

进一步提升 Struts 2 对 Velocity 的支持力度

刘冬 (javayou@gmail.com), 开发工程师,

发布日期: 2007 年 7 月 30 日

简介: 本文主要介绍如何解决 Struts 2 和最新版本的 Velocity 结合的问题以及进一步提升 Struts 2 对 Velocity 的支持力度。

级别: 中级

访问情况 2599 次浏览

建议: ▀



标记本文!

Struts 2 是 Struts 框架的一个全新版本, 它废弃了原有 Struts 1.x 的框架结构, 在 WebWork 的基础上全面提升了代码的配置灵活性、可测试性以及超强的扩展性。Struts 2 更是提供了对 Velocity 和 FreeMarker 模板引擎的支持, 这破天荒的举动大大的方便了 Velocity 和 FreeMarker 的开发者, 从而也更有效的推动了模板技术的发展。

在进行 Struts 2 和 Velocity 结合的试验中发现下面几个问题:

- 1. 与 VelocityTools 1.3 存在兼容性问题;
- 2. 不支持 Velocity Layout;
- 3. 处理 velocity 模板时存在编码问题;

好在由于 Struts 2 的超强扩展性, 使得这三个问题的解决变得非常简单, 而且我们无需修改 Struts 2 的源码, 下面分别给出三个问题的解决办法。

第一个问题: 与 VelocityTools 1.3 存在兼容性问题

这个问题是因为 VelocityTools 这个项目从 1.2 升级到 1.3 时修改了一些类的方法 (Method) 导致的。在 1.3 这个版本中 ToolboxManager 类的 getToolboxContext 方法改名为 getToolbox, 因此在启动程序的时候就会报 getToolboxContext 方法没找到的异常。

Struts 2 是在 VelocityManager 类的 createContext 方法中调用 ToolboxManager 的, 而 Struts 2 允许通过配置来修改 VelocityManager 的实现类, 因此我们只需要从 VelocityManager 继承一个子类, 并重写 createContext 方法即可。重写的代码很简单, 直接从 VelocityManager 的 createContext 方法中拷贝代码然后将 getToolboxContext 改为 getToolbox。

接着需要修改 Struts 2 的配置来启用这个新的子类, 配置如下 (struts.xml) :

```
<constant name="struts.velocity.manager.classname" value="struts2.VelocityFixedManager" />
```

或者是 struts.properties :

```
struts.velocity.manager.classname = struts2.VelocityFixedManager
```

内容
▪ 第一个问题: 与 VelocityTools 1.3 存在兼容性问题
▪ 第二个问题: 不支持 Velocity Layout
▪ 第三个问题: 处理 velocity 模板时存在编码问题
▪ 总结
▪ 参考资料
▪ 关于作者
▪ 建议

 回首页

第二个问题: 不支持 Velocity Layout

首先简单的介绍一下 Velocity 的 Layout 技术。Velocity 的 Layout 是通过 VelocityTools 这个项目来实现的。大多数网站上每个页面基本上都会遵循相同的排版, 例如一个页面由 banner 条、内容和底部版权说明组成。那么对于该网站上的每个网页, 我们都需要编写重复的代码来显示 banner 条以及底部版权说明。而 Velocity 的 Layout 技术可以让我们把页面的布局和页面的内容分开到不同 velocity 模板文件中。要使用 Layout 技术只需要把 VelocityViewServlet 换成 VelocityLayoutServlet 即可。VelocityLayoutServlet 内置了一个变量 \$screen_content, 它首先执行内容模板页面并把执行后的字符串结果存入 \$screen_content 变量中, 然后再去执行布局页面, 这样就相当于把内容页面嵌入到布局页面。

但是这个问题在 Struts 2 环境中就要麻烦很多。因为 Struts 2 并不是采用 Velocity Tools 那种基于 Servlet 的方式来调用 velocity 模板, 所以我们没法只是简单的替换成 VelocityLayoutServlet 就能解决。在 Struts 2 的默认配置中定义了一个名为 velocity 的 ResultType, 它对应的类是 org.apache.struts2.dispatcher.VelocityResult。该类没有跟 VelocityTools 打交道, 而是直接调用 Velocity 的核心 API 来执行 velocity 模板。因此我们只好根据 Velocity Tools 项目中 VelocityLayoutServlet 的思路来改写 Struts 2 对 velocity 模板的处理方法。

