



## LẬP TRÌNH JAVA 4

### BÀI 6: CUSTOM TAG, SIMPLE CUSTOM TAG, ATTRIBUTE CUSTOM TAG

#### PHẦN 1

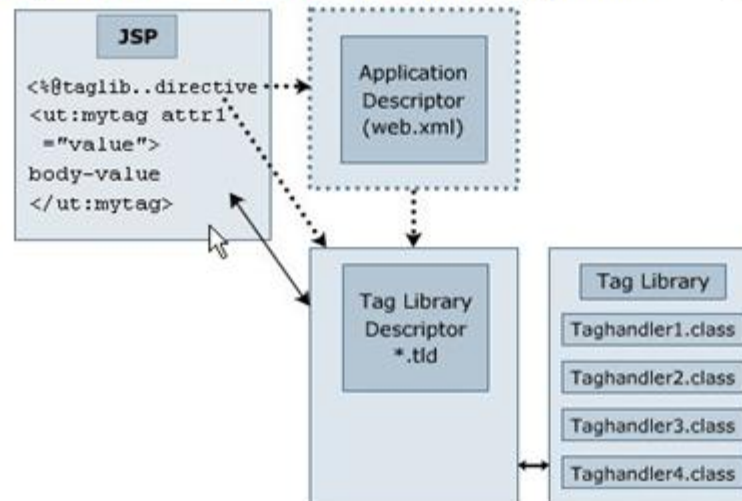
- ◎ Kết thúc bài học này bạn có khả năng
  - ❖ Custom tags là gì?
  - ❖ Tại sao cần Custom tags?
  - ❖ Tag Library
  - ❖ Simple Custom Tag
  - ❖ Attribute Custom Tag



- ☐ Là các phần tử JSP do User tự định nghĩa (ngược với các thẻ chuẩn tắc: standard tags)
- ☐ Đóng gói các tác vụ phải thực hiện nhiều lần
- ☐ Lấy từ thư viện thẻ (tag library) tự định nghĩa

- ❑ Được tùy biến thông qua các thuộc tính truyền từ trang JSP gọi chúng
- ❑ Có thể truyền lại tham số cho trang gọi
- ❑ Truy cập được tất cả các đối tượng có trong trang JSP
- ❑ Được lồng vào nhau, giao tiếp thông qua các biến cục bộ

```
<%@ taglib uri="/WEB-INF/tlds/TagHandler" prefix="u" %>
```



Working of Custom Tag Libraries

- ❑ Các thư viện thẻ (Custom Tag Library) đã có sẵn
  - ❖ **Core Tags:** Nhóm thẻ cơ bản
  - ❖ **Formatting tags:** Nhóm thẻ định dạng
  - ❖ **SQL tags:** Nhóm thẻ SQL
  - ❖ **XML tags:** Nhóm thẻ XML
  - ❖ **JSTL Functions:** Nhóm hàm JSTL

- ❑ Là tập các thẻ cùng chung mục đích
- ❑ Một hoặc nhiều thẻ có thể được đóng gói thành thư viện thẻ
- ❑ Các thành phần liên quan đến Tag Library
  - ❖ Tag Handler Class
    - Cách thức Tag xử lý
  - ❖ Tag Library Descriptor File
    - Mô tả Tag
  - ❖ JSP Page
    - Khai báo và sử dụng Tag

## ❑ Tag Handler Class

- ❖ Tag Handler chịu trách nhiệm xử lý trên Tag, chuyển Tag thành mã nguồn Java
- ❖ Tag Handler phải cài đặt lại `javax.servlet.jsp.tagext.Tag`
- ❖ Thường kế thừa từ `TagSupport` hoặc `BodyTagSupport`
- ❖ Tag Handler để trong Source Packages của ứng dụng web giống như Servlets, Java Beans

## ❑ Tag Library Descriptor File

- ❖ File XML mô tả

- Tag Name
- Các Attribute
- Chỉ định Tag Handler Class

- ❖ Thư viện được khai báo và sử dụng các Tag trong trang JSP

## ❑ JSP Page

- ❖ Import Tag Library - Tham chiếu đến URL của TLD
- ❖ Khai báo Tag Prefix Sử dụng các Tag



## ❑ Đặc điểm

- ❖ Không có thuộc tính
- ❖ Không có thân

## ❑ Định dạng

- ❖ `<prefix:TagName/>`

## ❑ Các bước xây dựng và sử dụng

- ❖ Bước 1: Tạo Tag Library Descriptor (TLD) mô tả thông tin về Simple Custom Tag
  - Có thể mô tả thông tin nhiều Tag
- ❖ Bước 2: Tạo lớp Tag Handler Class kế thừa TagSupport trong Source Packages
- ❖ Bước 3: Khai báo và sử dụng Simple Custom Tag trong JSP

## ❑ B1: Tạo Tag Library Descriptor (TLD)

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <taglib version="2.1"
3      xmlns="http://java.sun.com/xml/ns/javaee"
4      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5      xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
6      http://java.sun.com/xml/ns/javaee/web-
7      jsptaglibrary_2_1.xsd">
8      <tlib-version>1.0</tlib-version>
9      <short-name>Short Name</short-name>
10     <uri>URI</uri>
11     <tag>
12         <name>Tag Name</name>
13         <tag-class>Tag Handler Class</tag-class>
14         <body-content>empty</body-content>
15     </tag>
16 </taglib>

```

❖ <body-content>empty</body-Content> : Không có thân

## ❑ B2:Tạo Tag Handler Class kế thừa TagSupport

