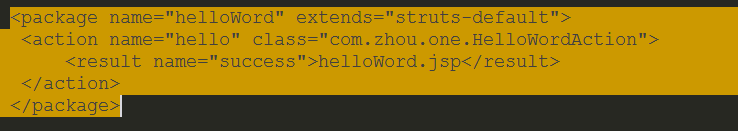
**使用Struts2前准备：**

把核心包拉进项目lib文件夹下，创建struts.xml文件

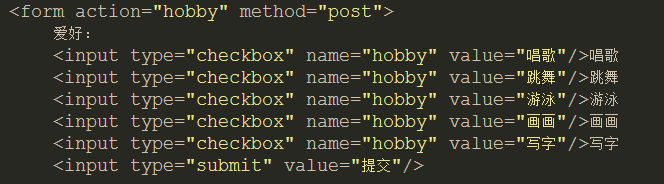
Helloword

配置Struts.xml



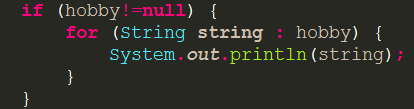
默认执行的是action中的execute方法，而且action一般继承ActionSupport,然后重写默认的execute方法

对于同一类型的，例如

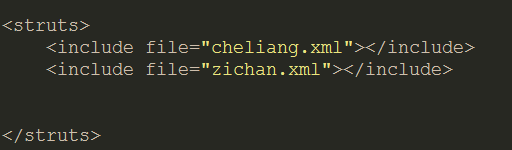


可以在action中使用数组或其他集合接收，例如





可以在Struts.xml中使用如下方法，引入其他的action的xml配置文件，使得一个action对应一个xml



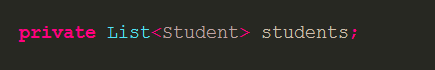
Package标签下

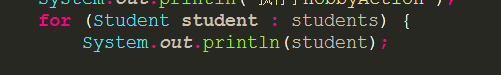


当使用了namespace属性后，在用URL访问时就得在项目名称后面跟上namespace的属性值

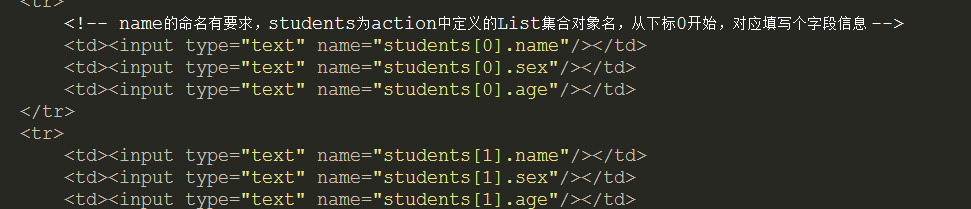


当要使用批量添加一个个体类时可以在action中使用集合，例如使用一个list





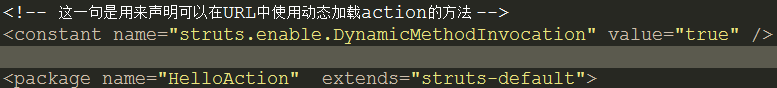
需要注意的是在获取前台数据时name的设置



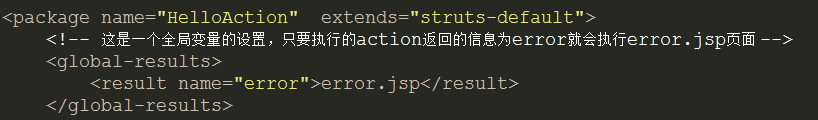
当在action中使用了method属性时，就会执行method值的方法，该方法一定要在action中自己定义有



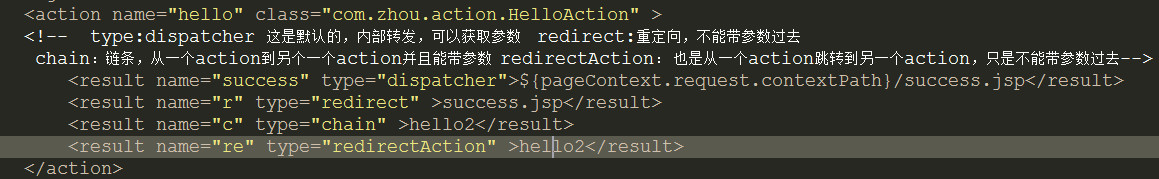
**动态的在URL声明要执行的action中的方法，先要在Struts中先定义配置信息如下**



