out.println("</table>");
68         out.println("</form>");
69         out.println("</body>");
70         out.println("</html>");
71     }
72
73 // Mô tả servlet
74 @Override
75 public String getServletInfo() {
76     return "Circle Area Calculation Servlet";
77 }

```

Khi chạy:

- Chạy từ url <http://localhost:8080/slot8/cal3>

Exercise 2:

You are asked to build a webpage to show your own roll number. The user can access your page via url **/getfirst**

Your Tasks include:

*) When the user access /getfirst for the first time, the system displays the form as below figure. The roll number in this form must be **your own roll number**

Question 1

(3 points)

You are asked to build a webpage to show your own roll number. The user can access your page via url:

/getfirst

Your Tasks include:

- (1 point) When the user access /getfirst for the first time, the system displays the form as below figure. The roll number in this form must be **your own roll number**.

HE150167

Get first letter(s)

Figure 1 – The page on first visit

Tạo form theo yêu cầu:

```

<!DOCTYPE html>
<html>
<head>
  <title>Get First Characters</title>
</head>
<body>
  <h1>HE150167</h1>

  <form action="getfirst" method="post">
    <table>
      <tr>
        <td>
          <label for="number">Get first</label>
          <!-- Hiển thị lại giá trị người dùng đã nhập bằng EL -->
          <input type="text" id="number" name="number"
            value="{param.number != null ? param.number : ''}" />
          <label for="number">letter(s)</label>
        </td>
      </tr>
      <tr>
        <td>
          <input type="submit" value="Get"/>
        </td>
      </tr>
    </table>
  </form>

```

- **(2 point)** When the user clicks [Get] button, the system check the input string in the textboxes. If this string representing a positive integer that called *[number]*, show the substring of first *[number]* characters of the roll number (see Figure 2). Otherwise, show the error message like Figure 3.

Note that, if *[number]* is larger than the length of the roll number, show the string of roll number only.

HE150167

Get first letter(s)

Your first 3 letter(s) is/are: HE1

Figure 2 – The page when user enter “3” and then click on button [Get]

HE150167

Get first letter(s)

Input string does not represent a positive integer

Tạo servlet có url pattern là **/getfirst**

Trong servlet:

Hàm doGet

```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.sendRedirect(string: "exercise2.jsp");
}
```

Hàm doPost:

```

44 private static final String ROLL_NUMBER = "HE150167";
45 // Handles the HTTP POST method
46 @Override
47 @protected void doPost(HttpServletRequest request, HttpServletResponse response)
48     throws ServletException, IOException {
49     // Set character encoding to UTF-8
50     request.setCharacterEncoding("UTF-8");
51     response.setContentType("text/html;charset=UTF-8");
52     // Get the user input
53     String input_ = request.getParameter("number");
54     String message = null;
55     // Validate input
56     if (input_ == null || input_.trim().isEmpty()) {
57         // User hasn't provided input, set error message
58         message = "Error: Please enter a valid number.";
59         request.setAttribute("error", message);
60     } else {
61         try {
62             // Parse input to an integer
63             int input = Integer.parseInt(input_.trim());
64             // Check for negative numbers
65             if (input < 0) {
66                 message = "Error: Number must be positive.";
67                 request.setAttribute("error", message);
68             }
69             // Check if input exceeds roll number length
70             else if (input > ROLL_NUMBER.length()) {
71                 message = "Error: Number exceeds the length of the roll number.";
72                 request.setAttribute("error", message);
73             } else {
74                 // Extract the first n characters from the roll number
75                 String firstNLetters = ROLL_NUMBER.substring(beginIndex: 0, endIndex: input);
76                 message = input + " first letter(s): " + firstNLetters;
77                 request.setAttribute("result", message);
78             }
79         } catch (NumberFormatException e) {
80             // Catch exception for invalid number format
81             message = "Error: Invalid input. Please enter a valid number.";
82             request.setAttribute("error", message);
83         }
84     }
85     // Forward request and response to JSP page
86     request.getRequestDispatcher("exercise2.jsp").forward(request, response);
87 }

```

Cập nhật form trong file **exercise2.jsp** để hiển thị kết quả:

```

6 <body>
7 <h1>HE150167</h1>
8 <form action="getfirst" method="post">
9 <table>
10 <tr>
11 <td>
12 <label for="number">Get first</label>
13 <!-- Hiển thị lại giá trị người dùng đã nhập bằng EL -->
14 <input type="text" id="number" name="number"
15 <value="{param.number != null ? param.number : ''}"/>
16 <label for="number">letter(s)</label>
17 </td>
18 </tr>
19 <tr>
20 <td>
21 <input type="submit" value="Get"/>
22 </td>
23 </tr>
24 </table>
25 </form>
26 <!-- Hiển thị thông báo lỗi hoặc kết quả -->
27 <div>
28 <p style="color:red;">
29 <{error != null ? error : ''}>
30 </p>
31 <p style="color:green;">
32 <{result != null ? result : ''}>
33 </p>
34 </div>
35 </body>
36 </html>

```

Exercise 3:

Tạo form Login với username và password, username: yourName, password là 123@.

Nếu đăng nhập sai thì thông báo nhập sai (chữ màu đỏ) phía trên form nhập.

Nếu nhập đúng chuyển sang trang welcome.jsp

Exercise 4:

Trang home gồm các file: Header.jsp, tableft.jsp và footer.jsp Trang danh sách list giống hệt trang home chỉ khác phần content:

Làm lại các bài xử lý Servlet bằng JSP:

Exercise 01:

Example Tính chu vi của hình chữ nhật

Nhập vào chiều dài:

Nhập vào chiều rộng:

Nếu nhập vào chiều dài, chiều rộng là số âm hoặc chữ cái thì sẽ thông báo lỗi nhập dữ liệu và yêu cầu nhập lại.

Exercise 02:

Example Tính chu vi và diện tích của hình chữ nhật

Nhập vào chiều dài:

Nhập vào chiều rộng:

Tính Chu vi: ☐ Tính Diện tích: ☐

Nếu nhập vào chiều dài, chiều rộng là số âm hoặc chữ cái thì sẽ thông báo lỗi nhập dữ liệu và yêu cầu nhập lại.

Sau khi nhập dữ liệu đúng, nếu người dùng bấm vào Chu vi thì hiển thị chu vi của Hình chữ nhật, nếu người dùng bấm Diện tích thì hiển thị diện tích của hình chữ nhật. Nếu tích chọn cả hai thì hiển thị cả 2 thông số về chu vi và diện tích.

Exercise 03:

Login form

User name:

Password:

Exercise 04:

Thông tin sinh viên

Họ và Tên đệm:

Tên:

Giới tính: ☐ Nam ☐ Nữ

Chọn môn học:

Nhập Điểm (0 - 10):

Tạo trang JSP để xử lý form như mẫu với yêu cầu:

Nếu để trống các trường thì sẽ thông báo lỗi không được để trống,

Nếu nhập điểm <0 hoặc >10 thì thông báo điểm phải nằm trong khoảng $[0,10]$.