```
1 package packageName;
2 import java.io.IOException;
3 import java.util.logging.Level;
4 import java.util.logging.Logger;
5 import javax.servlet.jsp.JspWriter;
6 import javax.servlet.jsp.tagext.TagSupport;
7
8 public class TenTag extends TagSupport {
9     @Override
10    public int doStartTag() {
11        . . .
12        return SKIP_BODY;
13    }
14    @Override
15    public int doEndTag() {
16        return EVAL_PAGE;
17    }
18 }
```

## ❑ B2:Tạo Tag Handler Class kế thừa TagSupport

### ❖ Trong Tag Handler Class

#### ➤ Override doStartTag()

- ✓ Xử lý khi bắt đầu mở Tag
- ✓ Return SKIP\_BODY

#### ➤ Override doEndTag()

- ✓ Xử lý sau khi kết thúc thân Tag
- ✓ Return EVAL\_PAGE

### ❖ Trong phương thức doStartTag và doEndTag có thể sử dụng thuộc tính pageContext kế thừa được từ TagSupport

## ❑ B2:Tạo Tag Handler Class kế thừa TagSupport

❖ Một số phương thức hay dùng với pageContext

Đối tượng	Phương thức
out	JspWriter getOut()
session	HttpSession getSession()
request	ServletRequest getRequest()
response	ServletResponse getResponse()
exception	Exception getException()
errorData	ErrorData getErrorData()
page	Object getPage()
config	ServletConfig getServletConfig()
context	ServletContext getServletContext()

❖ Include và Forward với pageContext

Thao tác	Phương thức
Include	void include(String relativeURLPath)
Forward	void forward(String relativeURLPath)

## ❑ B3: Khai báo và sử dụng trong JSP

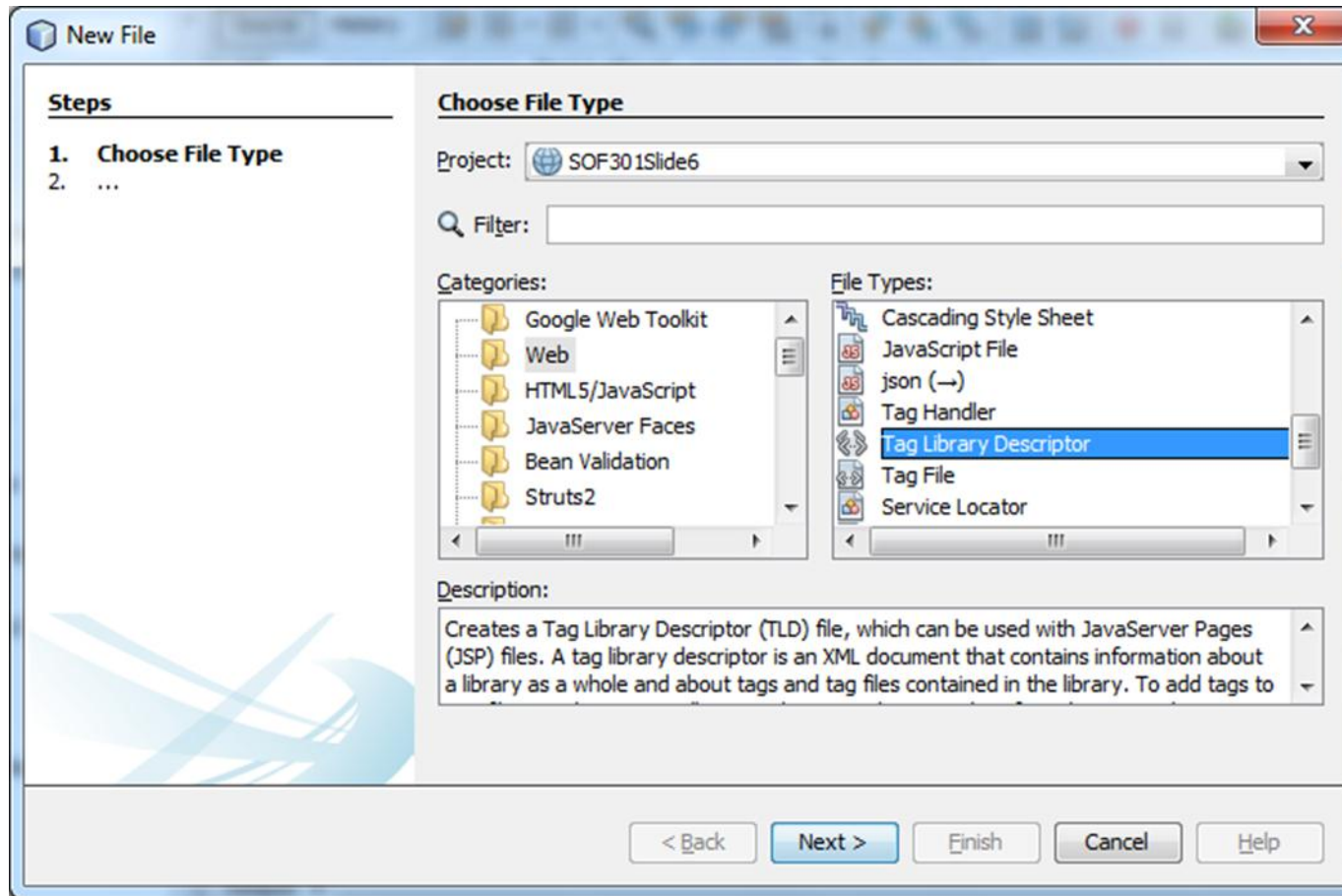
### ❖ Khai báo:

```
<%@taglib prefix="prefixName" uri="/WEB-INF/tlds/name.tld" %>
```

### ❖ Sử dụng:

```
<prefixName:TagName/>
```

## ❑ B1: Tạo Tag Handler Class kế thừa TagSupport





## ❑ B1: Tạo Tag Handler Class kế thừa TagSupport

New Tag Library Descriptor

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

TLD Name: HelloTag

Project: SOF301Slide6

Location: WEB-INF

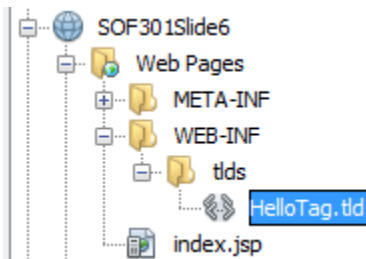
Folder: tlds/ Browse...

Created File: D:\DemoSOF301\Slide5\SOF301Slide6\web\WEB-INF\tlds\HelloTag.tld

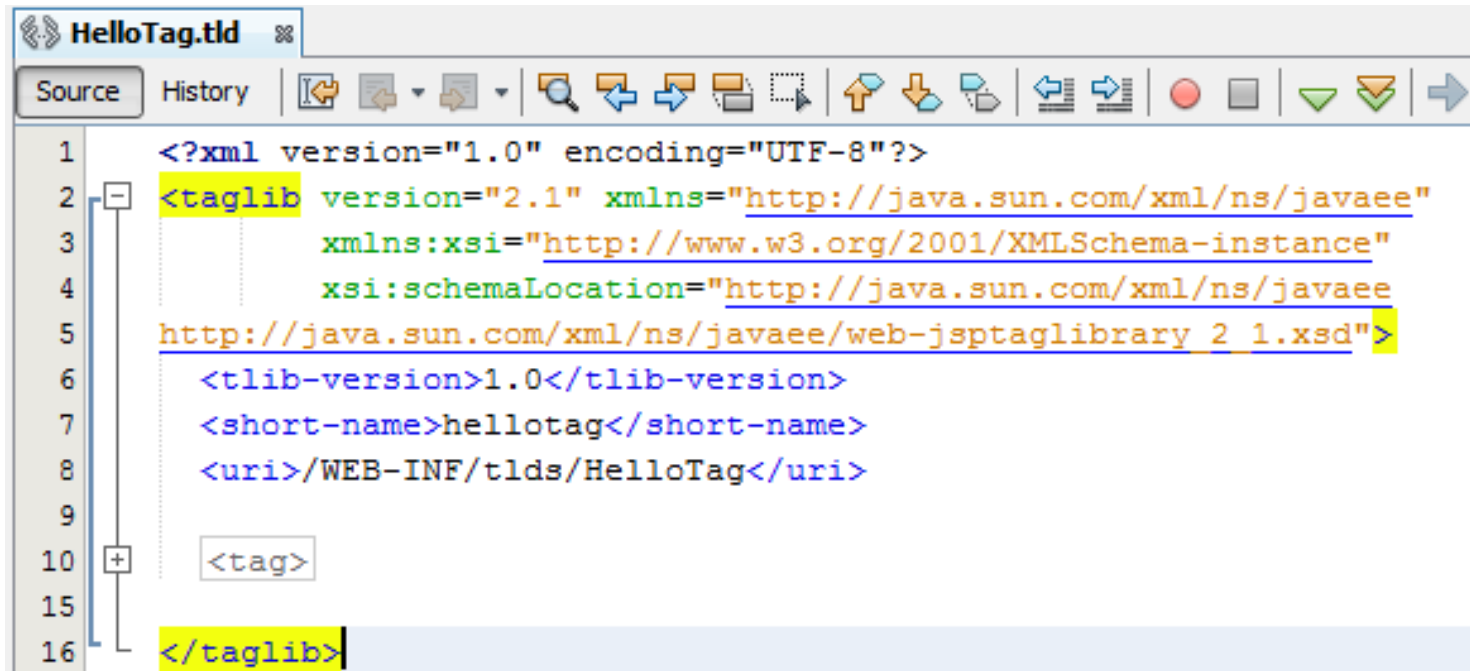
URI: /WEB-INF/tlds/HelloTag

Prefix: hellotag

< Back Next > Finish Cancel Help

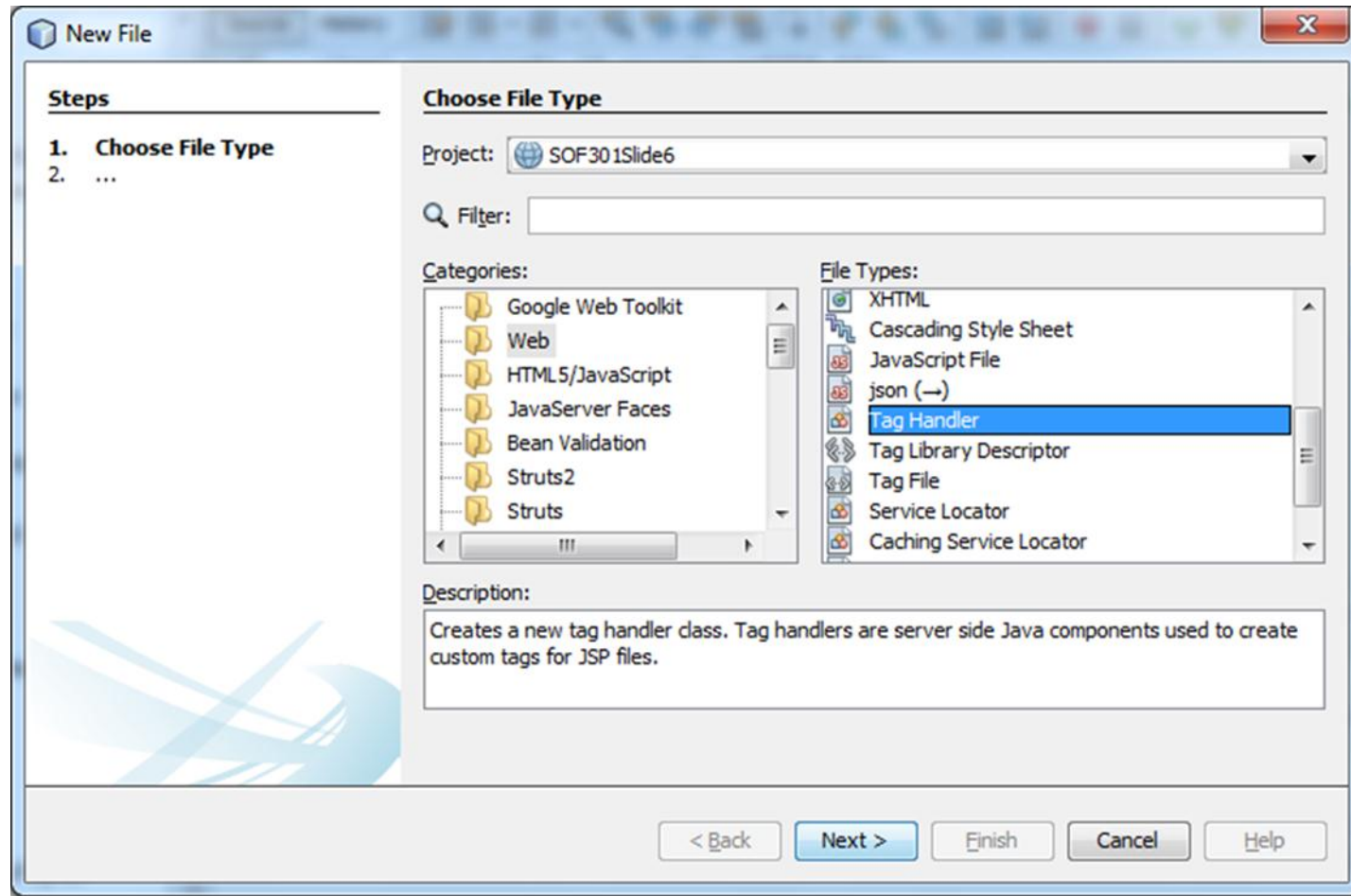


## ❑ B1: Tạo Tag Library Descriptor (TLD)



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <taglib version="2.1" xmlns="http://java.sun.com/xml/ns/javaee"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
5       http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd">
6     <tlib-version>1.0</tlib-version>
7     <short-name>hellotag</short-name>
8     <uri>/WEB-INF/tlds/HelloTag</uri>
9
10    <tag>
15
16  </taglib>
```

## ❑ B2: Tạo Tag Handler Class kế thừa TagSupport



## ❑ B2: Tạo Tag Handler Class kế thừa TagSupport

**New Tag Handler**

**Steps**

1. Choose File Type
2. **Name and Location**
3. TLD Information

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

Tag Support Class to Extend:

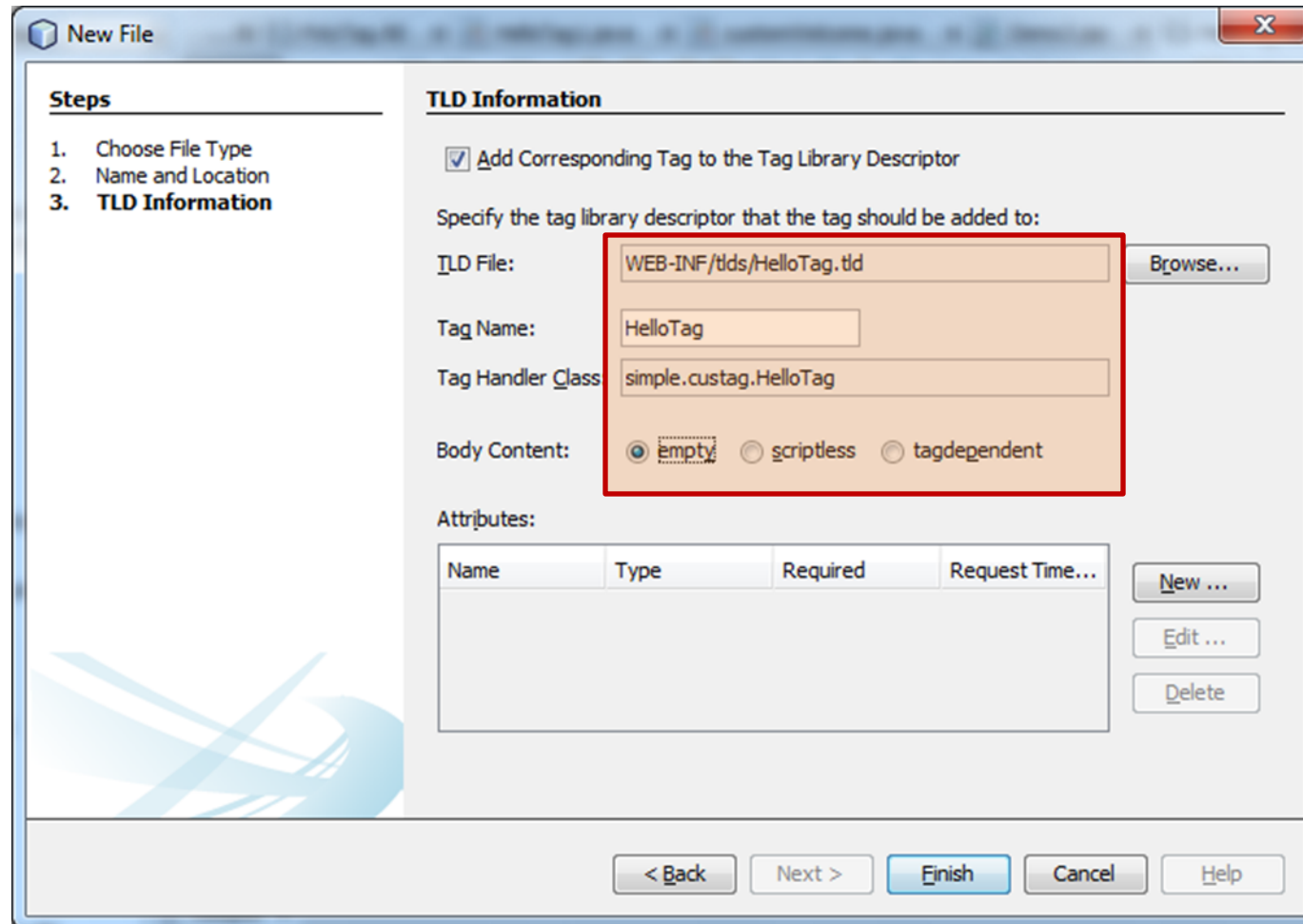
☒ SimpleTagSupport (J2EE 1.4)

