

CHUYÊN ĐỀ JAVA

JSP CUSTOM TAG

Nguyễn Hoàng Anh – nhanh@fit.hcmus.edu.vn

Nội dung trình bày

- Tag Library
- Custom Tag
 - Simple Custom Tag
 - Attribute Custom Tag
 - Body Custom Tag
 - Nested Custom Tag

TAG LIBRARY

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

SIMPLE CUSTOM TAG

Simple Custom Tag

- Đặc điểm
 - Không có thuộc tính
 - Không có thân
- Định dạng
<prefix:TagName/>

Các bước xây dựng và sử dụng

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

B1:Tạo Tag Handler Class kế thừa TagSupport

```
1  package packageName;
2  import java.io.IOException;
3
4  import java.util.logging.Level;
5  import java.util.logging.Logger;
6  import javax.servlet.jsp.JspWriter;
7  import javax.servlet.jsp.tagext.TagSupport;
8
9
10 public class TenTag extends TagSupport {
11     @Override
12     public int doStartTag() {
13         . . .
14         return SKIP_BODY;
15     }
16     @Override
17     public int doEndTag() {
18         return EVAL_PAGE;
19     }
20 }
```

Bước 1: Tạo lớp Tag Handler Class

- Trong Tag Handler Class
 - Cài đặt lại doStartTag()
 - Xử lý khi bắt đầu mở Tag
 - Return SKIP_BODY
 - Cài đặt lại doEndTag()
 - Xử lý sau khi kết thúc thân Tag
 - Return EVAL_PAGE
- Trong doStartTag và doEndTag có thể sử dụng
 - thuộc tính pageContext kế thừa được từ TagSupport

B1: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()

B1:Tạo Tag Handler Class kế thừa TagSupport

- Include và Forward với pageContext

Thao tác	Phương thức
Include	<code>void include(String relativeURLPath)</code>
Forward	<code>void forward(String relativeURLPath)</code>

B2: Tạo Tag Library Descriptor (TLD)

- File XML mô tả
 - Tag Name
 - Các Attribute
 - Chỉ định Tag Handler Class
- Tạo TLD trong Web Pages

B2: 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>
```


B2: Tạo Tag Library Descriptor (TLD)

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

B3: Khai báo và sử dụng trong JSP

- Khai báo

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

- Sử dụng

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

B1:Tạo Tag Handler Class kế thừa TagSupport

```
1 public class HelloTag extends TagSupport {
2     @Override
3     public int doStartTag() {
4         JspWriter out = this.pageContext.getOut();
5         try {
6             out.println("<b>Xin chào Nguyễn Hoàng Anh</b>");
7             out.println("<b>This is a Simple Custom Tag</b>");
8         } catch (IOException ex) {
9             Logger.getLogger(HelloTag.class.getName())
10                .log(Level.SEVERE, null, ex);
11        }
12        return SKIP_BODY;
13    }
14    @Override
15    public int doEndTag() {
16        return EVAL_PAGE;
17    }
18 }
19
```

B2: 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>customtaglibrary</short-name>
10  <uri>/WEB-INF/tlds/CustomTagLibrary</uri>
11  <tag>
12    <name>HelloTag</name>
13    <tag-class>customs.HelloTag</tag-class>
14    <body-content>empty</body-content>
15  </tag>
16 </taglib>
```

B3: Khai báo và sử dụng trong JSP

- Khai báo

```
<%@taglib  
    prefix="nhanh"  
    uri="/WEB-INF/tlds/CustomTagLibrary" %>
```

