

EasyJWeb-Velocity 脚本简明教程

很多人下载了 EasyJWeb 的开源应用示例，但是对动态页面模板文件中的标签使用不是很熟悉，这里简单介绍一下。EasyJWeb 特定把视图限定为 Velocity，因为我们觉得在 Velocity 在表现上从可维护性、简洁性乃灵活性上，比 JSP 及其它视图技术表现强得多。虽然我们可以非常简单把 EasyJWeb 扩展成支持多种视图技术，但当前我们不打算这样做！

使用 EasyJWeb 的目的，是在于把页面—程序完成分开，就也就是一个项目中，程序员使用专业 Java 开发工具（如 Eclipse、JBuilder 等）来编写、调试、测试程序，页面制作人员使用专业的网页制作工具（如 Macromedia Dreamweaver）来设计制作网页，而两者之间的协调通过一个规范的接口协议来解决。需要在页面里面加如一些标签，来生成动态内容，这一工作可以交由网页制作人员来完成。因为使用 Velocity 作为视图，由于他的语法、功能及使用方法都比较简单，因此一般情况下，一天以内就能让页面制作人员熟练掌握其用法。

Velocity 是一个基于 java 的模板引擎（template engine），它允许任何人仅仅简单的使用模板语言（template language）来引用由 java 代码定义的对象。作为一个比较完善的模板引擎，Velocity 的功能是比较强大的，但强大的同时也增加了应用复杂性。

一、基本语法

1、"# "用来标识 Velocity 的脚本语句，包括 #set、#if、#else、#end、#foreach、#end、#include、#parse、#macro 等；

如：

```
#if($info.imgs)

#else

#end
```

2、"\$ "用来标识一个对象(或理解为变量)；如

如：\$i、\$msg、\$TagUtil.options(...)等。

3、"{}"用来明确标识 Velocity 变量。

比如在页面中，页面中有一个 \$someone name，此时，Velocity 将把 someone name 作为变量名，若我们程序是想要在 someone 这个变量的后面紧接着显示 name 字符，则上面的标签应该改成 \${someone} name。

4、"! "用来强制把不存在的变量显示为空白。

如当页面中包含 \$msg，如果 msg 对象有值，将显示 msg 的值，如果不存在 msg 对象同，则在页面中将显示 \$msg 字符。这是我们不希望的，为了把不存在的变量或变量值为 null 的对象显示为空白，则只需要在变量名前加一个 “!” 号即可。

如：\$!msg

二、在 EasyJWeb 中的最佳实践

EasyJF 团队普通文档通用模板

理论上你可以在 EasyJWeb 模板使用所有 Velocity 的脚本及功能，但我们不推荐你在界面模板中使用过多过复杂的脚本表达方式，在万不得已的情况下，不要在界面模板中加入任何复杂的逻辑，更不要在界面模板中加入变量声明、逻辑运算符等等。

在 EasyJWeb 中，我们提供了五条基本的模板脚本语句，基本上就能满足所有应用模板的要求。这四条模板语句很简单，可以直接由界面设计人员来添加。在当前很多 EasyJWeb 的应用实践中，我们看到，所有界面模板中归纳起来只有下面四种简单模板脚本语句即可实现：

1、`!obj` 直接返回对象结果。

如：在 html 标签中显示 java 对象 msg 的值。 `<p>${msg}</p>`

在 html 标签中显示经过 HtmlUtil 对象处理过后的 msg 对象的值。 `<p>${HtmlUtil.doSomething(${msg})}</p>`

2、`#if(!obj)#else #end` 判断语句

如：在 EasyJWeb 各种开源应用中，我们经常看到的用于弹出提示信息 msg 的例子。

```
#if($msg)
```

```
<script>
```

```
alert('${msg}');
```

```
</script>
```

```
#end
```