☐ BodyTagSupport

Description:

< Back   Next >   Finish   Cancel   Help

## ❑ B2: Tạo Tag Handler Class kế thừa TagSupport



**New File**

**Steps**

1. Choose File Type
2. Name and Location
3. **TLD Information**

**TLD Information**

☒ Add Corresponding Tag to the Tag Library Descriptor

Specify the tag library descriptor that the tag should be added to:

TLD File:

Tag Name:

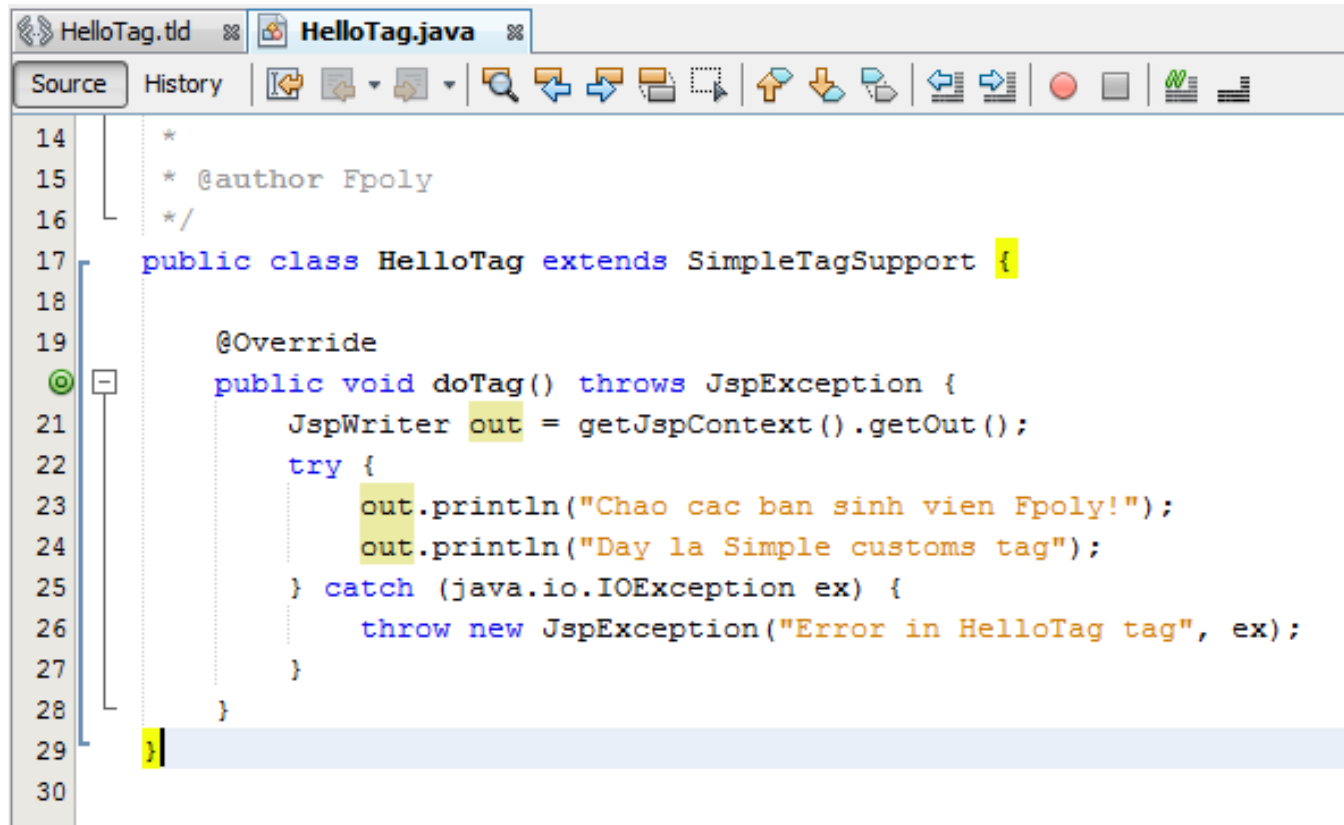
Tag Handler Class:

Body Content: ☒ empty ☐ scriptless ☐ tagdependent

Attributes:

Name	Type	Required	Request Time...
------	------	----------	-----------------

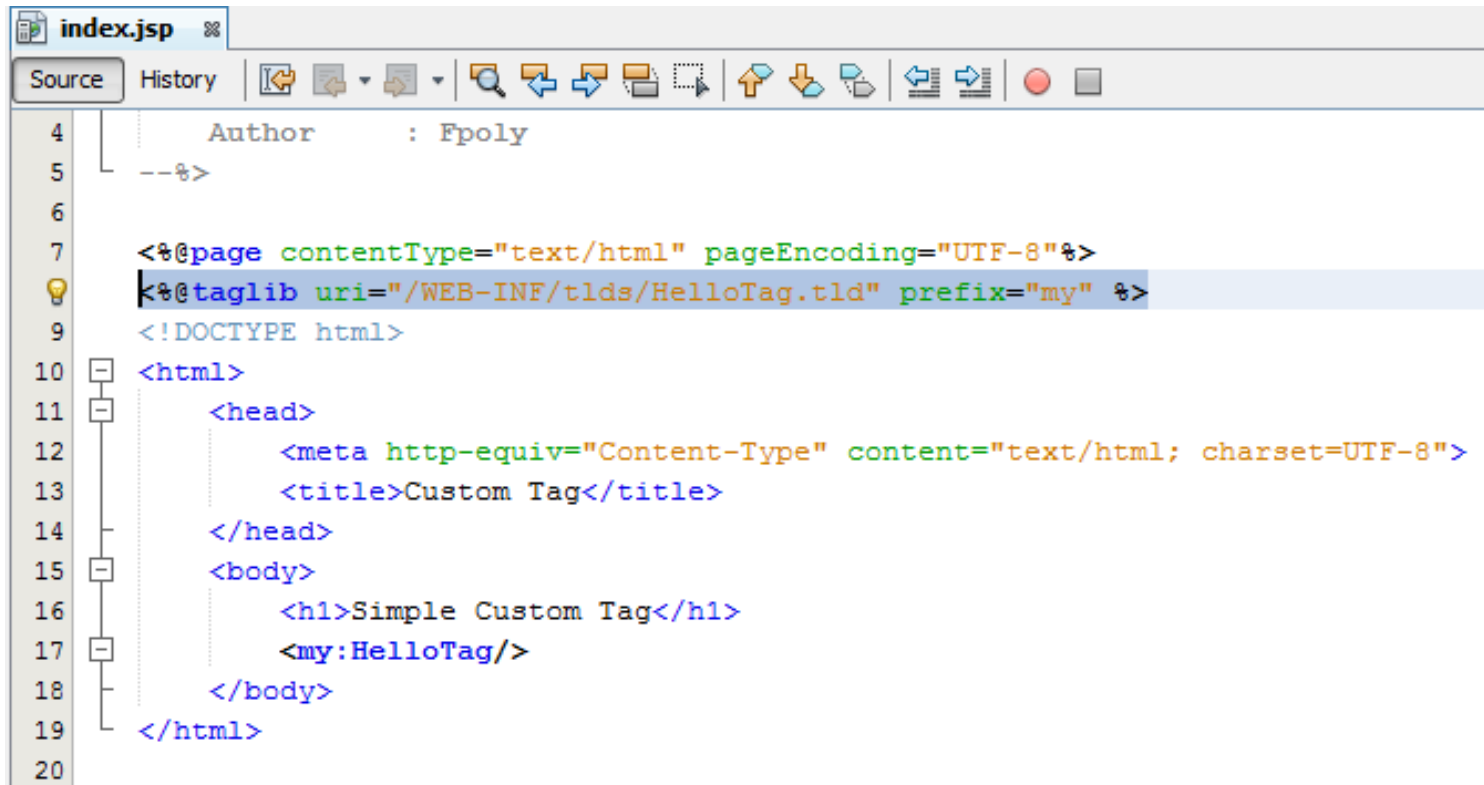
## ❑ B2:Tạo Tag Handler Class kế thừa TagSupport



```
14  *
15  * @author Fpoly
16  */
17  public class HelloTag extends SimpleTagSupport {
18
19      @Override
20      public void doTag() throws JspException {
21          JspWriter out = getJspContext().getOut();
22          try {
23              out.println("Chao cac ban sinh vien Fpoly!");
24              out.println("Day la Simple customs tag");
25          } catch (java.io.IOException ex) {
26              throw new JspException("Error in HelloTag tag", ex);
27          }
28      }
29  }
```

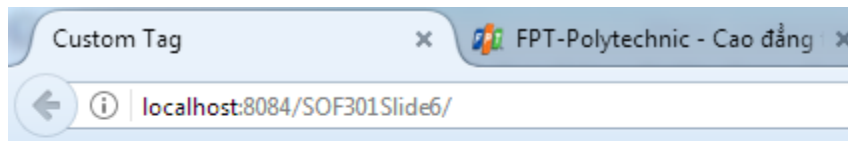
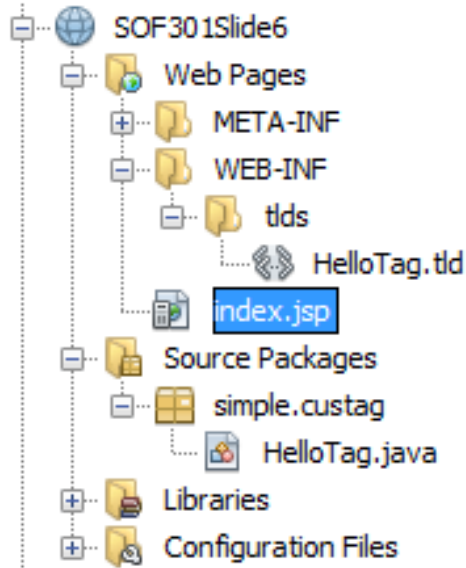
```
<uri>/WEB-INF/tlds/HelloTag</uri>
<tag>
    <name>HelloTag</name>
    <tag-class>simple.custag.HelloTag</tag-class>
    <body-content>empty</body-content>
</tag>
```

## ❑ B3: Khai báo và sử dụng trong JSP



```
index.jsp
Source History
4      Author      : Fpoly
5      -->
6
7      <%@page contentType="text/html" pageEncoding="UTF-8"%>
8      <%@taglib uri="/WEB-INF/tlds/HelloTag.tld" prefix="my" %>
9      <!DOCTYPE html>
10     <html>
11     <head>
12         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
13         <title>Custom Tag</title>
14     </head>
15     <body>
16         <h1>Simple Custom Tag</h1>
17         <my:HelloTag/>
18     </body>
19 </html>
20
```

## ❑ B3: Khai báo và sử dụng trong JSP



## Simple Custom Tag

Chào các bạn sinh viên Fpoly! Đây là Simple customs tag





# DEMO

Chạy và giải thích





## LẬP TRÌNH JAVA 4

### BÀI 6: CUSTOM TAG, SIMPLE CUSTOM TAG, ATTRIBUTE CUSTOM TAG

#### PHẦN 2

## ❑ Đặc điểm

- ❖ Có một hay nhiều thuộc tính
- ❖ Không có thân

## ❑ Định dạng

- ❖ `<prefix:TagName attribute1 = "... "  
attribute2 = "... "  
attributeN = "... " />`

## ❑ B1:Tạo Tag Handler Class kế thừa TagSupport

```
1  public class TagName extends TagSupport {
2      private String attribute1;
3      private String attribute2;
4      ...
5      private String attributeN;
6      public TagName() {
7          this.attribute1=...
8          this.attribute2=...
9          . . .
10         this.attributeN=...
11     }
12     public void setAttribute1(String value1) {attribute1=value1;}
13     public void setAttribute2(String value2) {attribute2=value2;}
14     ...
15     public void setAttributeN(String valueN) {attributeN=valueN;}
16     ...
17 }
```

## ❑ B1:Tạo Tag Handler Class kế thừa TagSupport

```
1 public class TagName extends TagSupport {
2     @Override
3     public int doStartTag() {
4         //Sử dụng các thuộc tính
5         ...
6         return SKIP_BODY;
7     }
8     @Override
9     public int doEndTag() {
10         return EVAL_PAGE;
11     }
12     @Override
13     public void release(){
14         this.attribute1=...
15         this.attribute2=...
16         . . .
17         this.attributeN=...
18         //Release ...
19     }
20 }
```

## ❑ B2: Tạo Tag Library Descriptor (TLD)

```
1 <taglib> <tlib-version>1.0</tlib-version>
2   <short-name>customtaglibrary</short-name><uri>URI TLD</uri>
3   <tag>
4     <name>Tag Name</name>
5     <tag-class>Tag Handler Class</tag-class>
6     <attribute>
7       <name>attribute1</name>
8       <required>true/false</required>
9       <rtexprvalue>true/false</rtexprvalue>
10    </attribute>
11    . . .
12    <attribute>
13      <name>attributeN</name>
14      <required>true/false</required>
15      <rtexprvalue>true/false</rtexprvalue>
16    </attribute>
17    <body-content>empty</body-content>
18  </tag>
19 </taglib>
```

