

前端开发 VM 模板环境说明文档

一、	Velocity 模板(VM)语言介绍	2
1.	Velocity 的定义	2
2.	Velocity 的基本语法	3
3.	引用符: \$	3
4.	安静引用符: !	3
5.	形式引用符: { }	4
6.	定义语句: #set	4
7.	条件语句: #if, #elseif, #else	4
8.	循环语句: #foreach	5
9.	宏定义语句: #macro	6
10.	包含语句: #include, #parse	7
11.	停止语句: #stop	7
12.	整数运算	7
13.	逻辑运算符	8
14.	字符串的加法	8
15.	转义	8
16.	注释	9
二、	开发与发布环境基本设置	10

一、 Velocity 模板(VM)语言介绍

1. Velocity 的定义

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。

当 Velocity 应用于 web 开发时，界面设计人员可以和 java 程序开发人员同步开发一个遵循 MVC 架构的 web 站点，也就是说，页面设计人员可以只关注页面的显示效果，而由 java 程序开发人员关注业务逻辑编码。Velocity 将 java 代码从 web 页面中分离出来，这样为 web 站点的长期维护提供了便利，同时也为我们在 JSP 和 PHP 之外又提供了一种可选的方案。

例：

```
<html>
<body>
Hello $customer.Name!
<div>
<ul>
#foreach( $mud in $nudsOnSpecial )
#if( $customer.hasPurchased( $mud ) )
<li>$flogger.getPromo( $mud )</li>
#end
#end
</ul>
</div>
```

Velocity 是一个基于 java 的模板引擎（template engine）

Velocity Template Language (VTL)来引用 java 对象。

渲染: Velocity 模版就好比一张空白表格, 表格中留白的部分等着 java 程序来填写, 这一过程 叫做 模版渲染

2. Velocity 的基本语法

“#” 开头 (#*directive*, 指令):

如: #set (声明语句, 声明一个对象)

#if 、#else、#end (条件判断)

#foreach、#end (循环)

#include、#parse (包含, 嵌入子模板)

#macro (定义宏方法)

3. 引用符: \$

Velocity 采用\$来调用对象, 对象的属性, 对象的方法

例如:

\$memberId (调用对象)

\$offer.memberId (调用对象某属性)

\$offer.getMemberId() (调用对象的某方法)

变量的简略标记是有一个前导"\$"字符后跟一个 VTL 标识符 (Identifier.) 组成。一个 VTL 标识符必须以一个字母开始(a .. z 或 A .. Z)。剩下的字符将由以下类型的字符组成:

字母 (a .. z, A .. Z)

数字 (0 .. 9)

连字符("-")

下划线 ("_")

4. 安静引用符: !

用来强制把不存在的变量显示为空白，当你不清楚某个变量是否存在，加上“！”

例如：渲染模版的 JAVA 程序中未定义 `memberId` 变量，

如模版中调用 `$memberId`，会直接将“`$memberId`”打出来。但是调用 `$! memberId`，就什么都不显示

5. 形式引用符： `{}`

“`{}`”用来明确标识 Velocity 变量，当你调用变量后面紧跟着其它的 html 代码时，一定要用 `{}` 将变量名括起来。

例如：有个变量 `memberId`，它的值是 Tom，在模板中调用 `$memberIdisBadBoy`，会把 `$memberIdisBadBoy` 作为变量名，并将“`$memberIdisBadBoy`”直接打出来

正确写法：`${memberId} isBadBoy`，会显示：`TomisBadBoy`

6. 定义语句： `#set`

```
#set ($foo = "text" ) String
#set ($foo = 100 ) Integer
#set ($foo = [ 1, 2, 3 ] ) List
#set ($foo = { 1 : 2, 3 : 4 } ) Map
#set ($foo = $bar ) Variable
#set ($foo = $foo.Bar ) Property
#set ($foo = $foo.getValue() ) Method
```

7. 条件语句： `#if`， `#elseif`， `#else`

如：

```
<script language="javascript" type="text/javascript">
  #if($!sex == "male")
    alert('你好，先生！');
  #elseif($!sex == "female")
    alert('你好，女士！');
  #else
    alert('你好！');
  #end
</script>
```

8. 循环语句: #foreach

循环读取集合 list 中的对象，并作相应的处理

```
#foreach( $info in $list) $info.someList #end
```

例 1:

```
<div>
  <ul>
    #forEach($offer in $offerList)
      <li>$velocityCount</li>
      <li>$!offer.subject</li>
    #end
  </ul>
</div>
```

例 2:

```
#foreach ($i in [ "a", "b", "c", "d" ])
The current letter is ${i}<br/>
#end
```

例 3:

```
#foreach ($i in [ 1..10 ])
The current number is ${i}<br/>
#end
```

9. 宏定义语句: #macro

#macro(macroName, var1,var2 ……)#end 脚本函数(宏)调用, 不推荐在界面模板中大量使用

例: 定义 tablerow 宏

```
#macro ( tablerows $color $offerlist )
    #foreach ( $offer in $offerlist )
        <li>${offer.subject}</li>
    #end
#end
```