上面的脚本表示当对象 msg 对象存在时，输出 `<script>` 等后面的内容。

3、`#foreach($info in $list) $info.someList #end` 循环读取集合 list 中的对象，并作相应的处理。

如：EasyJF 开源论坛系统中论(0.3)坛首页显示热门主题的 html 界面模板脚本：

```
#foreach( $info in $hotList1)
```

```
<a href="/bbsdoc.ejf?easyJWebCommand=show&&cid=${info.cid}" target="_blank">${info.title}</a><br>
```

```
#end
```

上面的脚本表示循环遍历 hotList1 集合中的对象，并输出对象的相关内容。

4、`#macro(macroName)#end` 脚本函数(宏)调用，不推荐在界面模板中大量使用。

如：在使用 EasyJWeb Tools 快速生成的添删改查示例中，可以点击列表的标题栏进行升降排序显示，这是我们在 EasyJWeb 应用中经常看到的一个排序状态显示的模板内容。

函数(宏)定义，一般放在最前面

```
#macro(orderPic $type)
```

```
#if($orderField.equals($type))
```

```

```

```
#end
```

```
#end
```

具体的调用如： `头衔#orderPic("title")`

5、包含文件 `#inlude("模板文件名")`或 `#parse("模板文件名")`

主要用于处理具有相同内容的页面，比如每个网站的顶部或尾部内容。

使用方法，可以参考 EasyJF 开源 Blog 及 EasyJF 开源论坛中的应用！

如：`#parse("/blog/top.html")`或`#include("/blog/top.html")`

`parse` 与 `include` 的区别在于，若包含的文件中有 Velocity 脚本标签，将会进一步解析，而 `include` 将原样显示。

三、关于#set 的使用

在万不得已的时候，不要在页面视图自己声明 Velocity 脚本变量，也就是尽量少使用 `#set`。有时候我们需要在页面中显示序号，而程序对象中又没有包含这个序号属性同，可以自己定义。如在一个循环体系中，如下所示：

```
#set ($i=0)
#foreach($info in $list)
  序号:$i
#set($i=$i+1)
#end
```

可以使用 `#set` 声明的

四、Velocity 脚本语法摘要

1、声明：`#set ($var=XXX)`

左边可以是以下的内容

Variable reference

String literal

Property reference

Method reference

Number literal `#set ($i=1)`

ArrayList `#set ($arr=["yt1","t2"])`

算术运算符

2、注释：

单行`## XXX`

多行`##* xxx`

`XXXX`

`XXXXXXXXXXXX*##`

References 引用的类型

3、变量 Variables

以 "\$" 开头，第一个字符必须为字母。character followed by a VTL Identifier. (a .. z or A .. Z).

变量可以包含的字符有以下内容：

alphabetic (a .. z, A .. Z)

numeric (0 .. 9)

hyphen ("-")

underscore ("_")

4、Properties

```
$Identifier.Identifier
```

```
$user.name
```

hashtable user 中的的 name 值.类似: user.get("name")

5、Methods

```
object user.getName() = $user.getName()
```

6、Formal Reference Notation

用 {}把变量名跟字符串分开

如

```
#set ($user="csy")
```

```
${user}name
```

返回 csyname

```
$username
```

```
#!username
```

\$与\$!的区别

当找不到 username 的时候, \$username 返回字符串"\$username", 而#!username 返回空字符串""

7、双引号 与 引号

```
#set ($var="helo")
```

```
test"$var" 返回 testhelo
```

```
test'$var' 返回 test'$var'
```

可以通过设置 stringliterals.interpolate=false 改变默认处理方式

8、条件语句

```
#{if ($foo )
```

```
<strong>Velocity!</strong>
```

```
#end
```

```
#{if($foo)
```

```
#{elseif()
```

```
#{else
```

```
#{end
```

当\$foo 为 null 或为 Boolean 对象的 false 值执行.

9、逻辑运算符:== && || !