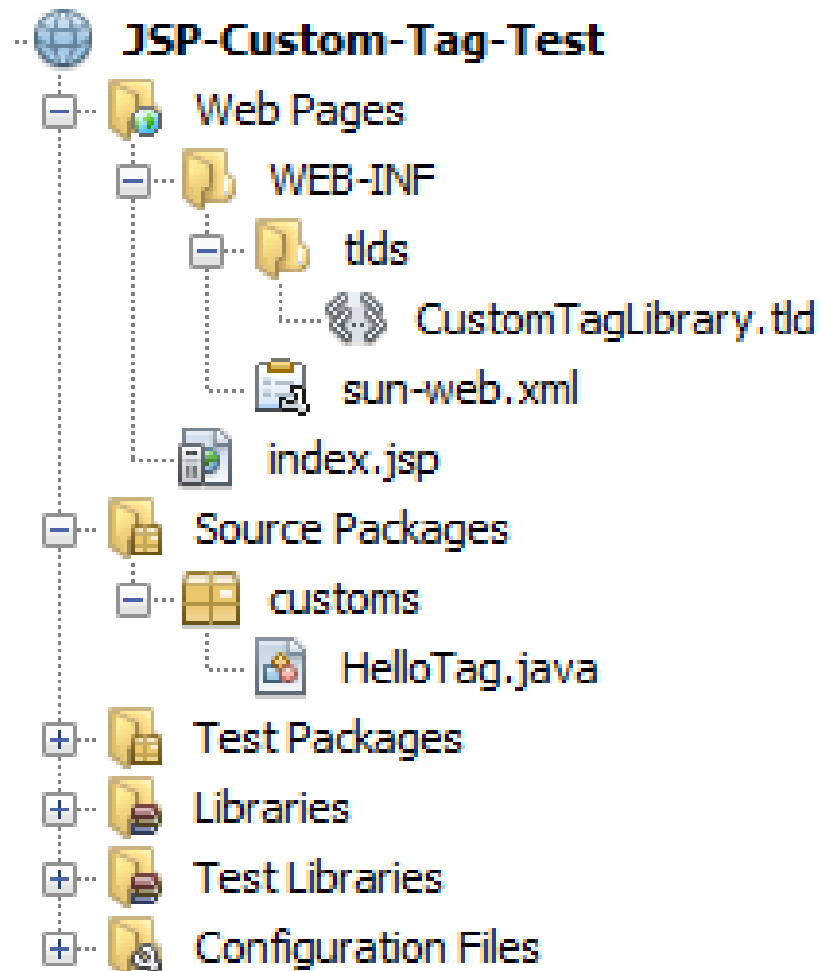
- Sử dụng

```
<nhanh:HelloTag/>
```

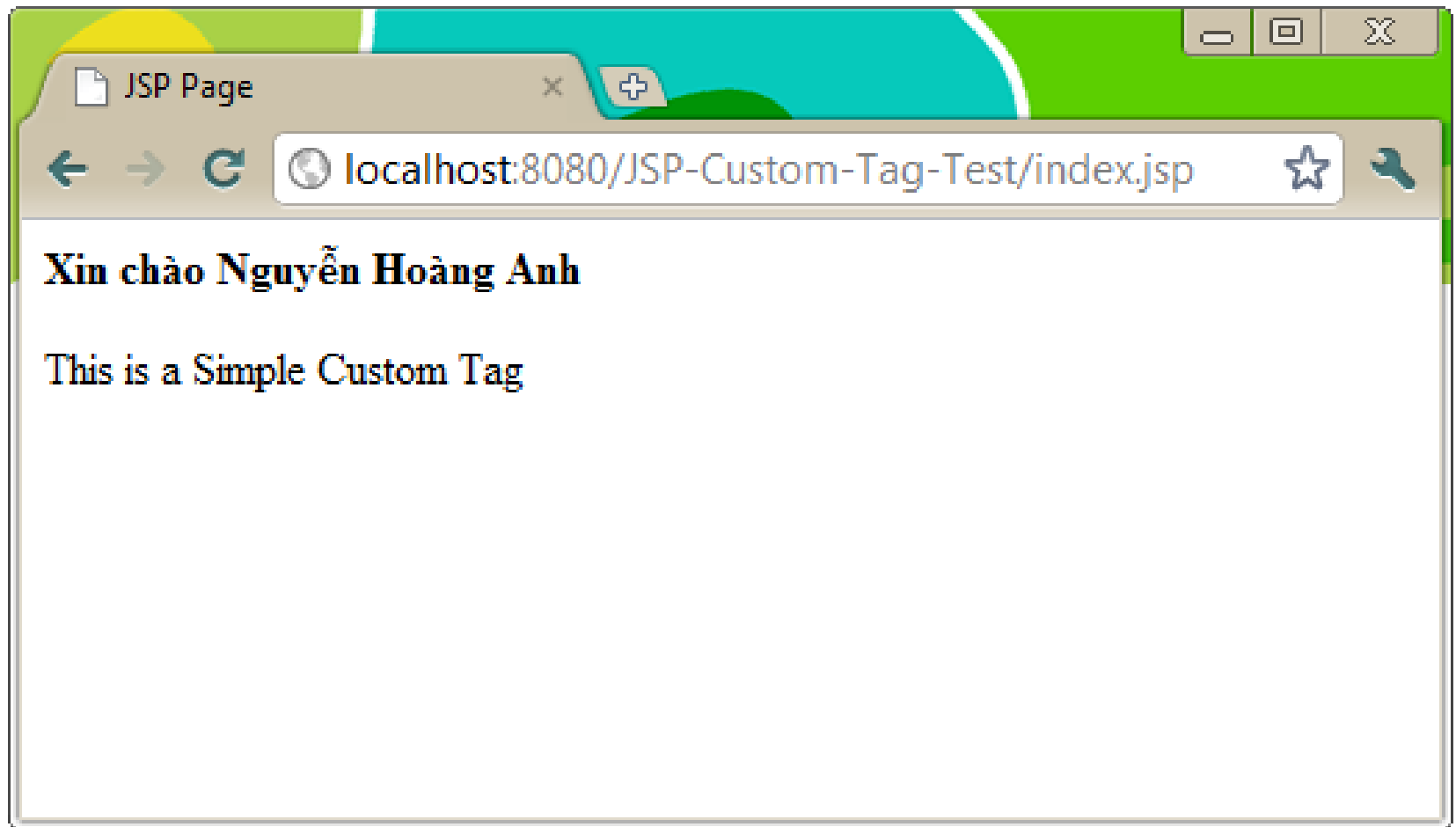
B3: Khai báo và sử dụng trong JSP

```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2
3 <%@taglib prefix="nhanh"
4     uri="/WEB-INF/tlds/CustomTagLibrary.tld" %>
5
6 <html>
7     <head>
8         <meta http-equiv="Content-Type"
9             content="text/html; charset=UTF-8">
10         <title>JSP Page</title>
11     </head>
12     <body>
13         <nhanh:HelloTag/>
14     </body>
15 </html>
```

