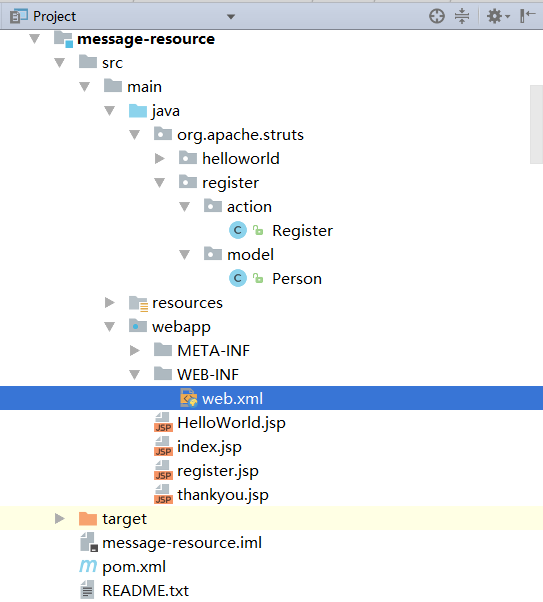
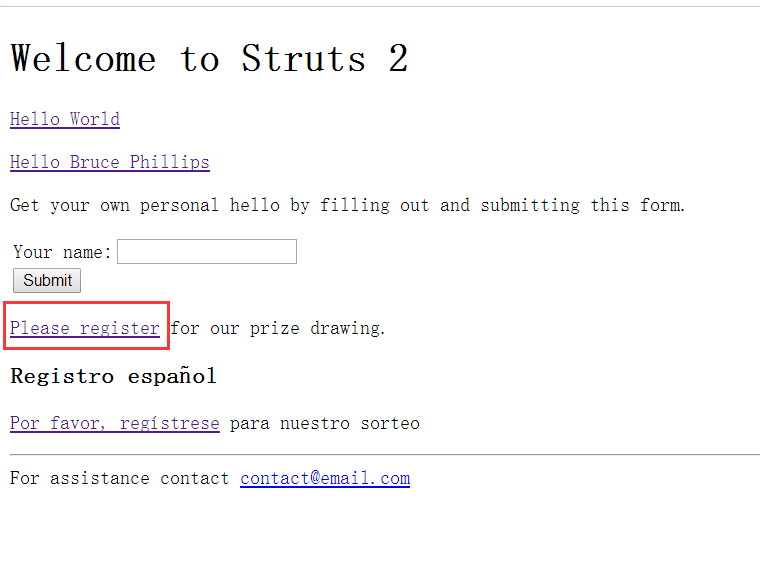
# Struts文本国际化

# 项目包结构：



# 运行效果：



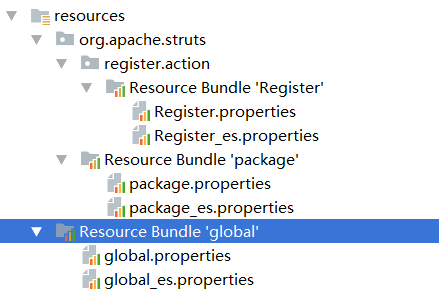


# 代码分析：

该项目介绍的是使用struts中的标签**s:text**

引入i18n文本，达到国际化文本的效果；

## 国际化文本定义方法：

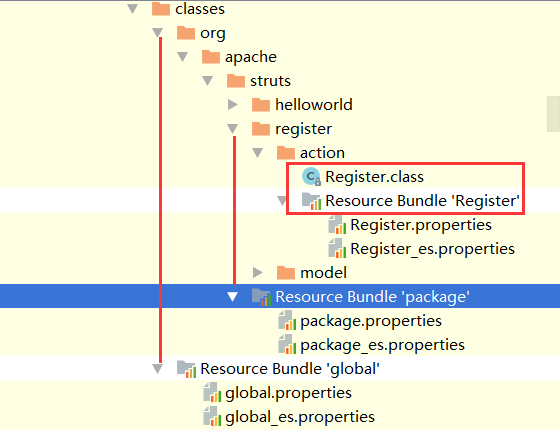


global定义在根目录下，可以猜想是所有的jsp文件都可以访问，定义在global中的值都会放到所有request中去。

package定义在某一个包下（org.apache.struts）,可以猜想该包下所有action都会放置这些值

Register定义在对应action相同的包下，可以猜想这些值只会放到该action的请求下。

编译之后：



参数列表：这里只列出en语言的

**global级别：**

**contact**=**For assistance contact <a href='mailto:contact@email.com'>contact@email.com</a>**

**package级别：**

**greeting**=**Welcome to The Wonderful World of Struts 2  
instructions**=**Register for a prize by completing this form.**