调用#tablerows 来画表格:

```
#set ( $color = "blue" )  
<div><ul> # tablerows($color , $offerlist ) </ul></div>
```

10. 包含语句: #include, #parse

使用方法,如:

```
#parse( "/blog/top.vm" )  
#include( "/blog/top.vm" )  
#include( "one.gif","two.txt","three.htm" )  
#include( "greetings.txt", $seasonalstock )
```

parse 与 include 的区别在于,若包含的文件中有 Velocity 脚本标签,将会进一步解析并将结果渲染成 html 显示出来,而 include 将原样显示。

11. 停止语句: #stop

#stop 停止语句,执行到#stop时,模版引擎立刻停止渲染,通常用于调试

12. 整数运算

```
+, -, *, /, %  
#set( $foo = $bar + 3 )  
#set( $foo = $bar - 4 )  
#set( $foo = $bar * 6 )  
#set( $foo = $bar / 2 )  
#set( $foo = $bar % 5 )
```

13. 逻辑运算符

!, &&, ||, ==, <=, <, >, >=

14. 字符串的加法

字符串不能用 + 运算

```
#set( $size = "Big" )
#set( $name = "Ben" )
#set($clock = $size + $name )      错误!!!!
#set($clock = "$size$name" )      正确!!!!!!!!!!
The clock is $clock.

结果: The clock is BigBen
```

15. 转义

用 \ 来进行转义, 让#失效

例:


```
\#if( $jazz )
    Vyacheslav Ganelin
\#end
输出的是
\#if( $jazz )
    Vyacheslav Ganelin
\#end

\\#if( $jazz )
    Vyacheslav Ganelin
\\#end
输出的是  \ Vyacheslav Ganelin\
```

16. 注释

注释单条语句 用 ##

例:

```
##set($memberId= "jack" )
```

注释一段语句 用## *#

例:

```
*#
#set($memberId1= "jack" )
#set($memberId2="tom")
*#
```

参考资源: <http://blog.csdn.net/xuzhongxuan/archive/2008/08/01/2755305.aspx>

二、 开发与发布环境基本设置

我们的模板开发环境大部分都是基于 WebX 框架，页面使用了 velocity 模板，WEB 层使用的是 WEBX 框架里面的 TemplateAction, TemplateScreen, AbstractModule, TemplateControl, 解决了操作和页面的跳转。持久层使用 ibatis，Bean 容器使用的是 spring 具体形式是由 layou, screen, control 三部分组成。

Layout 代表一个或一系列页面的外框架

Screen 代表一个或一系列页面的主体内容区

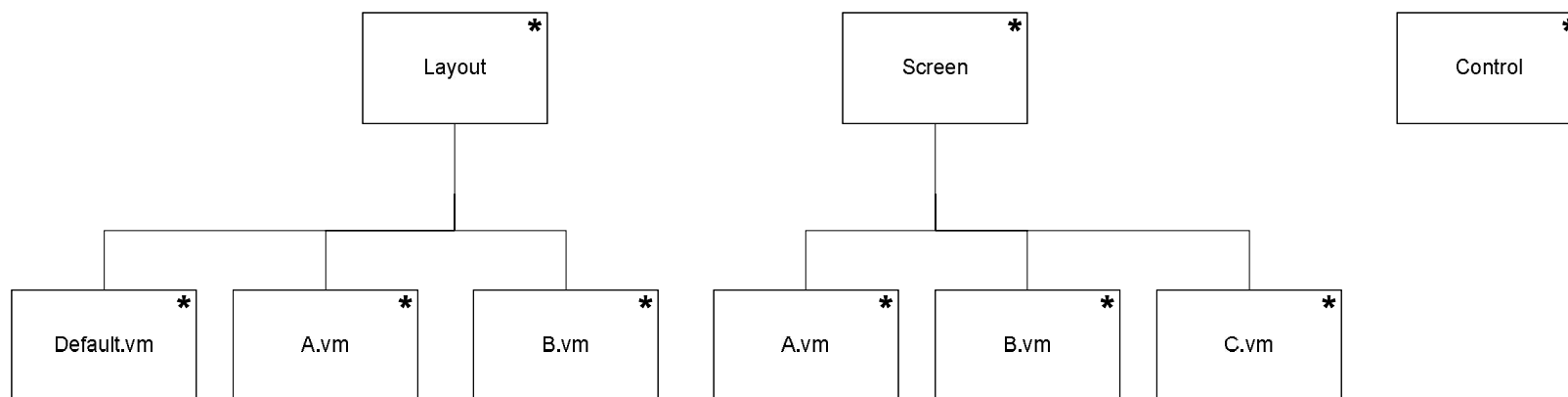
Control 代表一系列页面共用的一些功能模块，具有自己的程序和逻辑

每个 screen 里的 vm 都会有相应同名的 layout 模板相对应，作为其框架，如果没有对应，则对应 layout 最近级目录中的 default.vm; 而 control 可以被 screen 引用，也可以被 layout 引用，做为共用功能模块

如下图结构:

A.vm 和 B.vm 都有其各自对应的同名 layout A.vm B.vm

C.vm 没有其对应的同名的 laout,则对应 layout 中最近级目录下的 default.vm



映射服务器上目录结构:

