BÀI 6:
Quốc tế hóa trang JSP & CustomTag

Mục tiêu bài học

- Sử dụng thư viện FMT trong JSTL để cài đặt I18N
- Làm việc với custom tag trong JSP

Giới thiệu về FMT Tag Library

- Formatting tags are used to format and display text, numbers, date, or time in a specific format.
- Some of the formatting tags include:
  - `<formatNumber>`
  - `<parseNumber>`
  - `<formatDate>`
  - `<timeZone>`
  - `<setLocale>`

Giới thiệu về FMT Tag Library

- Thư viện thẻ định dạng (FMT Tag Lib) cung cấp các thẻ cho phép định dạng các thông điệp, cũng như định dạng các giá trị chuẩn như: số, ngày tháng, phần tram v.v...
- Url của thư viện fmt là **http://java.sun.com/jsp/jstl/fmt** và prefix là **fmt.**
- Thư viện fmt còn cho phép quốc tế hóa (i18n) và bản địa hóa (l10n) các ứng dụng web, để cho phép ứng dụng web có thể hiển thị tốt trên nhiều ngôn ngữ khác nhau.

Giới thiệu về FMT Tag Library

- Để sử dụng thư viện fmt, ta cần chèn câu khai báo directive sau:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"  %>
```

- Sau đó, ta có thể gọi ra các thẻ bên trong thư viện fmt.

Giới thiệu về FMT Tag Library

- Bảng bên dưới chứa mô tả ý nghĩa của các thẻ trong thư viện fmt:

Tên thẻ	Ý nghĩa
fmt:parseNumber	Được dùng để parse chuỗi theo một trong các định dạng: tiền tệ, số hoặc phần trăm
fmt:timeZone	Cho phép parse time zone theo định dạng
fmt:formatNumber	Cho phép định dạng giá trị số theo một định dạng hoặc độ chính xác thập phân
fmt:parseDate	Cho phép parse một chuỗi thể hiện một giá trị ngày tháng
fmt:bundle	Dùng để tạo ra đối tượng ResourceBundle cho phép lấy thông tin trong file cấu hình
fmt:setTimeZone	Dùng để thiết lập time zone
fmt:setBundle	Cho phép load file resource bundle cho file
fmt:message	Cho phép lấy ra một giá trị của message trong file cấu hình theo key
fmt:formatDate	Cho phép định dạng giá trị ngày tháng/thời gian theo một style/pattern nào đó (VD: dd/mm/yyyy)

Minh họa

- Demo và thực hành quốc tế hóa ứng dụng web với thư viện FMT trong JSTL

Sử dụng Custom Tags trong JSP

- JSTL cung cấp một số lượng lớn các thẻ mà người dùng có thể sử dụng trong các trang JSP để thực hiện nhiều tác vụ khác nhau.
- Tuy nhiên, nhiều khi các thẻ trong thư viện JSTL không thể thực thi được tất cả những nhiệm vụ theo nhu cầu của người dùng.
- Trong những tình huống như vậy, ta cần sử dụng các thẻ tùy biến (custom tags) để có thể đáp ứng được các nhu cầu chuyên biệt của từng dự án, từng người dùng.
- Custom tags là những thành phần có thể sử dụng lại (reusable components) cung cấp những lợi ích sau:
  - Giảm thiểu việc sử dụng các đoạn scriptlets trong trang JSP
  - Có tính tái sử dụng cao

Sử dụng Custom Tags trong JSP

- Custom tags có thể được phân loại thành những loại sau:
  - Các thẻ rỗng, không có thân (Empty tags)
  - Các thẻ có thuộc tính (Tags with attributes)
  - Các thẻ có nội dung (Tags with a body)
- Custom tags thường được định nghĩa dưới dạng các thư viện thẻ, và có một file XML được sử dụng để cấu hình thông tin của từng thẻ trong thư viện.

Sử dụng Custom Tags trong JSP

- Để tạo ra các custom tags, ta cần thực hiện các tác vụ sau:
  - Tạo ra một hay nhiều các tag handler.
  - Tạo ra một file cấu hình - Tag Library Descriptor (TLD).
- Tag handler:
  - Là một lớp Java được sử dụng bởi Web container. Tag handler sẽ cung cấp nghiệp vụ cho custom tag để xử lý yêu cầu của người dùng.
  - Các phương thức của tag handler sẽ được kế thừa từ gói `javax.servlet.jsp.tagext`.
  - Cài đặt các interfaces `Tag`, `SimpleTag`, hoặc `BodyTag`.

Sử dụng Custom Tags trong JSP