**registeraction级别：**

**personBean.firstName**=**First name  
personBean.lastName**=**Last name  
personBean.age**=**Age  
personBean.email**=**Email  
thankyou**=**Thank you for registering %{personBean.firstName}.  
submit**=**Register**

**依照这个级别，看效果是否对应：**

**index.jsp页面**

<**s:text name="contact"** />引入的是**global中的contact，如果没有定义该值，则显示contract**

**HelloWorld.jsp页面**

<**h1**><**s:text name="greeting"** /></**h1**>引入的是**package中的greeting**

<**s:text name="contact"** />引入的是**global中的contact**

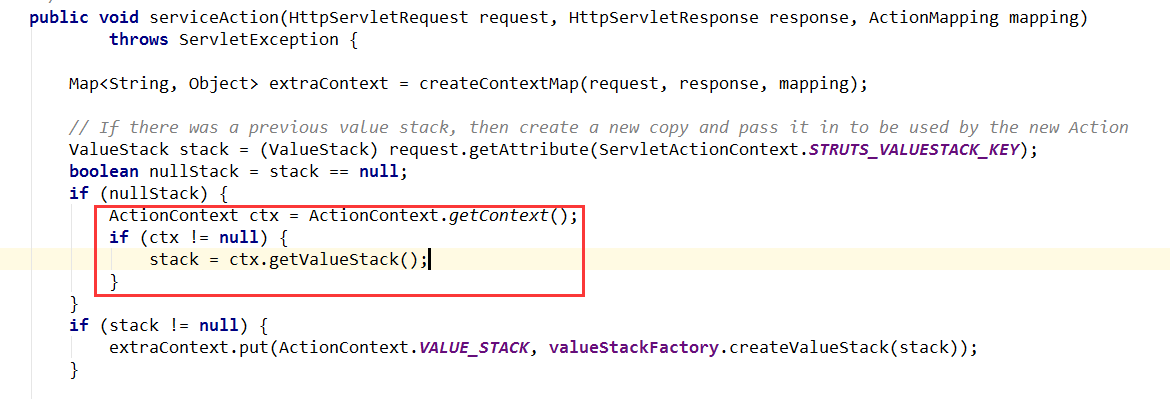
**register.jsp页面**

<**h1**><**s:text name="greeting"** /></**h1**>  
<**h3**><**s:text name="instructions"** /></**h3**>  
  
<**s:form action="register"**>  
  
 <**s:textfield key="personBean.firstName"** />  
 <**s:textfield key="personBean.lastName"** />  
 <**s:textfield key="personBean.email"** />   
 <**s:textfield key="personBean.age"** />  
   
 <**s:submit key="submit"** />  
   
</**s:form**>   
 <**hr** />  
<**s:text name="contact"** />

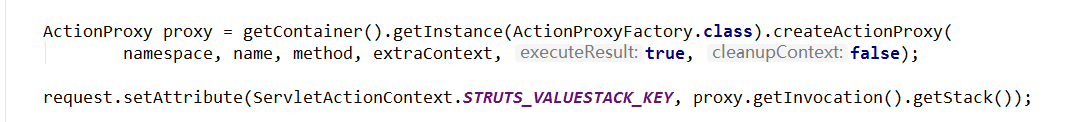
**这些值是如何从资源文件中读取到jsp页面中的呢？**

**猜想下，这些值肯定是读取之后放到request中的，那么是什么时候读取的呢？一定是请求action之前读取的，那么就可能是**StrutsPrepareAndExecuteFilter，找了下代码：





从ActionContext中获取到了值栈ValueStack,放到了request中



JspTag=>Tag=>IterationTag=>BodyTag

TagSupport=>IterationTag

**class** BodyTagSupport **extends** TagSupport **implements** BodyTag

StrutsBodyTagSupport **extends** BodyTagSupport{

**protected** ValueStack getStack() {  
 **return** TagUtils.*getStack*(**pageContext**);  
}

}

TagUtils来读取值栈