Kết quả



Kết quả



CUSTOM TAG WITH ATTRIBUTES

Custom Tag có thuộc tính

- Đặc điểm
 - Có một hay nhiều thuộc tính
 - Không có thân

- Định dạng

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

Các bước xây dựng và sử dụng

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

B1:Tạo Tag Handler Class kế thừa TagSupport

- Để tạo được Custom Tag có thuộc tính

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

- Trong Tag Handler Class
 - Thêm thuộc tính attribute1, attribute2,...,attributeN
 - Thêm các phương thức:

```
public void setAttribute1(String value1) {...}  
public void setAttribute2(String value2) {...}  
...  
public void setAttributeN(String valueN) {...}
```

- Cài đặt lại phương thức doStartTag, doEndTag

Bước 1: Tạo lớp Tag Handler Class

- Trong Tag Handler Class
 - Cài đặt lại doStartTag()
 - Xử lý khi bắt đầu mở Tag
 - Return SKIP_BODY
 - Cài đặt lại doEndTag()
 - Xử lý sau khi kết thúc thân Tag
 - Return EVAL_PAGE
- Trong doStartTag và doEndTag có thể sử dụng
 - thuộc tính pageContext kế thừa được từ TagSupport

B1:Tạo Tag Handler Class kế thừa TagSupport

- Một số phương thức hay dùng với pageContext

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

B1:Tạo Tag Handler Class kế thừa TagSupport

- Include và Forward với pageContext

Thao tác	Phương thức
Include	<code>void include(String relativeURLPath)</code>
Forward	<code>void forward(String relativeURLPath)</code>

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

```
1  package customs;
2
3  import java.io.IOException;
4  import java.util.logging.Level;
5  import java.util.logging.Logger;
6  import javax.servlet.jsp.JspWriter;
7  import javax.servlet.jsp.tagext.TagSupport;
8
9  public class EmployeeTag extends TagSupport {
10
11     private String employeeID;
12     private String employeeName;
13
14     public EmployeeTag() {
15         this.employeeID = "";
16         this.employeeName = "";
17     }
18     . . .
19
20
```

B1:Tạo Tag Handler Class kế thừa TagSupport

```
1  public String getEmployeeID() {  
2      return employeeID;  
3  }  
4  
5  public void setEmployeeID(String employeeID) {  
6      this.employeeID = employeeID;  
7  }  
8  
9  public String getEmployeeName() {  
10     return employeeName;  
11 }  
12  
13 public void setEmployeeName(String employeeName) {  
14     this.employeeName = employeeName;  
15 }  
16  
17  
18  
19  
20
```

B1:Tạo Tag Handler Class kế thừa TagSupport

```
1      @Override
2      public int doStartTag() {
3          JspWriter out = this.pageContext.getOut();
4          try {
5              out.println("<p>" + this.employeeID + " - "
6                          + this.employeeName + "</p>");
7          } catch (IOException ex) {
8              Logger.getLogger(EmployeeTag.class.getName()).
9                  log(Level.SEVERE, null, ex);
10         }
11
12         return SKIP_BODY;
13     }
14
15     @Override
16     public void release() {
17         this.employeeID = null;
18         this.employeeName = null;
19     }
20 }
```

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>customtaglibrary</short-name>
8   <uri>/WEB-INF/tlds/CustomTagLibrary</uri>
9
10   <tag>
11     . . .
12   </tag>
13
14 </taglib>
```


B2: Tạo Tag Library Descriptor (TLD)

