



FPT POLYTECHNIC



Conceive Design Implement Operate

GIỚI THIỆU VỀ JSP, FORM DATA VỚI JSP

LẬP TRÌNH JAVA #3 (P2.1)

www.poly.edu.vn

- ❑ Giới thiệu trang JSP
- ❑ JSP Directives
 - ❖ `<%@page%>`
 - ❖ `<%@include%>`
 - ❖ `<%@taglib%>`
- ❑ JSP Standard Actions:
 - ❖ `<jsp:include>`
 - ❖ `<jsp:forward>`
 - ❖ `<jsp:param>`
- ❑ Biểu thức EL cơ bản



- ❑ JSP (Java Server Page), EL (Expression Language) và JSTL (Java Standard Tag Library) là bộ ba giúp lập trình render giao diện từ phía server.
- ❑ Trước JSP 2.0 trang JSP là sự trộn lẫn giữa mã java và HTML, làm cho JSP rất khó đọc, khó quản lý (không chuẩn markup).
- ❑ Từ 2.x trở đi, EL và JSTL được áp dụng làm giảm độ phức tạp của lập trình JSP đồng thời trang JSP rõ ràng hơn, dễ quản lý hơn, đúng chuẩn markup hơn.
- ❑ *Trong bài học này chúng ta chỉ nghiên cứu JSP và EL cơ bản, JSTL sẽ được nghiên cứu ở những bài tiếp theo.*

```
<%@page pageEncoding="utf-8" %>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <meta charset="utf-8">
```

```
    <title>FPT Polytechnic</title>
```

```
</head>
```

```
<body>
```

```
    <h1>${student}</h1>
```

```
    <jsp:include page="student-details.jsp"/>
```

```
</body>
```

```
</html>
```

Directives

EL (Expression Language)

Standard Actions

❑ `<%@page pageEncoding="utf-8"%>`

❖ Khai báo trang JSP hỗ trợ UTF-8 (Tiếng Việt)

❑ `<%@include file="sub-page.jsp"%>`

❖ Chỉ thị này giúp module hóa các thành phần giao diện, giúp chèn toàn bộ mã JSP của trang sub-page.jsp tại vị trí đặt chỉ thị @include. Công việc chèn này xảy ra tại thời điểm dịch. Vì vậy xem mã trong sub-page.jsp là một phần của trang JSP hiện tại.

❑ `<%@taglib uri="" prefix=""%>`

❖ Nhúng bộ thư viện thẻ vào trang JSP (trình bày kỹ ở phần JSTL)

main.jsp

```
<%@ page pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Insert title here</title>
</head>
<body>
    <h1>Directive Demo</h1>
    <%@include file="sub.jsp"%>
</body>
</html>
```

sub.jsp

```
<form action="/upload" method="post">
    <input name="fullname" placeholder="Fullname?">
    <br>
    <button>Submit</button>
</form>
```



❑ Chỉ thị @page ngoài thuộc tính pageEncoding còn có một số thuộc tính khác ít được sử dụng như hình bên, bạn có thể tham khảo thêm.

Ⓐ autoFlush="true"

Ⓐ buffer="8kb"

Ⓐ contentType="text/html; charset=ISO-8859-1"

Ⓐ deferredSyntaxAllowedAsLiteral="false"

Ⓐ errorPage

Ⓐ extends

Ⓐ import

Ⓐ info

Ⓐ isELIgnored="false"

Ⓐ isErrorPage="false"

Ⓐ isThreadSafe="true"

Ⓐ language="java"

Ⓐ session="true"

❑ **<jsp:include page="sub-page.jsp"/>**

- ❖ Render giao diện của sub-page.jsp tại vị trí đặt thẻ (tương tự req.getRequestDispatcher().include() trong servlet)
- ❖ Chú ý sự khác biệt so với chỉ thị <%@include%>
 - <%@include%> chèn mã trong khi đó <jsp:include> render kết quả
 - <%@include%> xảy ra lúc dịch trong khi đó <jsp:include> xảy ra lúc chạy

❑ **<jsp:forward page="sub-page.jsp"/>**

- ❖ Chuyển tiếp sang sub-page.jsp (tương tự req.getRequestDispatcher().forward() trong servlet)

❑ Chú ý:

- ❖ <jsp:include/> sau khi render giao diện sẽ quay trở lại trang JSP hiện tại để thực hiện tiếp mã phía dưới trong khi đó <jsp:forward> không quay trở lại trang hiện tại
- ❖ Trước khi forward không nên render giao diện, nhiệm vụ render giao diện là của trang JSP cuối cùng trong chuỗi forward.

PHÂN BIỆT <JSP:INCLUDE> VÀ @INCLUDE

1...
2...
<%@include file="sub.jsp"%>
3...

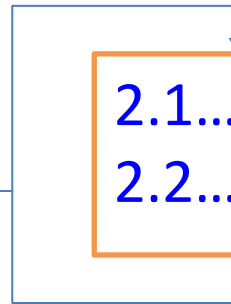
2.1...
2.2... sub.jsp



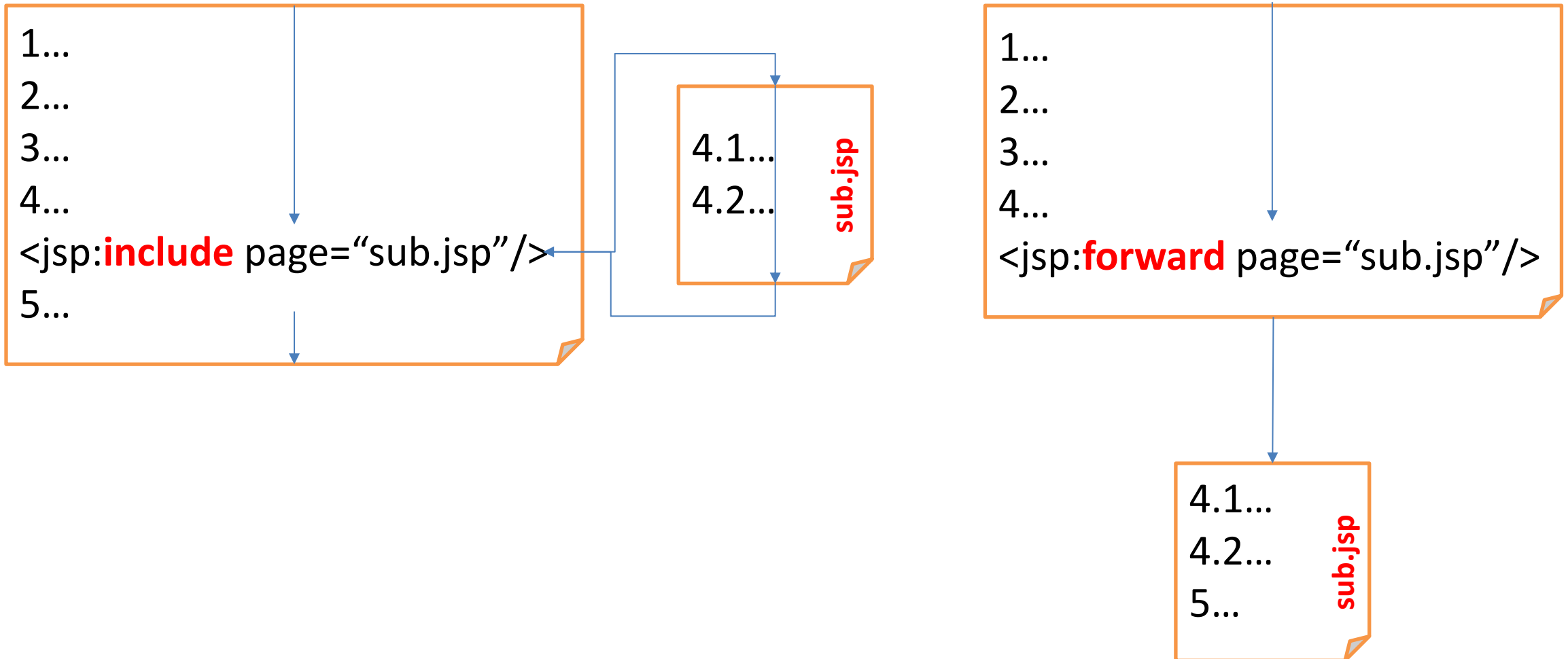
1...
2...
2.1...
2.2...
3...

1...
2...
<jsp:include page="sub.jsp"/>
3...