## ❑ B2: Tạo Tag Library Descriptor (TLD)

- ❖ `<required>true</required>` : Thuộc tính bắt buộc sử dụng trong Tag
- ❖ `<required>false</required>` : Thuộc tính tùy chọn sử dụng trong Tag
- ❖ `<rtexprvalue>true</rtexprvalue>` : Thuộc tính có thể gán giá trị bằng JSP Expression
- ❖ `<rtexprvalue>false</rtexprvalue>` : Thuộc tính không được gán giá trị bằng JSP Expression
- ❖ `<body-content>empty</body-Content>` : Không có thân

## ❑ B3: Khai báo và sử dụng trong JSP

❖ Khai báo:

```
<%@taglib prefix="prefixName" uri="URI TLD" %>
```

❖ Sử dụng:

```
<prefixName:TagName attribute1="..."  
    attribute2="..."  
    ...  
    attributeN="..."/>
```



## ❑ B1: Tạo Tag Handler Class

**New Tag Handler**

**Steps**

1. Choose File Type
- 2. Name and Location**
3. TLD Information

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

Tag Support Class to Extend:

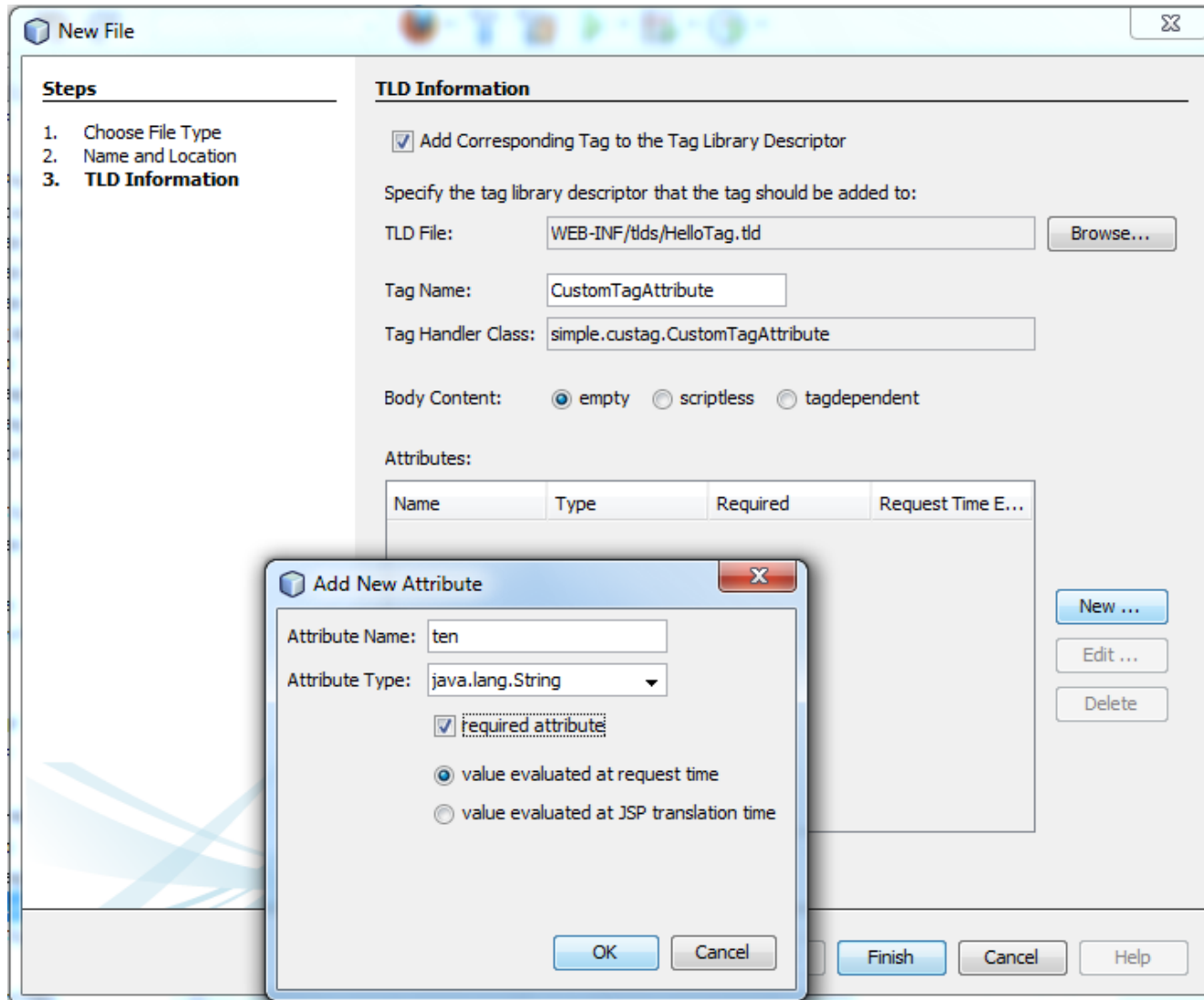
☒ SimpleTagSupport (J2EE 1.4)

☐ BodyTagSupport

Description:

< Back   Next >   Finish   Cancel   Help

## ❑ B1: Tạo Tag Handler Class



## ❑ B1: Tạo Tag Handler Class

**New File**

**Steps**

1. Choose File Type
2. Name and Location
3. **TLD Information**

**TLD Information**

☒ Add Corresponding Tag to the Tag Library Descriptor

Specify the tag library descriptor that the tag should be added to:

TLD File:

Tag Name:

Tag Handler Class:

Body Content: ☒ empty ☐ scriptless ☐ tagdependent

Attributes:

Name	Type	Required	Request Time E...
ten	java.lang.String	true	true

**Add New Attribute**

Attribute Name:

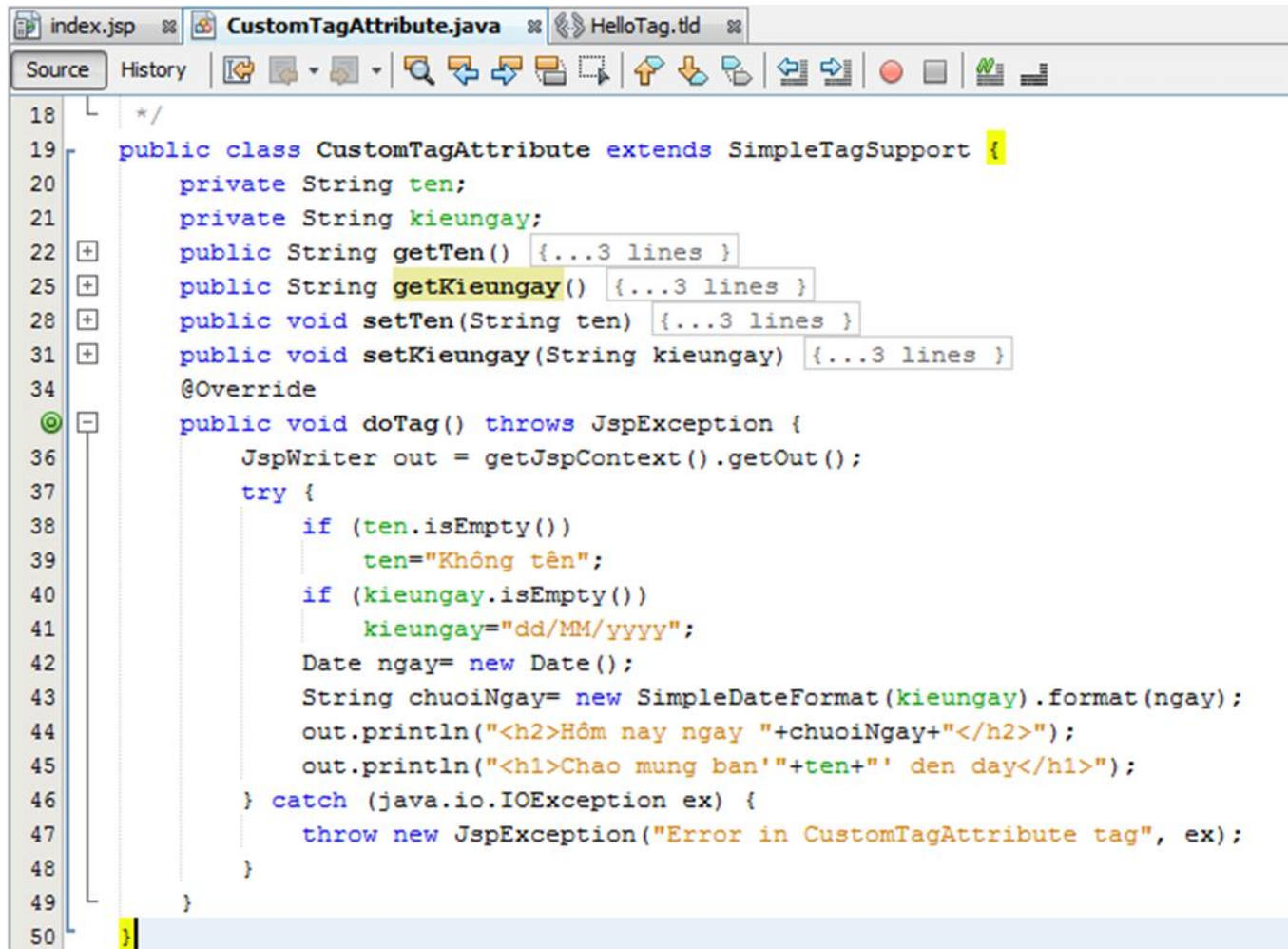
Attribute Type:

☐ required attribute

☒ value evaluated at request time

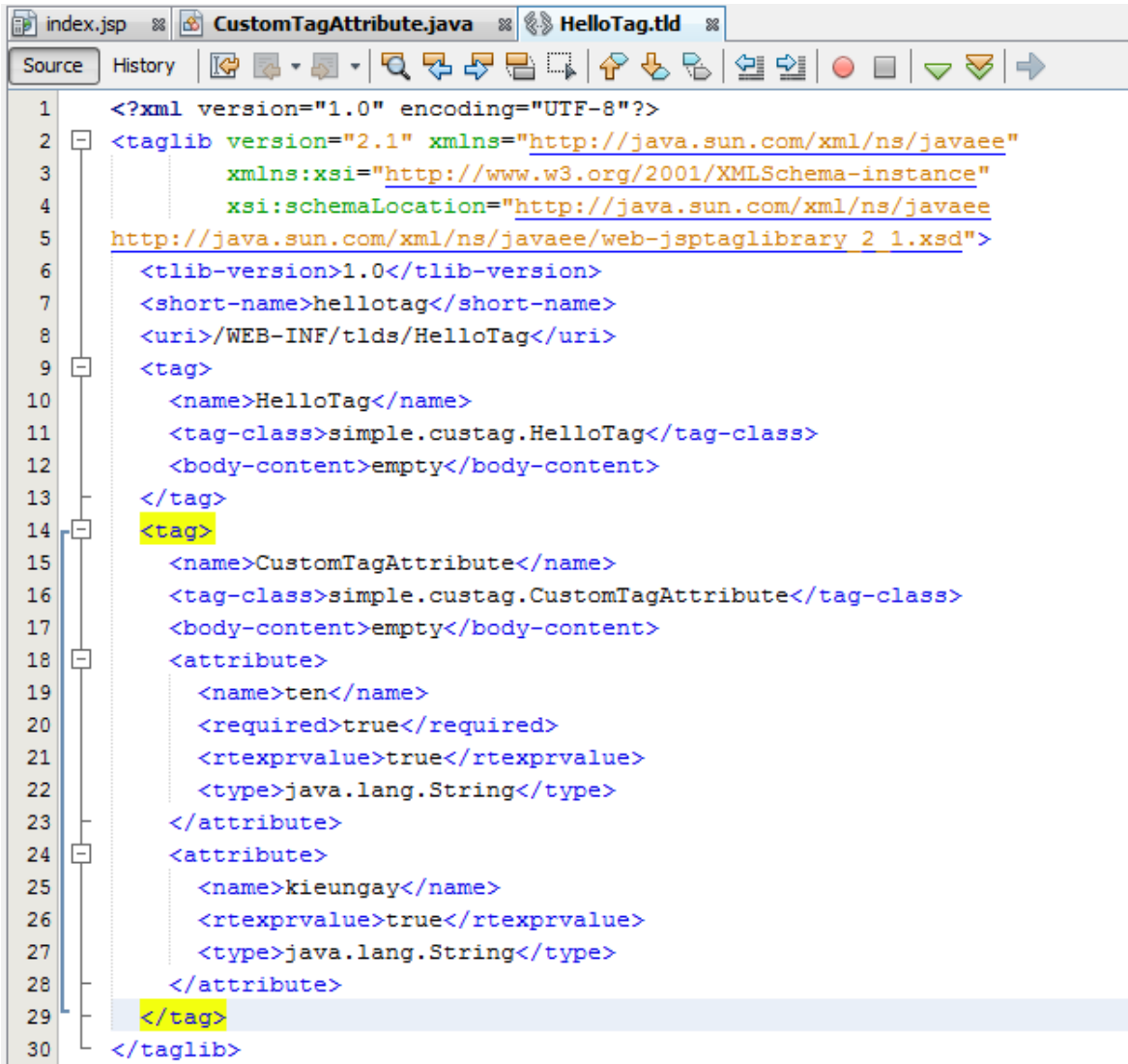
☐ value evaluated at JSP translation time

