**freemarker使用手册**

目录

[一、 引言 1](#_Toc307925828)

[1. 编写目的 1](#_Toc307925829)

[2. 术语定义 1](#_Toc307925830)

[3. 概述 1](#_Toc307925831)

[二、 下载安装 2](#_Toc307925832)

[三、 freemarker使用入门 2](#_Toc307925833)

[（1） 模板 + 数据模型 = 输出 2](#_Toc307925834)

[（2） 数据模型 3](#_Toc307925835)

[（3） 模板 5](#_Toc307925836)

[四、 Demo 7](#_Toc307925837)

# 引言

## 编写目的

本文档是freemark使用说明，编写目的是：为即将使用freemark的开发人员提供帮助。

## 术语定义

暂无。

## 概述

FreeMarker允许Java servlet保持图形设计同[应用程序](http://baike.baidu.com/view/330120.htm" \t "_blank)逻辑的分离，这是通过在模板中密封HTML完成的。模板用servlet提供的数据动态地生成 HTML。模板语言是强大的直观的，[编译器](http://baike.baidu.com/view/487018.htm" \t "_blank)速度快，输出接近静态HTML页面的速度。

　　FreeMarker是一个模板引擎，一个基于模板生成文本输出的通用工具，使用纯Java编写

　　FreeMarker被设计用来生成HTML Web页面，特别是基于MVC模式的应用程序

　　虽然FreeMarker具有一些编程的能力，但通常由Java程序准备要显示的数据，由FreeMarker生成页面，通过模板显示准备的数据

　　FreeMarker不是一个Web应用框架，而适合作为Web应用框架一个组件

　　FreeMarker与容器无关，因为它并不知道HTTP或Servlet；FreeMarker同样可以应用于非Web应用程序环境

　　FreeMarker更适合作为Model2框架（如Struts）的视图组件，你也可以在模板中使用JSP标记库

FreeMarker是免费的下载和安装

# 下载安装

首先从官方网站下载freemarker。

下载地址为http://freemarker.sourceforge.net/

下载以后把freemarker.jar包拷贝到lib目录下即可

# freemarker使用入门

## 模板 + 数据模型 = 输出

FreeMarker基于设计者和程序员是具有不同专业技能的不同个体的观念。他们是分工劳动的：设计者专注于表示——创建HTML文件、图片、Web页面的其它可视化方面；程序员创建系统，生成设计页面要显示的数据。经常会遇到的问题是：在Web页面（或其它类型的文档）中显示的信息在设计页面时是无效的，是基于动态数据的。在这里，你可以在HTML（或其它要输出的文本）中加入一些特定指令，FreeMarker会在输出页面给最终用户时，用适当的数据替代这些代码。

下面是一个例子：

<html>

<head>

<title>Welcome!</title>

</head>

<body>

<h1>Welcome ${user}!</h1>

<p>Our latest product:

<a href="${latestProduct.url}">${latestProduct.name}</a>!

</body>

</html>

这个例子是在简单的HTML中加入了一些由${…}包围的特定代码，这些特定代码是FreeMarker的指令，而包含FreeMarker的指令的文件就称为模板（Template）。至于user、latestProduct.url和latestProduct.name来自于数据模型（data model）。数据模型由程序员编程来创建，向模板提供变化的信息，这些信息来自于数据库、文件，甚至于在程序中直接生成。模板设计者不关心数据从那儿来，只知道使用已经建立的数据模型。

下面是一个可能的数据模型：

(root)

+- user = "Big Joe"

+- latestProduct

+- url = "products/greenmouse.html"

+- name = "green mouse"

数据模型类似于计算机的文件系统，latestProduct可以看作是目录，而user、url和name看作是文件，url和name文件位于latestProduct目录中（这只是一个比喻，实际并不存在）

当FreeMarker将上面的数据模型合并到模板中，就创建了下面的输出：

<html>

<head>

<title>Welcome!</title>

</head>

<body>

<h1>Welcome Big Joe!</h1>

<p>Our latest product:

<a href="products/greenmouse.html">green mouse</a>!