关于 ResultType

Struts 2 中把各种各样的视图例如 JSP、Velocity、FreeMarker 等等都封装成 ResultType 的子类, 也就是说每一种视图 Struts 2 都有与之对应的 ResultType 的实现类, 本文中 Velocity 视图对应的类是 VelocityResult

在处理之前我们先大概解释一下 VelocityLayoutServlet 的工作原理，先看几个程序源码：

```
<!-- layout.vm -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<HTML>
  <HEAD>
    <TITLE>${page_title}</TITLE>
  </HEAD>
  <BODY>
    ${screen_content}
  </BODY>
</HTML>
```

上面是布局页面的代码，这个布局页面包含两个变量，一个是 \$page_title，用来指定页面标题；第二个是 \$screen_content，这是 VelocityLayoutServlet 定义的变量，这个变量最终会被替换成实际页面的内容。

```
## index.vm
#set($layout = "layout.vm") ##选择布局页面
#set($page_title ="Struts2Velocity")
#set($name = "Winter Lau")
<p>
Hello <b>$name</b>, Welcome to Struts2 with velocity <br/>
</p>
```

上面是另外一个页面 index.vm 的代码。当 VelocityLayoutServlet 处理 index.vm 时，首先执行 index.vm，并把执行后的结果保存在一个字符串中，并往 VelocityContext 中增加变量名为 \$screen_content，值为 index.vm 的执行结果，然后执行 layout.vm 页面，这样最终呈现在用户面前的就是 index.vm 嵌入在 layout.vm 的效果。

因此我们可以依照上面的原理，从 VelocityResult 中继承一个子类并重写其 doExecute 这个方法。详细的代码请看附件中的源码。

类编写完毕同样需要对 Struts 进行如下配置（struts.xml），这个配置是告诉 Struts 使用 struts2.VelocityLayoutResult 这个类来处理采用 velocity 开发的视图：

```
<result-types>
<result-type name="velocity" class="struts2.VelocityLayoutResult"/>
</result-types>
```

接下来就可以测试新的功能了，我们配置 action 如下，打开浏览器访问 http://localhost:<port>/index.do 看看效果吧。

```
<action name="index">
  <result type="velocity"/>index.vm</result>
</action>
```

这样我们不需要 VelocityTools 也可以让 Struts 2 支持 Velocity Layout 了。

[🏠 返回首页](#)

第三个问题：处理 velocity 模板时存在编码问题

Velocity 有几个参数跟编码有关的分别是：

```
default.contentType=text/html
input.encoding=GBK
output.encoding=UTF-8
```

其中第一个参数指定了 velocity 模板执行后输出到浏览器时的 ContentType；第二个参数是 Velocity 引擎以什么编码方式读取 velocity 模板文件；第三个参数是输出页面的编码。

但是采用 Struts 2 后发现这三个参数并没有起作用，深入 Struts 2 的代码发现在处理 vm 编码时采用的是 Struts 2 的编码配置，而且读取 velocity 模板文件和输出 html 到浏览器采用的相同的编码，这种方式个人觉得 Struts 2 处理欠妥当，为了保留 Velocity 的特色，在 VelocityLayoutResult 的基础上进一步改进。我们需要重载 VelocityResult 的 setVelocityManager 方法，因为这个方法是用来初始化 Velocity 引擎的，我们在初始化的过程中读取前面提到的三个配置参数并把参数值保存在类的属性中，详细的代码如下：

```
@Inject
@Override
public void setVelocityManager(VelocityManager mgr) {
    super.setVelocityManager(mgr);
    if(mgr != null && velocityManager==null){
        this.velocityManager = mgr;
        ServletContext ctx = ServletActionContext.getServletContext();
        velocityManager.init(ctx);
        VelocityEngine engine = velocityManager.getVelocityEngine();
        defaultLayout = (String)engine.getProperty(PROPERTY_DEFAULT_LAYOUT);
        layoutDir = (String)engine.getProperty(PROPERTY_LAYOUT_DIR);
        if (!layoutDir.endsWith("/")){
            layoutDir += '/';
        }
    }
}
```

```
    }

    if (!layoutDir.startsWith("/")){
        layoutDir = "/" + layoutDir;
    }

    // for efficiency's sake, make defaultLayout a full path now
    defaultLayout = layoutDir + defaultLayout;

    inputEncoding = (String)engine.getProperty(PROPERTY_INPUT_ENCODING);
    outputEncoding = (String)engine.getProperty(PROPERTY_OUTPUT_ENCODING);
    contentType = (String)engine.getProperty(PROPERTY_CONTENT_TYPE);

    if (outputEncoding != null && contentType != null) {
        contentType = contentType + ";charset=" + outputEncoding;
    }
}
```