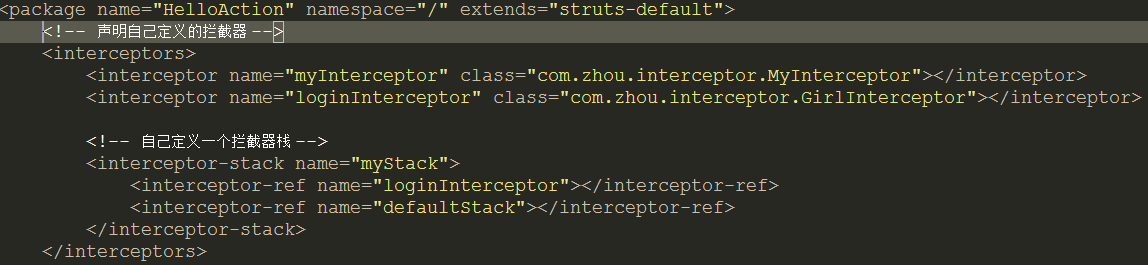
**全局变量的使用，例如当访问action方法返回的操作码是error,都会执行error.jsp**



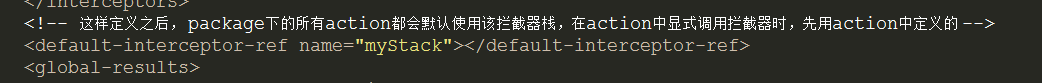
**result中type属性的配置，例如：**



**拦截器与拦截器栈的使用**

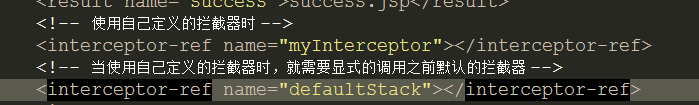


**如果加了一下一句话，就会使得action默认使用自定义的拦截器栈**

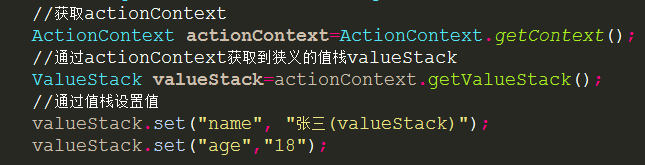


**当如果不想使用自定义的拦截器栈时可以去掉该语句，或者在**

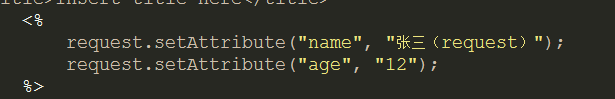
**action中显式调用自己想调用的拦截器栈**

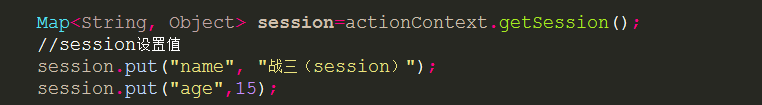


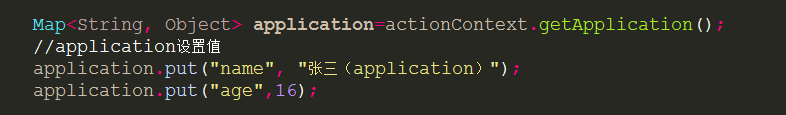
**在action中设置各种值的方法：**

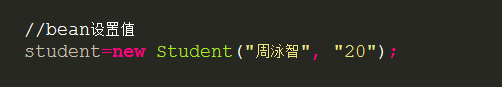