## ❑ B1: Tạo Tag Handler Class



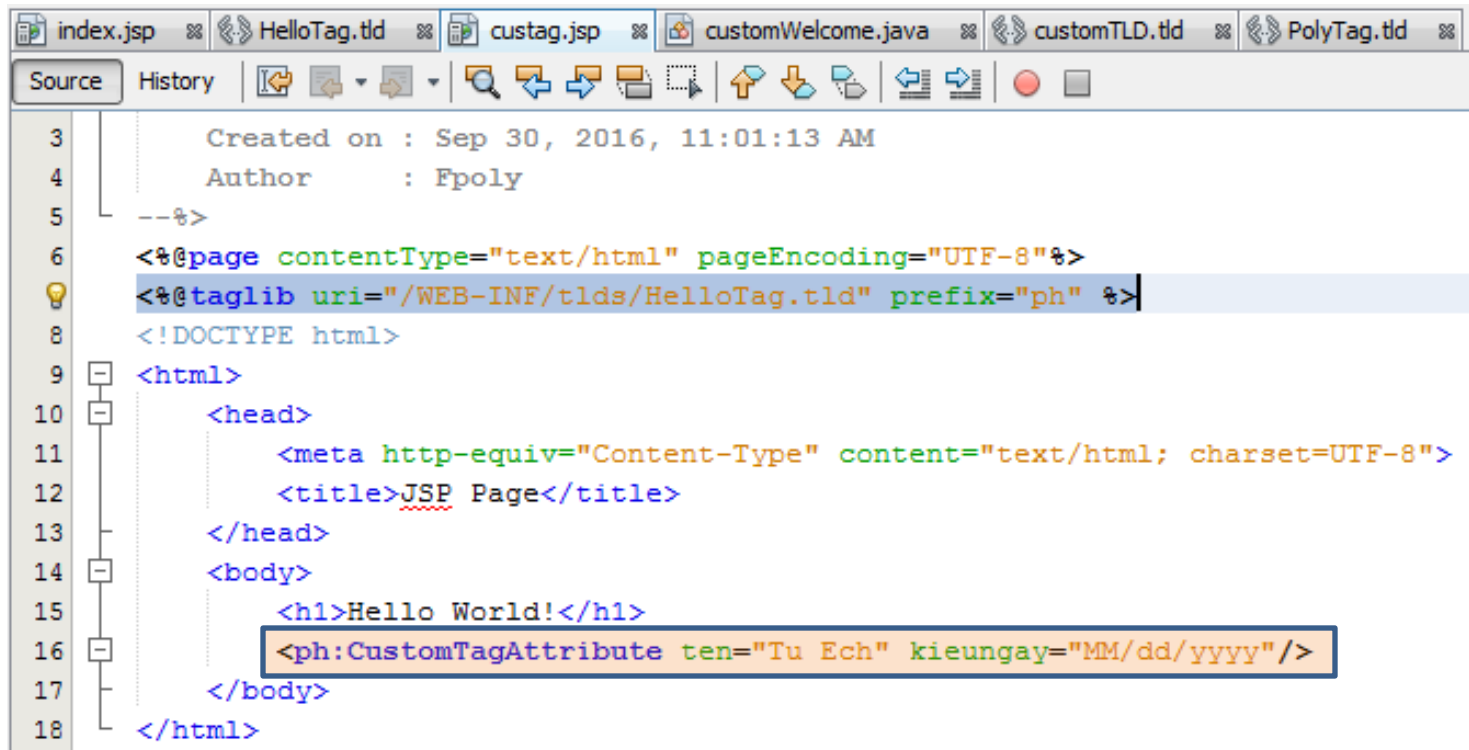
```
18  /*
19  public class CustomTagAttribute extends SimpleTagSupport {
20      private String ten;
21      private String kieungay;
22      public String getTen() { ...3 lines }
23      public String getKieungay() { ...3 lines }
24      public void setTen(String ten) { ...3 lines }
25      public void setKieungay(String kieungay) { ...3 lines }
26      @Override
27      public void doTag() throws JspException {
28          JspWriter out = getJspContext().getOut();
29          try {
30              if (ten.isEmpty())
31                  ten="Không tên";
32              if (kieungay.isEmpty())
33                  kieungay="dd/MM/yyyy";
34              Date ngay= new Date();
35              String chuoiNgay= new SimpleDateFormat(kieungay).format(ngay);
36              out.println("<h2>Hôm nay ngày "+chuoiNgay+"</h2>");
37              out.println("<h1>Chao mung ban'+ten+' den day</h1>");
38          } catch (java.io.IOException ex) {
39              throw new JspException("Error in CustomTagAttribute tag", ex);
40          }
41      }
42  }
```

## ❑ B2: Tạo Tag Library Descriptor (TLD)



```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <taglib version="2.1" xmlns="http://java.sun.com/xml/ns/javaee"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
5      http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd">
6      <tlib-version>1.0</tlib-version>
7      <short-name>hellotag</short-name>
8      <uri>/WEB-INF/tlds/HelloTag</uri>
9      <tag>
10         <name>HelloTag</name>
11         <tag-class>simple.custag.HelloTag</tag-class>
12         <body-content>empty</body-content>
13     </tag>
14     <tag>
15         <name>CustomTagAttribute</name>
16         <tag-class>simple.custag.CustomTagAttribute</tag-class>
17         <body-content>empty</body-content>
18         <attribute>
19             <name>ten</name>
20             <required>true</required>
21             <rtexprvalue>true</rtexprvalue>
22             <type>java.lang.String</type>
23         </attribute>
24         <attribute>
25             <name>kieungay</name>
26             <rtexprvalue>true</rtexprvalue>
27             <type>java.lang.String</type>
28         </attribute>
29     </tag>
30 </taglib>
```

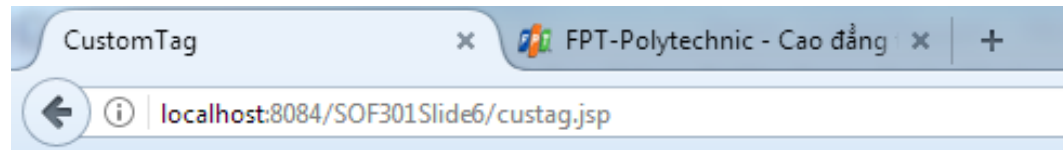
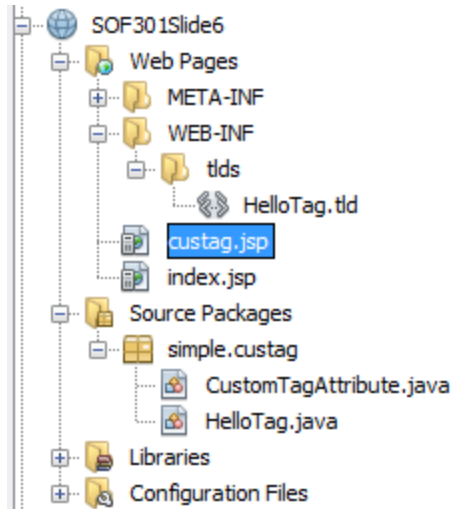
## ❑ B3: Khai báo và sử dụng trong JSP



The screenshot shows an IDE with several tabs: index.jsp, HelloTag.tld, custag.jsp, customWelcome.java, customTLD.tld, and PolyTag.tld. The 'Source' tab is active, displaying the code for index.jsp. The code includes a page declaration, a taglib declaration for 'ph' (pointing to 'HelloTag.tld'), and an HTML document structure. A custom tag is used in the body: <ph:CustomTagAttribute ten="Tu Ech" kieungay="MM/dd/yyyy"/>. The IDE interface includes a toolbar with various editing and navigation icons.

```
3      Created on : Sep 30, 2016, 11:01:13 AM
4      Author      : Fpoly
5      --%>
6      <%@page contentType="text/html" pageEncoding="UTF-8"%>
7      <%@taglib uri="/WEB-INF/tlds/HelloTag.tld" prefix="ph" %>
8      <!DOCTYPE html>
9      <html>
10     <head>
11         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12         <title>JSP Page</title>
13     </head>
14     <body>
15         <h1>Hello World!</h1>
16         <ph:CustomTagAttribute ten="Tu Ech" kieungay="MM/dd/yyyy"/>
17     </body>
18 </html>
```

## ❑ B3: Khai báo và sử dụng trong JSP



**Custom Tag**

**Hôm nay ngày 07/03/2017**

**Chao mung ban'Tu Ech' den day**



# DEMO

Chạy và giải thích





- ❖ Custom tags là gì?
- ❖ Tại sao cần Custom tags?
- ❖ TAG LIBRARY
- ❖ Simple Custom Tag
- ❖ Attribute Custom Tag





**Cảm ơn**