

## **Hướng dẫn sử dụng thư viện JSTL core trong JSP**

### **Falcuty: Nguyễn Hùng Cường**

JSTL: JSP Standard Tag Libraries: Là một thư viện chuẩn của JSP, bao gồm một số các thẻ chuẩn để xử lý những tác vụ phổ biến nhất trong JSP.

JSTL bao gồm 4 thư viện con sau:

Core: Là thư viện chứa các thẻ cơ sở, dùng để thực hiện các thao tác như khai báo/hiển thị biến, thực hiện những câu lệnh if, switch, các vòng lặp v.v...

SQL: Là thư viện con của JSTL chứa các thẻ cho phép thao tác với CSDL, VD như mở kết nối, tạo recordset, insert/update/delete/select, transaction, param v.v...

FMT (Formatting and I18N): Là thư viện con của JSTL chứa các thẻ cho phép định dạng ngày tháng, số, phần trăm và thực hiện quốc tế hóa ứng dụng web.

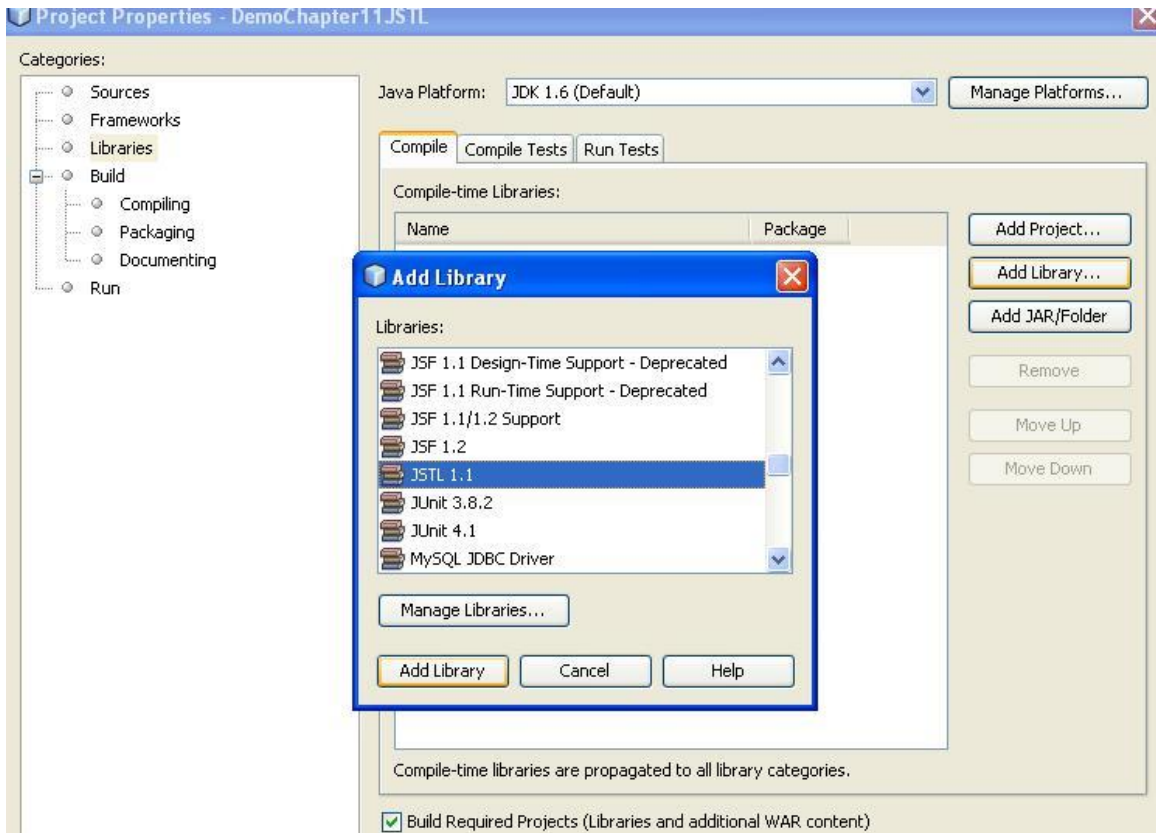
XML: Là thư viện con của JSTL cho phép xử lý và thao tác với các file XML.

#### **Bước 1: Tạo Java Web Application**

Mở NetBeans IDE, tạo mới một web application, sau đó tạo các trang JSP.

#### **Bước 2: Add JSTL Library**

Click phải chuột lên Project, chọn mục Properties, chọn Libraries, chọn nút Add Library để add thư viện JSTL vào project. Chú ý là phải add JSTL Library vào Project thì mới sử dụng được JSTL.



### Bước 3: Import thư viện JSTL

Chú ý: Muốn sử dụng các thẻ trong JSTL ta phải import các thư viện này vào trang JSP bằng cách sử dụng directive `<%@taglib>`, directive này gồm 2 thuộc tính sau:  
 prefix: Chỉ ra tiền tố cho taglib, tiền tố này có thể được đặt tên tùy ý.  
 uri: URI của mỗi taglib.

```
<%@ taglib uri="http://java.sun.com/jstl/core_rt" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jstl/sql_rt" prefix="sql" %>
```

### Bước 4: Sử dụng các thẻ Core Tag Library trong thư viện của JSTL

`<c:set>`: Khai báo một biến  
`<c:out>`: In ra giá trị của một biến  
`<c:remove>`: Xóa một biến

```
<!-- Khai bao mot bien su dung JSTL -->
<c:set var="tony" value="demo" scope="page" />

<!-- in ra gia tri cua bien -->
Tên biến là: <c:out value="${pageScope.tony}" />
```

<c:if>: Kiểm tra một điều kiện logic (câu lệnh if)  
Thuộc tính test: Dùng để khai báo điều kiện của khối if.

```
<c:if test="${pageContext.request.method=='POST'}">
```

<c:choose>: Thể hiện một câu lệnh switch-case

<c:when>: Thể hiện một case trong câu lệnh switch-case.

Thuộc tính test: Dùng để khai báo điều kiện của thẻ <c:when>.

<c:otherwise>: Thể hiện khối default trong câu lệnh switch-case.

```
<c:choose>
```

```
<c:when test="${param.enter=='1'}">Bánh cuốn.
```

```
<br />
```

```
</c:when>
```

```
<c:when test="${param.enter=='2'}">Phở gà 2 trứng.
```

```
<br />
```

```
</c:when>
```

```
<c:when test="${param.enter=='3'}">Mỳ vằn thắn.
```

```
<br />
```

```
</c:when>
```

```
<c:otherwise>Cô gái chân dài
```

```
<br />
```

```
</c:otherwise>
```

```
</c:choose>
```

<c:forEach>: Thể hiện một vòng lặp for

Ý nghĩa của các thuộc tính:

var: Khai báo tên biến đếm.

begin: Khai báo giá trị bắt đầu được duyệt.

end: Khai báo giá trị cuối cùng được duyệt.

step: Khai báo mỗi lần lặp sẽ thay đổi giá trị biến đếm bao nhiêu.

Dưới đây là VD ta thực hiện một vòng lặp duyệt qua các giá trị từ 1-10, biến đếm tên là i mỗi lần biến đếm sẽ nhảy qua 2 giá trị.

```
<!-- thực hiện một vòng lặp for -->
```

```
<tonyhungcuong:forEach var="i" begin="1" end="10" step="2">
```

```
<tonyhungcuong:out value="${i}" />
```

```
<br />
```

```
</tonyhungcuong:forEach>
```

<c:forTokens>: Thể hiện một vòng lặp cho phép phân tích một giá trị thành các token

```
<tonyhungcuong:set var="monphai" value="Thiếu Lâm:Dương Gia;Võ Đang|Nga My$Đường Môn$Ngũ Độc"
scope="page" />
```

```
|
<!-- minh hoa ve for tokens -->
```

```
<tonyhungcuong:forTokens items="{pageScope.monphai}"
    delims=".:|.$"
    var="vitieubao">
```

Tên môn phái là

```
<tonyhungcuong:out value="{vitieubao}" /> <br />
```

```
</tonyhungcuong:forTokens>
```