10、循环语句#foreach(\$var in \$arrays) // 集合包含下面三种 Vector, a Hashtable or an Array

```
#end
#foreach( $product in $allProducts )
<li>$product</li>
#end

#foreach( $key in $allProducts.keySet() )
<li>Key: $key -> Value: $allProducts.get($key)</li>
#end

#foreach( $customer in $customerList )
<tr><td>$velocityCount</td><td>$customer.Name</td></tr>
#end
11、velocityCount 变量在配置文件中定义
# Default name of the loop counter
# variable reference.
directive.foreach.counter.name= velocityCount
# Default starting value of the loop
# counter variable reference.
directive.foreach.counter.initial.value= 1
```

12、包含文件

```
#include( "one.gif","two.txt","three.htm")
```

13、Parse 导入脚本

```
#parse("me.vm")
```

14、#stop 停止执行并返回

15、定义宏 Velocimacros ,相当于函数 支持包含功能

```
#macro( d )
<tr><td></td></tr>
#end
调用
#d()
```

16、带参数的宏

```
#macro( tablerows $color $somelist )
#foreach( $something in $somelist )
<tr><td bgcolor=$color>$something</td></tr>
#end
#end
```

17、Range Operator

```
#foreach( $foo in [1..5])
```

附：《浅析 MVC 框架中 View 层的优雅设计及实例》

作者：EasyJF 开源团队 大峡

在基于 B/S 的应用程序开发中，从基本的技术分工上来说就是两大块，一是软件显示界面，另一个是程序逻辑。在 N 年前的脚本语言时代，无论是 asp、php 还是 jsp，我们基本是把这两者柔和在一起的。尽管我们想方设法做好很多函数或者包含文件来努力达到软件的复用，但仍然无法满足多变的用户需求，这主要是因为以前的纯脚本编码方式无法很好支持及应用面向对象(OO)领域中的强大功能。

在常见的 B/S 软件项目中，界面的设计包括 html 界面、Wap 界面及其它由文本字符协议为基本表示的界面等。以我们接触最多的 html 页面为例子，在做这些界面的时候往往需要美工先使用 photoshop 或 fireworks 等图形界面设计工具进行全局设计，然后再使用进行 Dreamweaver 等 html 页面制作工具进行加工制作。而强大的程序逻辑及后台处理都是由服务器端程序完成，这些程序具有较高的稳定性，其开发工具如 JBuilder、Eclipse 等对 View 层的界面无法很好的支持(当然那些用记事本写界面的应用除外)，这就使得很多 MVC 框架的设计都无法两全齐美。

作位一个比较友好的 MVC 的框架，在简化服务器应用开发的同时，还需要在 View 这一层设计上不要过多的影响到界面人员的工作，最基本的要求就是不要过多的加入一些设计软件不支持的标签等元素(如 Struts 的很多标签在 Dreamweaver 中都不支持)。这里我们以 EasyJWeb 为例，探讨在 View 层如何实现比较合理的设计。

EasyJWeb 作为一个快速 Java Web MVC 框架，其设计目标不尽是要简化软件开发人员的代码书写工作，更是要能方便界面设计人员的工作。

当然，要使界面能跟后台程序逻辑能融合，显示界面及程序逻辑之间需要一定的对话协议在所难免。EasyJWeb 作为一个 MVC 框架，同样也存在着这样的对话协议，这就是 EasyJWeb 中的界面模板标识语言。

在当前发布的版本中，EasyJWeb 界面模板标识语言使用的是 Apache 开源的 Velocity 模板引擎(template engine)，当然以后会根据实际需要加入更多的模板引擎以供框架应用者选择。

Velocity 是一个基于 java 的模板引擎 (template engine)，它允许任何人仅仅简单的使用模板语言 (template language) 来引用由 java 代码定义的对象。作为一个比较完善的模板引擎，Velocity 的功能是比较强大的，但强大的同时也增加了应用复杂性。