```
1 <tag>
2     <name>EmployeeTag</name>
3     <tag-class>customs.EmployeeTag</tag-class>
4     <attribute>
5         <name>employeeID</name>
6         <required>true</required>
7         <rtexprvalue>true</rtexprvalue>
8     </attribute>
9     <attribute>
10         <name>employeeName</name>
11         <required>true</required>
12         <rtexprvalue>true</rtexprvalue>
13     </attribute>
14     <body-content>empty</body-content>
15 </tag>
16 </taglib>
```

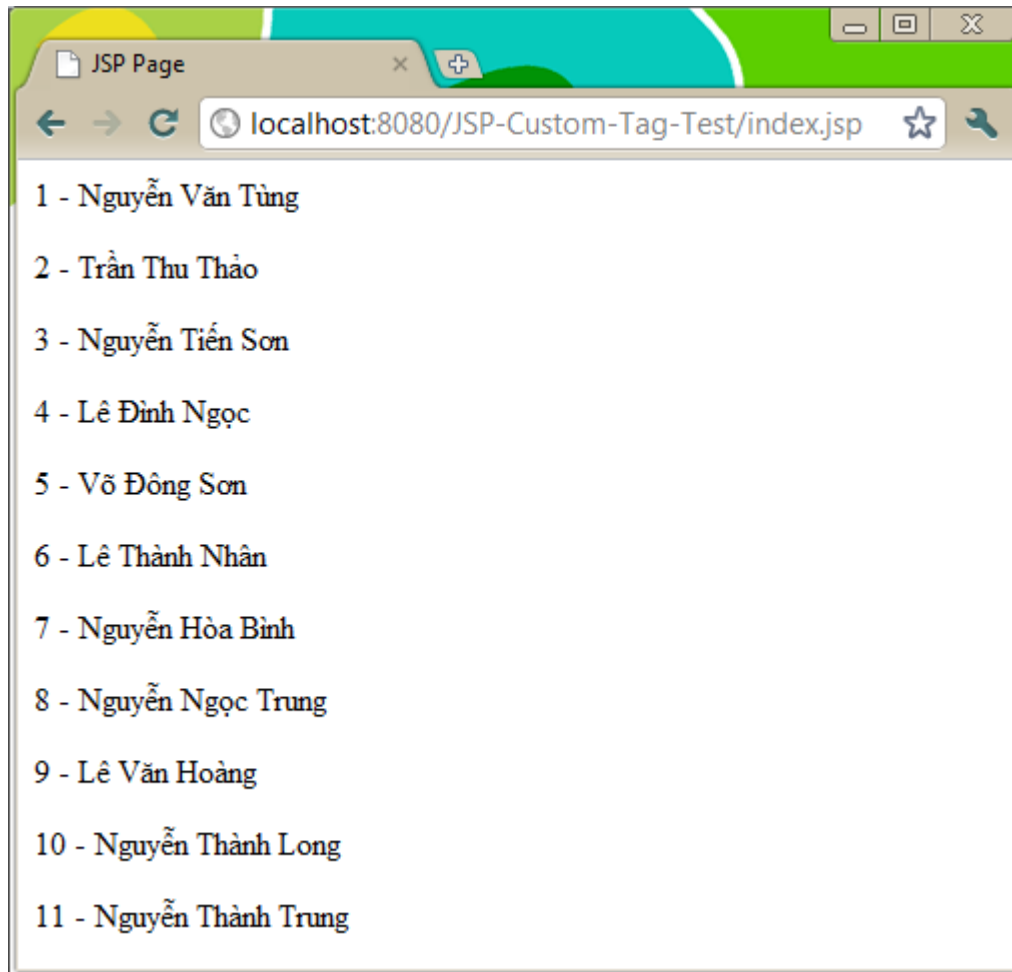
B3: Khai báo và sử dụng trong JSP

```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2   <%@taglib prefix="nhanh"
3       uri="/WEB-INF/tlds/CustomTagLibrary" %>
4 <html>
5   <head>
6     <meta http-equiv="Content-Type"
7         content="text/html; charset=UTF-8">
8     <title>JSP Page</title>
9   </head>
10  <body>
11    . . .
12  </body>
13 </html>
```

B3: Khai báo và sử dụng trong JSP

```
1      <%
2      String []names={"Nguyễn Văn Tùng", "Trần Thu Thảo",
3                      "Nguyễn Tiến Sơn", "Lê Đình Ngọc",
4                      "Võ Đông Sơn", "Lê Thành Nhân",
5                      "Nguyễn Hòa Bình", "Nguyễn Ngọc Trung",
6                      "Lê Văn Hoàng"};
7
8      for(int i=0; i<names.length; i=""+++) {
9          %>
10         <nhanh:EmployeeTag
11             employeeID="<%=String.valueOf(i+1)%>"
12             employeeName="<%=names[i]%>"/>
13
14         <%}%>
15     <nhanh:EmployeeTag employeeID="10"
16         employeeName="Nguyễn Thành Long"/>
17     <nhanh:EmployeeTag employeeID="11"
18         employeeName="Nguyễn Thành Trung"/>
```

B3: Khai báo và sử dụng trong JSP



BODY CUSTOM TAG

Body Custom Tag

- Đặc điểm
 - Có thuộc tính
 - Có thân

- Định dạng

```
<prefix:TagName >
```

```
    JSP Content
```

```
</prefix>
```

```
<prefix:TagName attribute1="..." ... attributeN="...">
```

```
    JSP Content
```

```
</prefix>
```

Các bước xây dựng và sử dụng

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

Bước 1: Tạo lớp Tag Handler Class

- Để tạo được Custom Tag có thuộc tính

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

JSP Content

```
</prefix:TagName>
```

- Trong Tag Handler Class
 - Thêm thuộc tính attribute1, attribute2,...,attributeN
 - Thêm các phương thức:

```
public void setAttribute1(String value1) {...}  
public void setAttribute2(String value2) {...}, ...  
public void setAttributeN(String valueN) {...}
```


Bước 1: Tạo lớp Tag Handler Class

- Trong Tag Handler Class
 - Cài đặt lại doStartTag()
 - Xử lý khi bắt đầu mở Tag
 - Thường return EVAL_BODY_INCLUDE thay cho SKIP_BODY
 - Cài đặt lại doEndTag()
 - Xử lý sau khi kết thúc thân Tag
 - Return EVAL_PAGE

Bước 1: Tạo lớp Tag Handler Class

- Một số phương thức hay dùng với pageContext

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

Bước 1: Tạo lớp Tag Handler Class

- Include và Forward với pageContext

Thao tác	Phương thức
Include	<code>void include(String relativeURLPath)</code>
Forward	<code>void forward(String relativeURLPath)</code>

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     ...
3     @Override
4     public int doStartTag() {
5         //Sử dụng các thuộc tính
6         ...
7         return EVAL_BODY_INCLUDE;
8     }
9     @Override
10    public int doEndTag() {
11        return EVAL_PAGE;
12    }
13    @Override
14    public void release(){
15        this.attribute1=...
16        this.attribute2=...
17        ...
18        this.attributeN=...
19        //Release ...
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>JSP</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>JSP</body-Content>` : Nội dung thân của thẻ là JSP

B3: Khai báo và sử dụng trong JSP

- Khai báo

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

- Sử dụng

```
<prefix:TagName attribute1="..." ... attributeN="...">  
    JSP Content  
</prefix>
```


Bước 1: Tạo lớp Tag Handler Class

```
1  package customs;
2
3  import java.io.IOException;
4  import java.util.logging.Level;
5  import java.util.logging.Logger;
6  import javax.servlet.jsp.JspWriter;
7  import javax.servlet.jsp.tagext.TagSupport;
8
9  /**
10   *
11   * @author NHAnh
12   */
13  public class CopyrightTag extends TagSupport {
14
15      private String bgColor;
16
17      public void setBgColor(String bgColor) {
18          this.bgColor = bgColor;
19      }
20  }
```

Bước 1: Tạo lớp Tag Handler Class

```
1  @Override
2  public int doStartTag() {
3      JspWriter out = this.pageContext.getOut();
4      try {
5          out.println("<table width='600' border='1'");
6          out.println("align='center' bgcolor='" +
7              this.bgColor + "' >");
8          out.println("<tr>");
9          out.println("<td>");
10         out.println("<center>");
11     } catch (IOException ex) {
12         Logger.getLogger(CopyrightTag.class.getName()).
13             log(Level.SEVERE, null, ex);
14     }
15     return EVAL_BODY_INCLUDE;
16 }
```

Bước 1: Tạo lớp Tag Handler Class