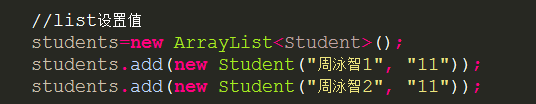
**Request设置值可以直接在jsp页面设置**

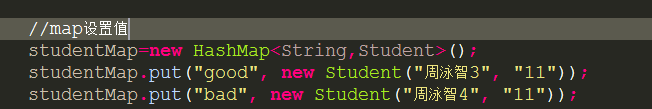




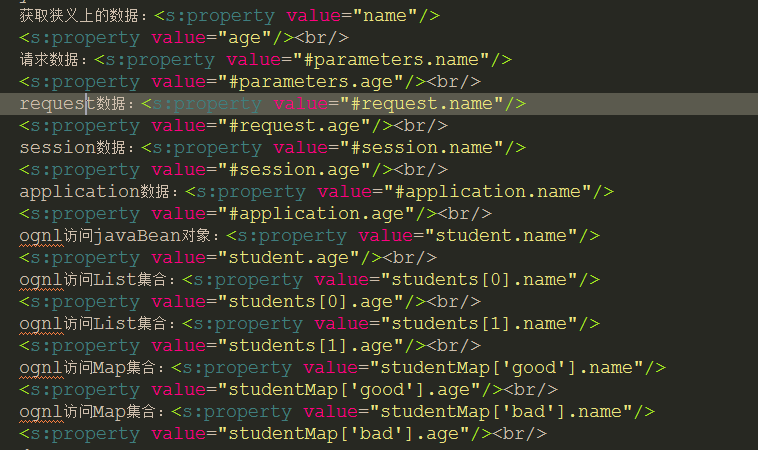








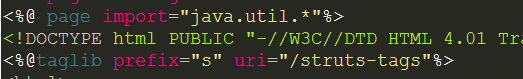
**在页面中获取action设置的各种值的方法：**

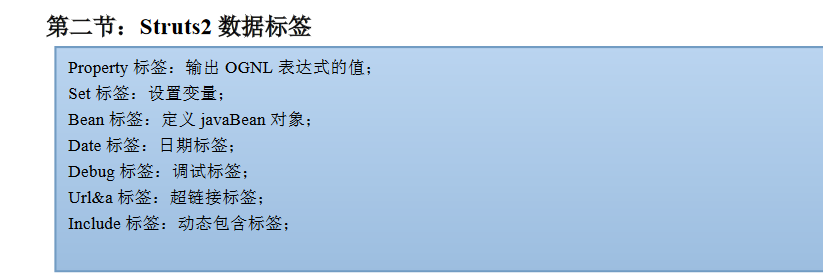


**要访问静态方法时需要在Struts.xml中配置下列信息**

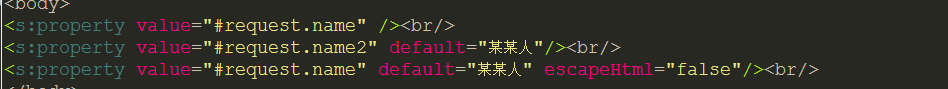


**Struts2的标签性能上会比JSTL标签弱，使用Struts2标签之前一定要先声明引入：**

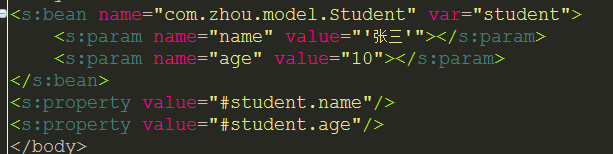




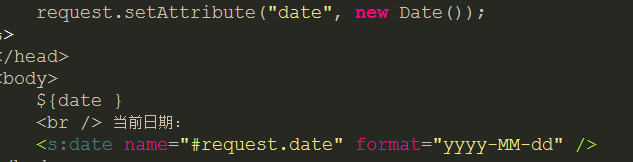
**使用property标签时，default属性为value取不到值时显示的数据，escapeHtml属性为设置是否解析html标签，FALSE为解析**