</body>

</html>

* 数据模型

典型的数据模型是树型结构，可以任意复杂和深层次，如下面的例子：

(root)

+- animals

+- mouse

+- size = "small"

+- price = 50

+- elephant

+- size = "large"

+- price = 5000

+- python

+- size = "medium"

+- test = "It is a test"

+- whatnot

because = "don't know"

类似于目录的变量称为hashes，包含保存下级变量的唯一的查询名字，

类似于文件的变量称为scalars，保存单值,scalars保存的值有两种类型：字符串（用引号括起，可以是单引号或双引号）和数字（不要用引号将数字括起，这会作为字符串处理），对scalars的访问从root开始，各部分用“.”分隔，如animals.mouse.price。

另外一种变量是sequences，和hashes类似，只是不使用变量名字，而使用数字索引，如下面的例子：

(root)

+- animals

+- (1st)

+- name = "mouse"

+- size = "small"

+- price = 50

+- (2nd)

+- name = "elephant"

+- size = "large"

+- price = 5000

+- (3rd)

+- name = "python"

+- size = "medium"

+- price = 4999

+- whatnot

+- fruits

+- (1st) = "orange"

+- (2nd) = "banana"

这种对scalars的访问使用索引，如animals[0].name

* 模板

在FreeMarker模板中可以包括下面三种特定部分：

1. ? ${…}：称为interpolations，FreeMarker会在输出时用实际值进行替代
2. ? FTL标记（FreeMarker模板语言标记）：类似于HTML标记，为了与HTML标记区分，用#开始（有些以@开始，在后面叙述）
3. ? 注释：包含在<#--和-->（而不是<!--和-->）之间

下面是一些使用指令的例子：

1. ? if指令

<#if animals.python.price < animals.elephant.price>

Pythons are cheaper than elephants today.

<#else>

Pythons are not cheaper than elephants today.

</#if>

1. ? list指令

<p>We have these animals:

<table border=1>

<tr><th>Name<th>Price

<#list animals as being>

<tr><td>${being.name}<td>${being.price} Euros

</#list>

</table>

输出为：

<p>We have these animals:

<table border=1>

<tr><th>Name<th>Price

<tr><td>mouse<td>50 Euros

<tr><td>elephant<td>5000 Euros

<tr><td>python<td>4999 Euros

</table>

1. ? include指令

<html>

<head>

<title>Test page</title>

</head>

<body>

<h1>Test page</h1>

<p>Blah blah...

<＃i nclude "/copyright\_footer.html">

</body>

</html>

1. ? 一起使用指令

<p>We have these animals:

<table border=1>

<tr><th>Name<th>Price

<#list animals as being>

<tr>

<td>

<#if being.size = "large"><b></#if>

${being.name}

<#if being.size = "large"></b></#if>

<td>${being.price} Euros

</#list>

</table>

# Demo

#### Decorator.java



#### Top2.ftl

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<html><body>

<DIV id=logo><A href="http://www.zhuopin.com/"><IMG alt=ShopNC多用户综合购物平台

src="zhuopin\_files/logo.png"></A></DIV>

<DIV id=search class=search>

<DIV class=details>

<UL id=header\_tab class=tab>

<LI class=current rel="a1"><SPAN>商品</SPAN> </LI>

<LI rel="a2"><SPAN>店铺</SPAN> </LI>

<LI rel="a3"><SPAN>帮助</SPAN> </LI></UL>

<DL id=a1>

<FORM id=search\_product onsubmit=searchProductSubmit(); method=get

action=http://www.zhuopin.com/home/product.php>

<DIV class=formstyle><INPUT style="COLOR: #cccccc" id=search\_product\_text

class=textinput value=请输入您需要的商品 name=keyword> <SPAN class=searchBtn><A

onclick="$('#search\_product').submit();"