```
1  @Override
2  public int doEndTag() {
3      JspWriter out = this.pageContext.getOut();
4      try {
5          out.println("</center>");
6          out.println("</td>");
7          out.println("</tr>");
8          out.println("</table>");
9      } catch (IOException ex) {
10         Logger.getLogger(CopyrightTag.class.getName())
11             .log(Level.SEVERE, null, ex);
12     }
13     return EVAL_PAGE;
14 }
15 }
```

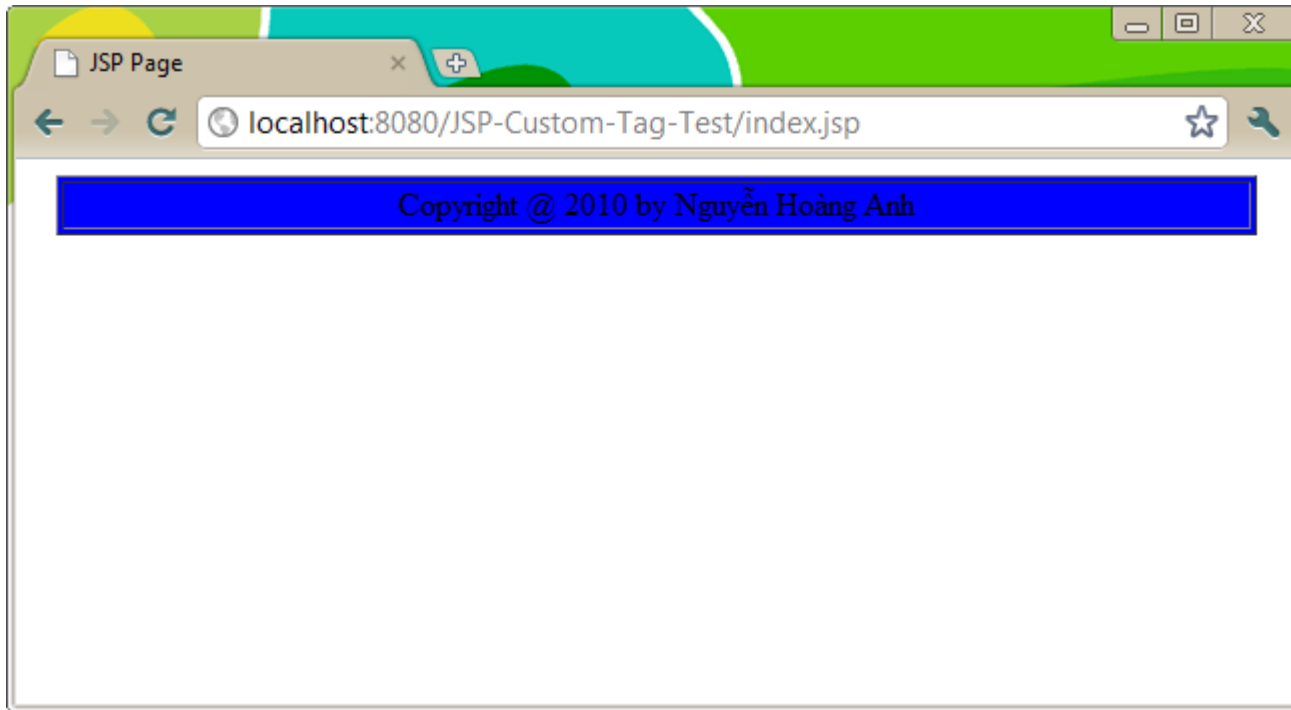
B2: Tạo Tag Library Descriptor (TLD)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <taglib>
3   <tlib-version>1.0</tlib-version>
4   <short-name>customtaglibrary</short-name>
5   <uri>/WEB-INF/tlds/CustomTagLibrary</uri>
6   <tag>
7     <name>CopyrightTag</name>
8     <tag-class>customs.CopyrightTag</tag-class>
9     <attribute>
10      <name>bgColor</name>
11      <required>true</required>
12      <rtexprvalue>true</rtexprvalue>
13    </attribute>
14    <body-content>JSP</body-content>
15  </tag>
16 </taglib>
```

B3: Khai báo và sử dụng trong JSP

```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <%@taglib prefix="nhanh"
3         uri="/WEB-INF/tlds/CustomTagLibrary" %>
4 <html>
5     <head>
6         <meta http-equiv="Content-Type"
7             content="text/html; charset=UTF-8">
8         <title>JSP Page</title>
9     </head>
10    <body>
11        <nhanh:CopyrightTag bgColor="#0000FF">
12            Copyright @ 2010 by Nguyễn Hoàng Anh
13        </nhanh:CopyrightTag>
14    </body>
15 </html>
```

B3: Khai báo và sử dụng trong JSP



NESTED CUSTOM TAG

Bước 1: Tạo lớp Parent Tag Handler Class

```
1 public class ParentTag extends TagSupport{
2     private boolean condition;
3
4     public boolean getCondition() {
5         return condition;
6     }
7     public void setCondition(boolean condition) {
8         this.condition = condition;
9     }
10
11
12     @Override
13     public int doStartTag() {
14         return EVAL_BODY_INCLUDE;
15     }
16
17     @Override
18     public int doEndTag() {
19         return EVAL_PAGE;
20     }
21 }
```


Bước 1: Tạo lớp Child Tag Handler Class

```
1 public class ChildTag extends TagSupport {
2     @Override
3     public int doStartTag() {
4         ParentTag pt = (ParentTag) this.getParent();
5         if (pt.getCondition() == true) {
6             return EVAL_BODY_INCLUDE;
7         } else {
8             return SKIP_BODY;
9         }
10    }
11    @Override
12    public int doEndTag() {
13        return EVAL_PAGE;
14    }
15 }
```

B2: Tạo Tag Library Descriptor (TLD)

```
1  ...
2  <tag>
3      <name>ParentTag</name>
4      <tag-class>customs.ParentTag</tag-class>
5      <attribute>
6          <name>condition</name>
7          <required>true</required>
8          <rtexprvalue>true</rtexprvalue>
9      </attribute>
10     <body-content>JSP</body-content>
11 </tag>
12 <tag>
13     <name>ChildTag</name>
14     <tag-class>customs.ChildTag</tag-class>
15     <body-content>JSP</body-content>
16 </tag>
17 ...
18
19
20
```

B3: Khai báo và sử dụng trong JSP

```
1 <body>
2     <nhanh:ParentTag condition="true">
3         <nhanh:ChildTag>
4             Hi Nguyễn Hoàng Anh
5         </nhanh:ChildTag>
6     </nhanh:ParentTag>
7     <nhanh:ParentTag condition="false">
8         <nhanh:ChildTag>
9             Không xuất hiện
10        </nhanh:ChildTag>
11    </nhanh:ParentTag>
12 </body>
```