- Dưới đây là bảng mô tả ý nghĩa của các lớp trong gói `javax.servlet.jsp.tagext` package.

Class	Description
`BodyContent`	*Là lớp con của lớp `JSPWriter` và thể hiện nội dung của một custom tag.*
`TagSupport`	*Hoạt động như một base class cho các tag handlers và cài đặt empty tags*
`BodyTagSupport`	*Cài đặt interface `BodyTag` và được dùng để phát triển các custom tags có nội dung (body)*
`SimpleTagSupport`	*Là một base class cho các tag handlers mà cài đặt intercace `SimpleTag`*
`TagData`	*Thể hiện các thuộc tính và giá trị của chúng.*
`TagInfo`	*Thể hiện thông tin được khai báo trong thẻ `<tag></tag>` là một thành phần của file TLD và được sử dụng bởi JSP engine trong quá trình chuyển đổi một trang JSP thành một servlet*
`TagLibraryInfo`	*Thể hiện thông tin mô tả của file TLD, VD như thông tin mô tả về các tags, hoặc thông tin về phiên bản.*
`TagVariableInfo`	*Thể hiện thông tin về các biến của một custom tag*

Sử dụng Custom Tags trong JSP

- Để tạo ra một tag handler, ta cần định nghĩa chức năng của một custom tag bằng cách sử dụng các phương thức sau của `Tag` interface:
  - `doStartTag()`: Được gọi khi Web container bắt gặp thẻ mở (start tag) của custom tag.
  - `doEndTag()`: Được gọi khi Web container bắt gặp thẻ đóng (end tag) của custom tag.
  - `release()`: Được gọi để giải phóng thể hiện của tag handler.

Sử dụng Custom Tags trong JSP

- Các phương thức được cài đặt bên trong tag handler sẽ phụ thuộc vào cấu trúc của tag.
- The following table lists some of the methods that need to be implemented according to the structure of the tag handler.

Cấu trúc của Tag Handler	***Các phương thức được cài đặt***
Simple tag không có thân và không có thuộc tính	`doStartTag`, `doEndtag`, và `release`
Các thẻ có thuộc tính	`doStartTag`, `doEndtag`, và các cặp phương thức `setAttribute` /`getAttribute`
Các thẻ có thân	`doStartTag`, `doEndTag`, `release`, `doInitBody`, `doAfterBody`
Các thẻ cài đặt `SimpleTag` *interface*	`doTag()`

Cấu hình Custom Tags

- File TLD: Là một file XML chứa danh sách và cấu hình thông tin mô tả của tất cả các custom tags trong thư viện custom tag.
- File TLD sẽ dùng để cấu hình cho thư viện thẻ, và tất cả các thẻ trong thư viện. Web container sẽ đọc thông tin trong file này để lấy ra thông tin mô tả về thẻ, và để sử dụng thẻ.
- Các thành phần của file TLD có thể được phân loại thành 2 nhóm.
  - Nhóm thứ nhất bao gồm các phần tử là một phần của thẻ gốc trong file TLD, là thẻ `<taglib>`. Nhóm này dùng để cấu hình thông tin mô tả về thư viện thẻ
  - Nhóm thứ hai, được đặt bên trong thẻ `<taglib>`, bao gồm các phần tử là thẻ con của thẻ `<tag>`. Nhóm này dùng để cấu hình thông tin của từng thẻ trong thư viện.

Cấu hình Custom Tags

- Bảng bên dưới mô tả về ý nghĩa của từng thẻ con của thẻ `<taglib>`.

Sub Tag	Ý nghĩa của thẻ
`<tlib-version>`	*Khai báo phiên bản của thư viện thẻ, VD:* `<tlib-version>1.0</tlib-version>`
`<jsp-version>`	*Khai báo phiên bản của version of JSP đang triển khai và sử dụng thư viện thẻ, VD:* `<jsp-version>1.2</jsp-version>`
`<short-name>`	*Tên của thư viện thẻ*
`<uri>`	*Khai báo URI của thư viện thẻ*
`<tag>`	*Dùng để khai báo từng thẻ trong thư viện*

Cấu hình Custom Tags

- Bảng bên dưới mô tả ý nghĩa của các thẻ con của thẻ `<tag>`.

Sub Tag	***Ý nghĩa***
`<name>`	*Tên của custom tag*
`<tag-class>`	*Tên class của tag handler của một custom tag*
`<body-content>`	*Thiết lập định dạng của body content của một custom tag*

Cấu hình Custom Tags

- Đoạn code dưới đây minh họa một file TLD:

```
<taglib version="2.1"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-
jsptaglibrary_2_1.xsd">
  <tlib-version>1.0</tlib-version>
  <jsp-version>2.0</jsp-version>
  <short-name>tld1</short-name>
  <uri>/WEB-INF/tlds/tld1</uri>
  <tag>
      <name>TestTag</name>
      <tag-class>FirstServlet</tag-class>
  </tag>
</taglib>
```

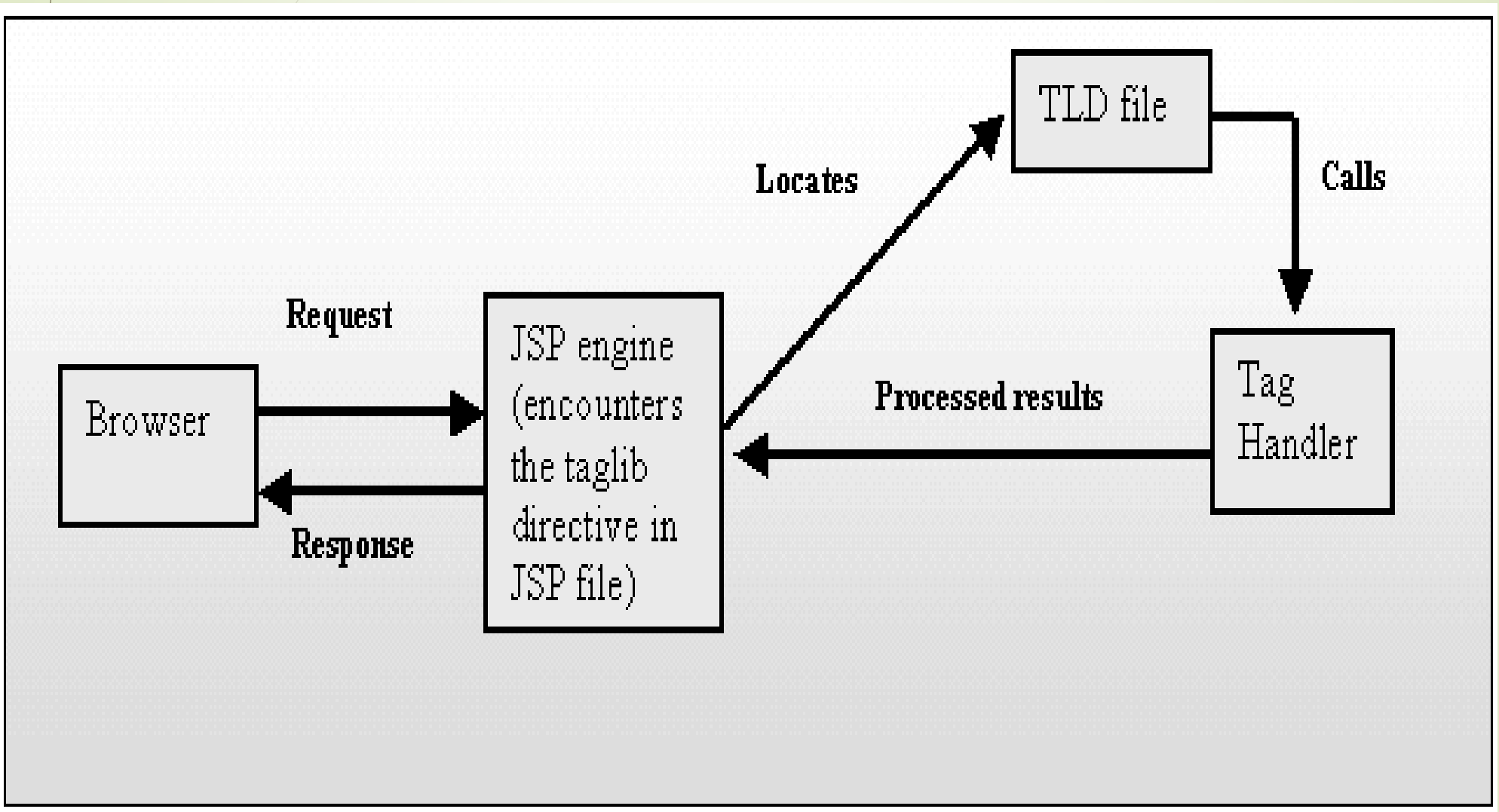
Sử dụng và gọi Custom Tags

- Để sử dụng các custom tags trong trang JSP, ta cần thực hiện các bước như sau:
  1. Đặt file TLD vào trong thư mục WEB-INF nằm bên trong ứng dụng Web.
  2. Đặt file class của tag handler vào trong thư mục WEB-INF/classes của ứng dụng Web.
  3. Chèn khai báo `taglib` directive vào trang JSP như dòng lệnh sau:

     ```
     <%@ taglib prefix="ct" uri="/WEB-INF/tld1.tld"%>
     ```
  4. Sử dụng custom tag tại vị trí mong muốn trong trang JSP, như dòng lệnh bên dưới:

     ```
     <ct:TestTag/>
     ```

Vòng đời của Custom Tags



Sử dụng Custom Tags

- Tạo một file Tag Library Descriptor (.tld) file bên trong thư mục tlds folder nằm trong thư mục WEB-INF của ứng dụng Web.
- Tạo một file Tag Handler tương ứng với mỗi custom tag trong thư viện.
- Chèn khai báo taglib directive trong trang JSP
- Sử dụng custom tag (chú ý về prefix)

Sử dụng Custom Tags

- Giá trị trả về của phương thức doStartTag()
  - SKIP_BODY
  - EVAL_BODY_INCLUDE
  - EVAL_BODY_BUFFERED
- Giá trị trả về của phương thức doEndTag()
  - SKIP_PAGE
  - EVAL_PAGE
- Giá trị trả về của phương thức doAfterBody()
  - SKIP_BODY
  - EVAL_BODY_AGAIN

Minh họa

- Demo và thực hành sử dụng custom tag

Tổng kết

- Sử dụng thư viện FMT trong JSTL để cài đặt I18N
- Làm việc với custom tag trong JSP