



FaceLets 自定义组件

百洋医药集团软件研究实验室技术文档

 百洋医药集团有限公司

PEARL OCEAN PHARMACEUTICAL HOLDINGS LIMITED - INFORMATION TECHNOLOGY DEPARTMENT

FaceLets 自定义组件

2009年9月11日

slx



FaceLets 自定义组件

关键词(Keyword): JSF, FaceLets, 自定义组件 自定义控件




JSF 自定义组件是个比较繁琐的事情, 需要扩展 UIComponent 接口, 编写渲染器, Tag 类, 定义 tld 文件等诸多步骤. 总之开发维护成本都比较高. FaceLets 作为构建在 JSF 标准之上的一个视图处理框架, 为我们提供了新的自定义组件方式, 可以便捷的实现组件复用. 这正是 FaceLets 的精华所在.

在我们的 OECF 项目进展到界面开发阶段时, 开发组发现很多类似页面的功能是通用的. 例如弹出窗体参选数据的功能, N 多界面上会使用到, 没有必要每个开发组成员在每个界面上都自己写. 这也就引出了封装自定义组件的需求. 通过查找资料多次试验这个组件终于诞生了. 先展示一下效果, 相信大家就会明白自定义组件的好处了.

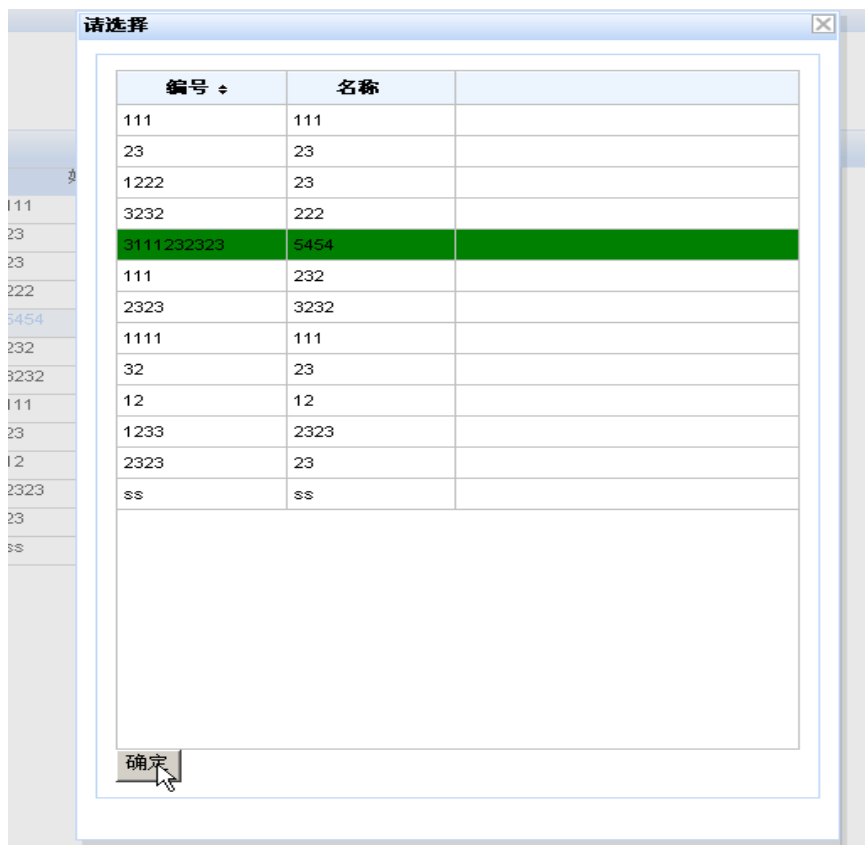
组件完成后开发人员只需要在界面上添加一行代码就可以实现弹出表格参选数据的功能:

```
<oecp:refTablePanle id="refPanl" refModel="#{testAction.refTableModel}"  
selectionMode="multi" />
```

点击需要回填的单元格后面的小图标, 就会弹出数据选择界面.

222	3232 
5454	3111232323 
232	111 

选择数据后点确定按钮, 数据将会回填到界面上



怎么样?一行代码就能搞定,是不是很爽呢?

下面就说一下到底如何创建 **FaceLets 自定义组件**. 上面说的那个组件由于比较复杂,就不在这里说了,大家可以在我们的开源项目--OECF 初版完成后,在她的源代码中看到具体的实现方式. 在这我就随便组合两个控件来介绍一下吧.

要使用自定义组件首先要在 web.xml 文件中增加下面几行:

```
<context-param>
  <param-name>facelets.LIBRARIES</param-name>
  <param-value>/WEB-INF/oeep.taglib.xml</param-value>
</context-param>
```

我们可以看到这几行中有指定一个库文件 oecp.taglib.xml, 该文件命名必须是 xxxx.taglib.xml 的格式. 然后再来看看这个文件的内容:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE facelet-taglib PUBLIC
  "-//Sun Microsystems, Inc.//DTD Facelet Taglib 1.0//EN"
  "http://java.sun.com/dtd/facelet-taglib_1_0.dtd">
```

```
<facelet-taglib>
<namespace>http://www.byyy.com/oecp</namespace>
<tag>
  <tag-name>demo</tag-name>
  <source>../components/demo.xhtml</source>
</tag>
</facelet-taglib>
```

指明命名空间<namespace>http://www.byyy.com/oecp</namespace>

同时指明组件标签和组件源文件.

组合控件的源文件就是这个了 demo.xhtml

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
xmlns:ui="http://java.sun.com/jsf/facelets"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:a4j="http://richfaces.org/a4j"
xmlns:rich="http://richfaces.org/rich">
```

```
<a4j:keepAlive beanName="demoBean"></a4j:keepAlive>
```

```
<rich:panel id="#{id}">
```

```
  <f:facet name="header">
```

```
    <h:outputText value="#{title}" />
```

```
  </f:facet>
```

```
<h:form >
```

```
<a4j:commandButton value="产生随机数" action="#{demoBean.rand}"
  reRender="txt"></a4j:commandButton>
```

```
<h:outputText value="#{demoBean.randtext}" id="txt" />
```

```
</h:form>
```

```
</rich:panel>
```

```
</ui:composition>
```

注意一定要包含在<ui:composition>标签内哦. 还要写一个后台类来支持随机数的生成

```
package org.oecp.web;

import org.jboss.seam.ScopeType;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.annotations.Scope;
import org.oecp.annotation.DefaultActionProxy;

@Name("demoBean")
@Scope(ScopeType.PAGE)
@DefaultActionProxy
public class DemoBean {

    private int randtext ;

    java.util.Random r=new java.util.Random();
    public void rand() {
        setRandtext(r.nextInt());
    }

    public int getRandtext() {
        return randtext;
    }

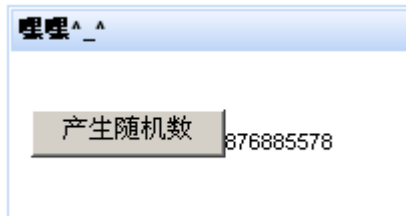
    public void setRandtext(int randtext) {
        this.randtext = randtext;
    }
}
```

```
}
```

在需要使用这个控件的界面中添加一行代码:

```
<oecp:demo id="demo" title="嘿嘿^_^" />
```

我们的**自定义控件**就出现在界面上了



可能有的同学会提出这样的疑问,几个控件放在一起,为什么我不写一个文件,然后 include 进来不就行了吗? Include 确实是一种实现代码复用的方式,不过比起自定义组件来它不能够传递变量.上面的小例子中我们有写到 `title="嘿嘿^_^"` 对应**自定义控件**中的 `<h:outputText value="#{title}" />` 也就是说 `title` 是一个变量,是可以替换的.另外,include 的方式还有一个缺点,就是被 include 进来的页面内的控件 `id` 很可能与包含它的页面内控件 `id` 发生重复,给开发人员命名时带来不必要的麻烦.而**自定义控件**则不会出现这样的问题.

可能大家会发现,我们的**自定义控件**在开发环境中是没有代码提示的.确实是这样,如果我们想要代码提示,就请自己定义一个 `tld` 文件吧.当然这个文件是否存在,以及写的是否正确都不会影响我们的**自定义控件**的运行.`tld` 文件如何定义比较简单,这里就不多说了,随便找个 `tld` 文件仿照一下就可以了.

Facelets 自定义组件的方法还是蛮简单的,看过我的介绍相信大家现在也都学会了吧.

提供该文档的机构为 [百洋软件研究实验室](#),更多的博客文章可以到 [百洋软件研究实验室博客](#) 查看.该文档附件欢迎各位转载,但是在没有获得文章作者许可之前,不得对文章内容或者版权信息进行更改,版权归[百洋软件研究实验室](#)所有,仅此声明.