SIMPLE IF CUSTOM TAG

Bước 1: Tạo lớp If Tag Handler Class

```
1 package customs;
2 import java.io.IOException;
3 import javax.servlet.jsp.JspException;
4 import javax.servlet.jsp.tagext.SimpleTagSupport;
5 public class IfTag extends SimpleTagSupport {
6     private boolean condition;
7
8     public boolean getCondition() {
9         return condition;
10    }
11
12    public void setCondition(boolean condition) {
13        this.condition = condition;
14    }
15    @Override
16    public void doTag() throws JspException, IOException{
17        this.getJspBody().invoke(null);
18    }
19 }
20 }
```

Bước 1: Tạo lớp Then Tag Handler Class

```
1 package customs;
2 import java.io.IOException;
3 import javax.servlet.jsp.JspException;
4 import javax.servlet.jsp.tagext.SimpleTagSupport;
5 public class ThenTag extends SimpleTagSupport {
6
7     @Override
8     public void doTag() throws JspException, IOException {
9         IfTag it=(IfTag) this.getParent();
10        if(it.getCondition()==true) {
11            this.getJspBody().invoke(null);
12        }
13    }
14 }
15
16
17
18
19
20
```

Bước 1: Tạo lớp Else Tag Handler Class

```
1 package customs;
2
3 import java.io.IOException;
4 import javax.servlet.jsp.JspException;
5 import javax.servlet.jsp.tagext.SimpleTagSupport;
6
7 public class ElseTag extends SimpleTagSupport {
8
9     @Override
10     public void doTag() throws JspException, IOException {
11         IfTag it = (IfTag) this.getParent();
12         if (it.getCondition() == false) {
13             this.getJspBody().invoke(null);
14         }
15     }
16 }
17
18 }
```

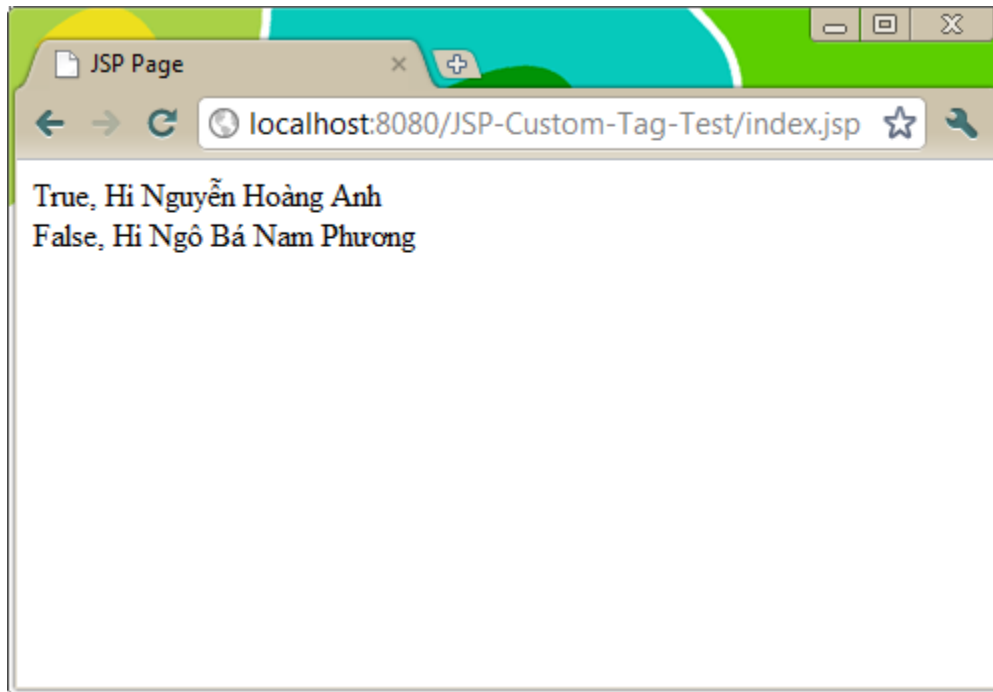
Bước 2: Tạo Tag Library Descriptor (TLD)

```
1 <tag>
2     <name>if</name>
3     <tag-class>customs.IfTag</tag-class>
4     <attribute>
5         <name>condition</name>
6         <required>true</required>
7         <rtexprvalue>true</rtexprvalue>
8     </attribute>
9     <body-content>scriptless</body-content>
10 </tag>
11 <tag>
12     <name>then</name>
13     <tag-class>customs.ThenTag</tag-class>
14     <body-content>scriptless</body-content>
15 </tag>
16 <tag>
17     <name>else</name>
18     <tag-class>customs.ElseTag</tag-class>
19     <body-content>scriptless</body-content>
20 </tag>
```


