Mục lục

[Mục lục 1](#_Toc24898833)

[Chương 1. JSTL 2](#_Toc24898834)

[1.1 Tổng quan 2](#_Toc24898835)

[1.2 Core Tags 3](#_Toc24898836)

[1.3 Formatting Tags 6](#_Toc24898837)

# Tags-Tiles

## Tổng quan

Đây có thể coi là phần nối tiếp cho phần [3.6.4] khi phần đó trình bày các thao tác cơ bản để làm việc với Apache Tiles như cách tạo ra layout, khai báo và trình bày các view theo các layout xác định…

Thư viện hỗ trợ cho tablib này là:

<dependency>

<groupId>javax.servlet.jsp.jstl</groupId>

<artifactId>javax.servlet.jsp.jstl-api</artifactId>

<version>1.2.1</version>

</dependency>

Khi cấu trúc layout là phức tạp hơn, chúng ta cần những thao tác uyển chuyển và linh hoạt hơn. Phần này sẽ trình bày về tập các tag mà Apache Tiles cung cấp để làm việc với framework này, kèm theo ví dụ.

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE tiles-definitions PUBLIC

"-//Apache Software Foundation//DTD Tiles Configuration 3.0//EN"

"http://tiles.apache.org/dtds/tiles-config\_3\_0.dtd">

<tiles-definitions>

…

<definition> … </definition>

…

</tiles-definitions>

Phần mã phía trên là cấu trúc của một file dùng để thiết lập các definition. Trong đồ án này, nó có tên là tiles.xml. Tất nhiên là có thể đặt những tên khác và tùy theo mục đích mà có thể tạo ra một hoặc nhiều file.

### <put-attribute />

Thẻ chứa nó là <definition />. Nó định nghĩa giá trị mà chúng ta thêm vào tại một vị trí có “name” đã được xác định trong template hoặc một definition khác. Giá trị có thể là chuỗi, template hoặc một definition khác.

<definition name=*"blank"* template=*"/WEB-INF/views/layout/layout.jsp"*/>

<definition name=*"index"* extends=*"blank"*>

<put-attribute name=*"title"* value=*"STU - YTBhoanghai"*/>

<put-attribute name=*"header"*

value=*"/WEB-INF/views/jsp/header.jsp"*/>

<put-attribute name=*"body"* value=*"/WEB-INF/views/jsp/body.jsp"*/>

<put-attribute name=*"footer"*

value=*"/WEB-INF/views/jsp/footer.jsp"*/>

</definition>

Đây là một ví dụ trong phần demo đồ án, chúng ta có thể thấy definition “index” kế thừa definition “blank”, thực hiện đẩy các giá trị nằm trong thuộc tính value vào vị trí “name” đã được xác định trong file layout.jsp. Như đã đề cập, các giá trị nhận vào sẽ được tự động chuyển thành kiểu tương ứng được quy định trong file layout.jsp.

### <add-attribute />

Tương tự như tag <put-addtribute /> nhưng nó được sử dụng trong tag <put-list-attribute /> sẽ được trình bày bên dưới.

### <put-list-attribute />

Trong một số trường hợp chúng ta cần một tập các thuộc tính để hiển thị thay vì thực hiện xác định cụ thể tên và vị trí, chúng ta sẽ dùng tag <put-list-attribute />. Để hiểu rõ hơn về chức năng của nó, sau đây là một ví dụ.

<definition name=*"blank"* template=*"/layout/layoutExample.jsp"*>

<put-list-attribute name=*"items"*>

<add-attribute value=*"/tiles/header.jsp"* />

<add-attribute value=*"/tiles/footer.jsp"* />

</put-list-attribute>

</definition>

2 attribute sẽ được gói trong đối tượng items có kiểu java.util.List và sẽ được gửi ra ngoài layoutExample.jsp. Để nhận và nạp các giá trị này, file layoutExample.jsp sẽ có nội dung như sau:

<%@taglib uri=*"http://tiles.apache.org/tags-tiles"* prefix=*"tiles"* %>

<%@taglib uri=*"* *http://java.sun.com/jsp/jstl/core"* prefix=*"c"* %>

<tiles:useAttribute id=*"list"* name=*"items"*

classname=*"java.util.List"* />

<c:forEach var=*"item"* items=*"*${list}*"*>

<tiles:insertAttribute value=*"*${item}*"* flush=*"true"* /><br>

</c:forEach>

Chúng ta tiếp nhận đối tượng items được gửi ra bằng tag <tiles:useAttribute /> đặt cho nó 1 id là “list”. Từ đây dùng vòng lặp forEach trong taglib core jstl để lặp và thực hiện insert lần lượt các thuộc tính trong “list” vào trang jsp.