同时修改 doExecute 方法以使用正确的编码进行处理，主要代码如下（请注意标识为粗体的的代码）：

```
Template t = getTemplate(stack, velocityManager.getVelocityEngine(),
    invocation, finalLocation, inputEncoding);

Context context = createContext(velocityManager, stack, request, response, finalLocation);
//将页面执行结果写入到字符串缓存中
StringWriter sw = new StringWriter();
t.merge(context, sw);
context.put(KEY_SCREEN_CONTENT, sw.toString());

Object obj = context.get(KEY_LAYOUT);
String layout = (obj == null) ? null : obj.toString();
if (layout == null)
{
    // no alternate, use default
    layout = defaultLayout;
}
else
{
    // make it a full(er) path
    layout = layoutDir + layout;
}

Template layout_vm = null;
try
{
    //System.err.println("----- > layout=" + layout);
    //load the layout template
    layout_vm = getTemplate(stack, velocityManager.getVelocityEngine(),
        invocation, layout, inputEncoding);
}
catch (Exception e)
{
    velocityManager.getVelocityEngine().getLog().error(
        "VelocityLayoutResult: Can't load layout \"" + layout + "\": " + e);

    // if it was an alternate layout we couldn't get...
    if (!layout.equals(defaultLayout))
    {
        // try to get the default layout
        // if this also fails, let the exception go
        layout_vm = getTemplate(stack, velocityManager.getVelocityEngine(),
            invocation, defaultLayout, inputEncoding);
    }
}

Writer writer = new OutputStreamWriter(response.getOutputStream(),
    outputEncoding);
response.setContentType(contentType);
layout_vm.merge(context, writer);
writer.flush();
```

总结

本文是在初步使用 Struts 2 时所发现的一些问题，通过对 Struts 2 代码的研究找出相应的解决办法，如果你在使用 Struts 2 和 Velocity 中有发现其他问题，欢迎跟我联系一起探讨。

参考资料

学习

- developerWorks [Struts 框架专题](#)：该专题包含大量和 Struts 框架相关的技术文章和教程。
- [Struts 与 Velocity 的集成](#)（developerWorks，2005 年 10 月）：《Struts Recipes》的合著者 George Franciscus 带您一步步地把 Velocity 模板引擎集成进 Struts 应用程序。
- [Java 技术专区](#)：寻找 Java 编程各方面的技术文章。

获得产品和技术

- [Struts 框架](#)：Apache 软件基金会的一个项目。
- [Velocity 模版引擎](#)：针对 MVC 框架的一个基于 Java 的模板引擎。

关于作者

刘冬，一直使用 J2EE/J2ME 从事移动互联网方面的开发，拥有开源博客产品 DLOG4J，目前 DLOG4J 已经推出3.0多用户版。您可以通过 Java 自由人网站来跟他联系，网址是：<http://www.dlog.cn/javayou>，另外他的邮件地址是 javayou@gmail.com。

建议



[回页首](#)

[打印此页面](#) [分享此页面](#) ▼ [关注 developerWorks](#) ▼

技术主题		查找软件	社区	关于 developerWorks	IBM
AIX and UNIX	Java technology	IBM 产品	群组	反馈意见	解决方案
Information Management	Linux	评估方式（下载，在线试用，Beta 版，云）	博客	在线投稿	软件
Lotus	Open source	行业	Wiki	投稿指南	支持门户
Rational	SOA and web services		文件	网站导航	产品文档
WebSphere	Web development	技术讲座	使用条款与条件	请求转载内容	红皮书 (英语)
	XML		报告滥用	相关资源	隐私条约
	更多...		更多...		浏览辅助
Cloud computing				ISV 资源 (英语)	
				IBM 教育学院教育培养计划	