Bước 3: Khai báo và sử dụng trong JSP

```
1 <body>
2   <nhanh:if condition="true">
3     <nhanh:then>
4       True, Hi Nguyễn Hoàng Anh
5     </nhanh:then>
6     <nhanh:else>
7       False, Hi Ngô Bá Nam Phương
8     </nhanh:else>
9   </nhanh:if>
10  <br/>
11  <nhanh:if condition="false">
12    <nhanh:then>
13      True, Hi Nguyễn Hoàng Anh
14    </nhanh:then>
15    <nhanh:else>
16      False, Hi Ngô Bá Nam Phương
17    </nhanh:else>
18  </nhanh:if>
19  </body>
```

Bước 3: Khai báo và sử dụng trong JSP



SIMPLE FOR CUSTOM TAG

Bước 1: Tạo lớp For Tag Handler Class

```
1 package customs;
2 import java.io.IOException;
3 import javax.servlet.jsp.JspException;
4 import javax.servlet.jsp.tagext.SimpleTagSupport;
5 public class ForTag extends SimpleTagSupport{
6     private int size;
7     public void setSize(int size) {
8         this.size = size;
9     }
10    @Override
11    public void doTag() throws JspException, IOException{
12        for(int i=0; i<this.size; i++){
13            this.getJspBody().invoke(null);
14        }
15    }
16 }
```

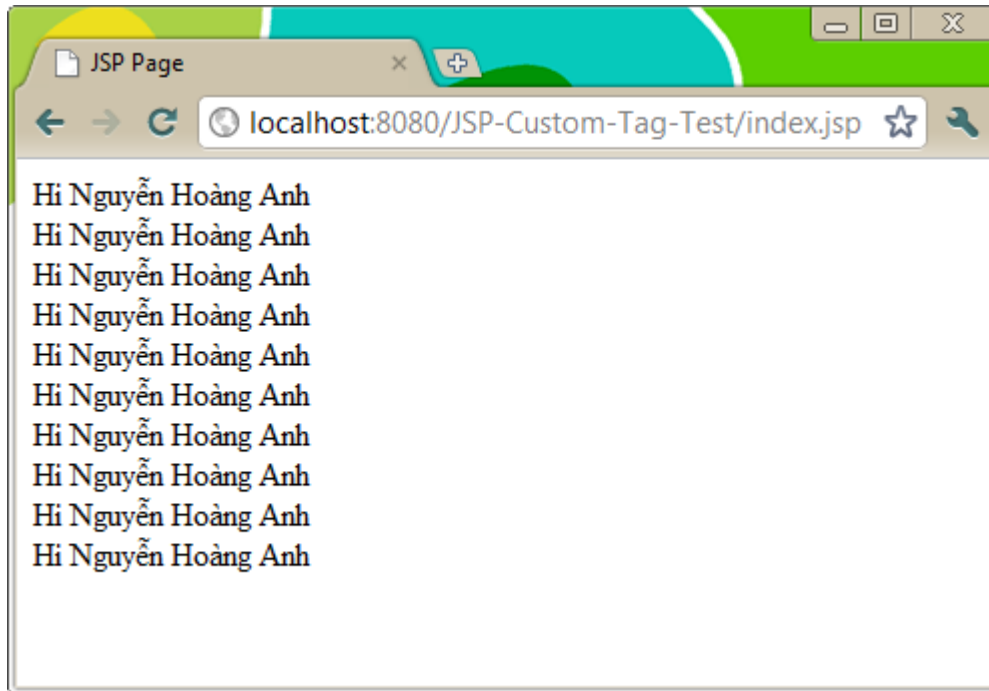
Bước 2: Tạo Tag Library Descriptor (TLD)

```
1 <tag>
2   <name>for</name>
3   <tag-class>customs.ForTag</tag-class>
4   <attribute>
5     <name>size</name>
6     <required>true</required>
7     <rtexprvalue>true</rtexprvalue>
8   </attribute>
9   <body-content>scriptless</body-content>
10
11 </tag>
```

Bước 3: Khai báo và sử dụng trong JSP

```
1 <body>
2     <nhanh:for size="10">
3         <nhanh:if condition="true">
4             <nhanh:then>
5                 Hi Nguyễn Hoàng Anh
6             </nhanh:then>
7             <nhanh:else>
8                 Hi Ngô Bá Nam Phương
9             </nhanh:else>
10        </nhanh:if>
11        <br/>
12    </nhanh:for>
13 </body>
```

Bước 3: Khai báo và sử dụng trong JSP



TÀI LIỆU THAM KHẢO

Tham khảo

- Marty Hall, Larry Brown, Core Servlets and JavaServer Pages™: Volume 1: Core Technologies, 2nd Edition (2003)
- <http://courses.coreservlets.com/Course-Materials/csajsp2.html> (2010)



HỎI VÀ ĐÁP