### <tiles:insertAttribute />

Nó thực hiện chèn giá trị của một attribute vào trang jsp. Ví dụ:

<tiles:insertAttribute name=*"top-header"* />

Khi có một attribute được đẩy ra ngoài trang jsp theo bất cứ cách nào <put-attribute /> hoặc <add-attribute />, tag này sẽ bắt lấy các attribute có “name” trùng với “name” của nó và hiển thị.

### <definition />

Tạo một definition tại thời điểm Runtime. Definition được tạo sẽ có mặt sẵn trên toàn bộ các request.

<definition name=*"blank"* template=*"/WEB-INF/views/layout/layout.jsp"*/>

Hoặc

<definition name=*"index"* extends=*"blank"*>

### <tiles:insertDefinition />

Nó được sử dụng tại trang jsp để chèn một definition đã được thiết lập trong file xml.

<tiles:insertDefinition name=*"example"* flush=*"true"*>

<tiles:putAttribute name=*"title"* value=*"My first page"* />

<tiles:putAttribute name=*"header"* value=*"/views/jsp/header.jsp"* />

<tiles:putAttribute name=*"menu"* value=*"/views/jsp/menu.jsp"* />

<tiles:putAttribute name=*"body"* value=*"/views/jsp/body.jsp"* />

<tiles:putAttribute name=*"footer"* value=*"/views/jsp/footer.jsp"* />

</tiles:insertDefinition>

Nó gửi các value của attribute qua tag <tiles:putAttribute /> đến definition đã được thiết lập trong file xml để tạo ra một definition hoàn chỉnh và chèn nó tại vị trí nơi mà tag <tiles:insertDefinition /> đã gọi nó.

### <tiles:insertTemplate />

Tương tự như các thức mà tag <tiles:insertDefinition /> hoạt động, sự khác biệt ở chỗ là nó chèn 1 template chứ không phải 1 definition.

<tiles:insertTemplate template=*"view/layout/example.jsp"* flush=*"true"*>

<tiles:putAttribute name=*"title"* value=*"My first page"* />

<tiles:putAttribute name=*"header"* value=*"/views/jsp/header.jsp"* />

<tiles:putAttribute name=*"footer"* value=*"/views/jsp/footer.jsp"* />

</tiles:insertTemplate>

Đôi khi có sự nhầm lẫn giữa template và definition. Một template sẽ định nghĩa bố cục, thiết lập các vị trí để chèn các thuộc tính và là 1 trang jsp. Definition thực hiện template đó, nó xác định các giá trị của từng thuộc tính sẽ được truyền vào template.

Một definition có thể extends một definition khác. Khi đó, nó kế thừa tất cả các thuộc tính mà definition cha đã thiết lập cho template của nó. Và nó có quyền ghi đè hoặc không ghi đè các thuộc tính được quy định trong template mà definition cha đã thực hiện.

### <tiles:getAsString />

<tiles:getAsString name=*"title"* defaultValue=*"STU | HoangHai"*/>

Lấy các giá trị của thuộc tính được chỉ định và hiển thị nó cho trình ghi hiện tại dưới dạng chuỗi. Như ví dụ trên, nếu không có thuộc tính nào quy định giá trị thì toàn bộ tag trên được ghi ra có dạng “STU | HoangHai”. Ngược lại,

<put-attribute name=*"title"* value=*"STU - YTBhoanghai"*/>

Một definition nào đó định nghĩa value là “STU – YTBhoanghai” thì giá trị được ghi ra lại là “STU – YTBhoanghai”

### <tiles:importAttribute />

Chúng ta biết rằng 1 attribute thì không có scope, nó chỉ được insert một lần và không thể được sử dụng lại. Chúng ta dùng tag <tiles:importAttribute /> để import một thuộc tính có scope được quy định. Trong scope đó, ta có thể dùng lại attribute đó nếu thấy cần thiết.

<tiles:importAttribute name=*"abc"* scope=*"session"*/>

<c:out value=*"*${abc}*"*/>

Khi một definition nào đó đẩy ra một attribute có “name” là abc, tag này sẽ đặt giá trị của attribute đó vào 1 scope và có thể được sử dụng lại nhiều lần.