2.1...
2.2... sub.jsp



<JSP:INCLUDE> VÀ <JSP:FORWARD>



- ❑ Tạo tham số truyền cho trang con trong <jsp:include> và <jsp:forward>

```
<%@ page pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Insert title here</title>
</head>
<body>
    <h1>Main Page</h1>
    <jsp:include page="sub.jsp">
        <jsp:param value="Hello sub page" name="message"/>
    </jsp:include>
</body>
</html>
```

<h3>Sub Page</h3>
<%=param.message%>



- ❑ EL được sử dụng để tạo các biểu thức làm việc với các attribute trong các scope (page, request, session và application) cũng như truy xuất được các giá trị parameter, cookie.
- ❑ Trong phạm vi bài học này chỉ làm việc với attribute được tạo sẵn từ servlet đặt vào request, còn lại sẽ được nghiên cứu ở các bài sau.
- ❑ Ví dụ:
 - ❖ **`${item}`**
 - Truy xuất giá trị của attribute item trong request
 - ❖ **`${listindex}`**
 - Truy xuất giá trị phần tử thứ index của mảng hoặc List trong request
 - ❖ **`${mapkey}`**
 - Truy xuất giá trị phần tử có khóa là key của Map trong request
 - ❖ **`${bean.prop}`**
 - Truy xuất giá trị thuộc tính prop của Bean trong request

TRUY XUẤT ATTRIBUTE & BEAN PROPERTY

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
    req.setAttribute("message", "Welcome to FPT Polytechnic!");
    req.setAttribute("now", new Date());
    req.getRequestDispatcher("/page.jsp").forward(req, resp);
}
```

```
<%@page pageEncoding="utf-8" %>                                     page.jsp
<!DOCTYPE html>
<html>
<body>
    <h1>${message}</h1>
    <h2>${now.date}</h2> <!-- now.getDate() -->
</body>
</html>
```

```
Map<String, Double> diems = new HashMap<>();  
diems.put("toan", 5.0);  
diems.put("ly", 7.0);
```

```
List<String> tens = new ArrayList<>();  
tens.add("Phượng");  
tens.add("Hồng");
```

```
req.setAttribute("map", diems);  
req.setAttribute("list", tens);
```

```
req.getRequestDispatcher("/views/el.jsp").forward
```

Servlet

MAP:

- map.toan: 5.0 = 5.0
- map.ly: 7.0 = 7.0
- (map.ly + map.toan)/2: 3.5

LIST:

- list[0]: Phượng
- list[1]: Hồng

MAP:

```
<li>map.toan: ${map.toan} = ${map['toan']}</li>  
<li>map.ly: ${map.ly} = ${map['ly']}</li>  
<li>(map.ly + map.toan)/2: ${(map.ly + toan)/2}</li>
```

LIST:

```
<li>list[0]: ${list[0]}</li>  
<li>list[1]: ${list[1]}</li>
```

JSP



❑ Lớp JavaBean là lớp

- ❖ Phải là public
- ❖ Có Constructor mặc định (không tham số)
- ❖ Có getter và setter

❑ Property

- ❖ Thuộc tính được truy xuất trong EL được xác định bằng cách lấy tên của getter sau đó loại chữ get và đổi ký tự đầu tiên của phần còn lại sang ký tự thường.
- ❖ Ví dụ: `user.getUsername()` => `${user.username}`

Thành phần	Phương thức	Giá trị
URL	<code>\${pageContext.request.requestURL}</code>	http://localhost:8080/ctxpath/my-jsp/poly
URI	<code>\${pageContext.request.requestURI}</code>	/ctxpath/my-jsp/poly
QueryString	<code>\${pageContext.request.QueryString()}</code>	Name=Teo&age=25
Protocol	<code>\${pageContext.request.protocol}</code>	http
ServerName	<code>\${pageContext.request.serverName}</code>	localhost
Server Port	<code>\${pageContext.request.serverPort}</code>	8080
Context Path	<code>\${pageContext.request.contextPath}</code>	/ctxpath
Servlet Path	<code>\${pageContext.request.servletPath}</code>	/my-jsp
Path Info	<code>\${pageContext.request.pathInfo}</code>	/poly
Parameter	<code>\${param.age}</code> hoặc <code>\${param['age']}</code>	25
Method	<code>\${pageContext.request.method}</code>	GET



- ☑ Giới thiệu trang JSP
- ☑ JSP Directives
 - ❖ `<%@page%>`
 - ❖ `<%@include%>`
 - ❖ `<%@taglib%>`
- ☑ JSP Standard Actions:
 - ❖ `<jsp:include>`
 - ❖ `<jsp:forward>`
 - ❖ `<jsp:param>`
- ☑ Biểu thức EL cơ bản





Cảm ơn