href="http://www.zhuopin.com/#">search</A></SPAN> </DIV><INPUT value=product

type=hidden name=search\_name> <INPUT value=0 type=hidden name=recommended>

<INPUT value=0 type=hidden name=searchcate> </FORM></DL>

<DL style="DISPLAY: none" id=a2>

<FORM id=search\_shop onsubmit=searchShopSubmit(); method=get

action=http://www.zhuopin.com/home/shop.php>

<DIV class=formstyle><INPUT style="COLOR: #cccccc" id=search\_shop\_text

class=textinput value=请输入您需要的店铺 name=keyword> <SPAN class=searchBtn><A

onclick="$('#search\_shop').submit();"

href="http://www.zhuopin.com/#">search</A></SPAN> </DIV><INPUT value=shop

type=hidden name=search\_name> <INPUT value=0 type=hidden name=recommended>

<INPUT value=0 type=hidden name=searchcate> </FORM></DL>

<DL style="DISPLAY: none" id=a3>

<FORM id=search\_help onsubmit=searchHelpSubmit(); method=get

action=http://www.zhuopin.com/help/index.php>

<DIV class=formstyle><INPUT style="COLOR: #cccccc" id=search\_help\_text

class=textinput value=请输入您需要的帮助 name=q> <SPAN class=searchBtn><A

onclick="$('#search\_help').submit();"

href="http://www.zhuopin.com/#">search</A></SPAN> </DIV></FORM></DL>

<DIV class=tag>

<H4>${name}:</H4>

<UL>

<#list product as temp>

<A

href="${temp.href}"

target=\_blank> ${temp.name}</A>&nbsp;&nbsp;

</#list>

</UL></DIV></DIV>

<DIV class=links>

<P><A href="http://www.zhuopin.com/home/product.php?action=search">高级搜索</A></P>

<P><A

href="http://www.zhuopin.com/help/index.php?q=搜索">搜索帮助</A></P></DIV></DIV></DIV>

</body></html>

#### top2.jsp



这样浏览top2.jsp时输出的结果为：



# 总结freemarker特性

1、通用目标

能够生成各种文本：HTML、XML、RTF、Java源代码等等

易于嵌入到你的产品中：轻量级；不需要Servlet环境

插件式模板载入器：可以从任何源载入模板，如本地文件、数据库等等

你可以按你所需生成文本：保存到本地文件；作为Email发送；从Web应用程序发送它返回给Web浏览器

2、强大的模板语言

所有常用的指令：include、if/elseif/else、循环结构

在模板中创建和改变变量

几乎在任何地方都可以使用复杂表达式来指定值

命名的宏，可以具有位置参数和嵌套内容

名字空间有助于建立和维护可重用的宏库，或者将一个大工程分成模块，而不必担心名字冲突

输出转换块：在嵌套模板片段生成输出时，转换HTML转义、压缩、语法高亮等等；你可以定义自己的转换

3、通用数据模型

FreeMarker不是直接反射到Java对象，Java对象通过插件式对象封装，以变量方式在模板中显示

你可以使用抽象（接口）方式表示对象（JavaBean、XML文档、SQL查询结果集等等），告诉模板开发者使用方法，使其不受技术细节的打扰

4、为Web准备

在模板语言中内建处理典型Web相关任务（如HTML转义）的结构

能够集成到Model2 Web应用框架中作为JSP的替代

支持JSP标记库

为MVC模式设计：分离可视化设计和应用程序逻辑；分离页面设计员和程序员

5、智能的国际化和本地化

字符集智能化（内部使用UNICODE）

数字格式本地化敏感

日期和时间格式本地化敏感

非US字符集可以用作标识（如变量名）

多种不同语言的相同模板

6、强大的XML处理能力

<#recurse> 和<#visit>指令（2.3版本）用于递归遍历XML树

在模板中清楚和直觉的访问XML对象模型