理论上你可以在 EasyjWeb 模板使用所有 Velocity 的脚本及功能，但我们不推荐你在界面模板中使用过多过复杂的脚本表达方式，在万不得已的情况下，不要在界面模板中加入任何复杂的逻辑，更不要在界面模板中加入变量声明、逻辑运算符等等。

在 EasyJWeb 中，我们提供了四条基本的模板脚本语句，基本上就能满足所有应用模板的要求。这四条模板语句很简单，可以直接由界面设计人员来添加。在当前很多 EasyJWeb

EasyJF 团队普通文档通用模板

的应用实践中，我们看到，所有界面模板中归纳起来只有下面四种简单模板脚本语句即可实现：

1、`!obj` 直接返回对象结果。

如：在 html 标签中显示 java 对象 `msg` 的值。`<p>!msg</p>`

在 html 标签中显示经过 `HtmlUtil` 对象处理过后的 `msg` 对象的值。`<p>!HtmlUtil.doSomething(!msg)</p>`

2、`#if(!obj)#else#end` 判断语句

如：在 EasyJWeb 各种开源应用中，我们经常看到的用于弹出提示信息 `msg` 的例子。

```
#if($msg)
<script>
alert('!msg');
</script>
#end
```

上面的脚本表示当对象 `msg` 对象存在时，输出 `<script>` 等后面的内容。

3、`#foreach($info in $list) $info.something #end` 循环读取集合 `list` 中的对象，并作相应的处理。

如：EasyJF 开源论坛系统中论坛首页显示热门主题的 html 界面模板脚本

```
#foreach($info in ShotList1)
<a href="/bbsdoc.ejf?easyJWebCommand=show&&cid=!info.cid"
target="_blank">!info.title</a><br>
#end
```

上面的脚本表示循环遍历 `hotList1` 集合中的对象，并输出对象的相关内容。

4、`#macro(macroName)#end` 脚本函数(宏)调用，不推荐在界面模板中大量使用。

如：在 EasyJF 简易订销管等系统中经常看到的一个排序状态显示的模板内容。

函数(宏)定义，一般放在最前面

```
#macro(orderPic $type)
#if($orderField.equals($type))

#end
#end
```

具体的调用如：`头衔#orderPic("title")`

总结：

当然，在实际应用项目中，为了实现界面的更加友好、人性化，会出现很多复、易变的需求。如根据对象的不同状态，显示不同的提示颜色、提示语音等功能。在这种时候，仍然要慎用太多的模板脚本功能，大多数需求都可以通过变通的方式解决，有些信息可以直接在对象中增加逻辑转化信息，有些界面要求可以通过在界面中使用与特定界面有关的表达方式来实现，如 html 页面中使用 javascript、css，Wap 页面中使用 WMLScript 等等。只有，这样才能确保你的系统核心不受界面的的影响及控制，才能更好的扩展及维护。

我们希望你能成为真正的程序高手，而不只是精通某个处于表层的脚本语言，因此也不希望广大 Java 开发人员在 View 层消耗太多的时间。

由于水平有限，本文所谈的内容有 N 多不合理或者需要改进的地方，恳请大家指正！

附：EasyJWeb 简介

EasyJWeb 是基于 java 技术，应用于 WEB 应用程序快速开发的 MVC 框架，框架设计构思来源于国内众多项目实践，充分借鉴了当前主流的开源 Web 框架（Struts、JSF、Tapestry、Webwork 等），吸取了其优点及精华，利用 Velocity 作为模板页面引擎，是一个实现了页面及代码完全分离的 MVC 开发框架，是一个旨在于为中小型 Web 应用系统提供快速开发实践的简易 Web 框架。

EasyJF 开源团队于 2006 年初才开始建设，因此当前整个开发团队组建以及所发布的作品，都显得极不成熟。EasyJWeb 仍然处于测试阶段，请广大的 Java 爱好者多多批评及建议，同时也非常欢迎您能加入到我们的国产开源队伍中。

EasyJWeb 官方网址：www.easyjf.com

EasyJF 团队官方网址：www.easyjf.com