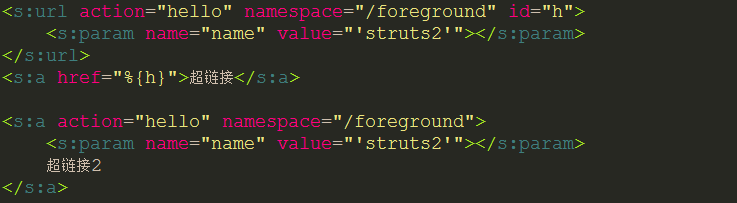
**使用bean标签可以先 在页面定义一个bean，再使用取值**



**使用date标签：**



**超链接标签的两种用法**

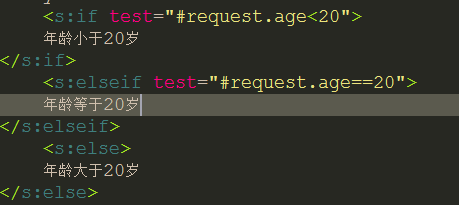


**Include标签的使用**

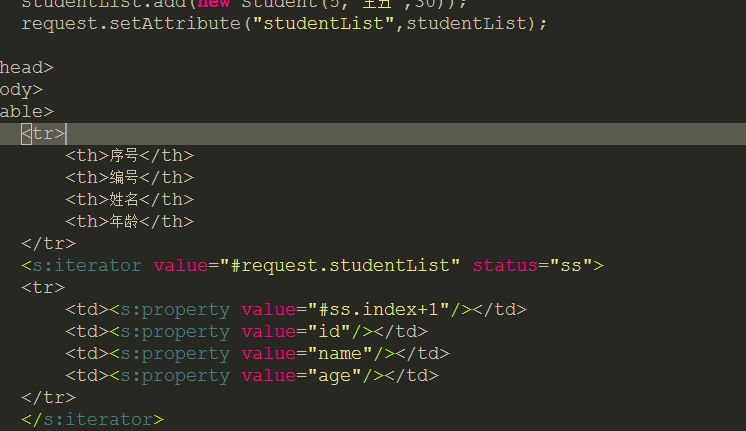




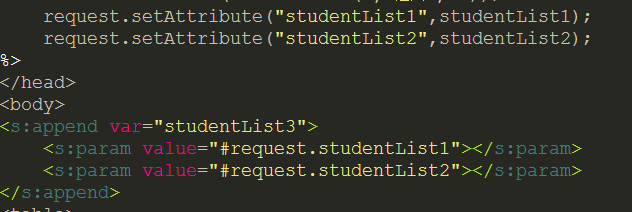
**If-else标签的使用，test属性为判断条件。例如**



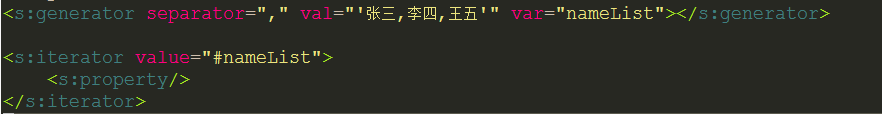
**Iterator的使用，status属性相当于一个引用**



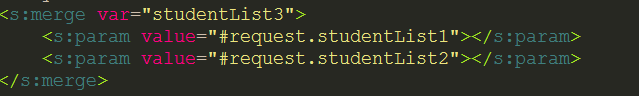
**Append标签，例如：把两个list添加到一个var变量studentList3**



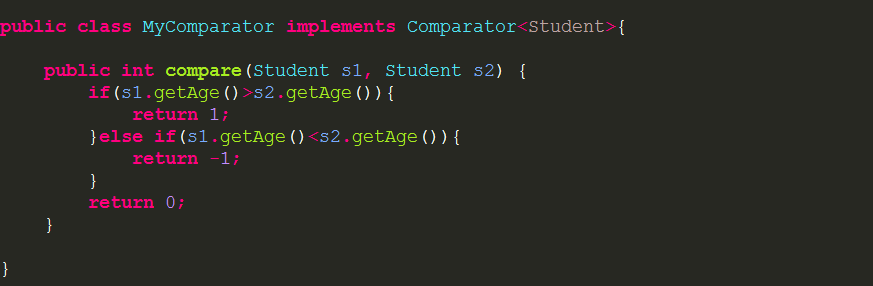
**generator标签，例如把val属性值按分隔符分开存进变量var中**



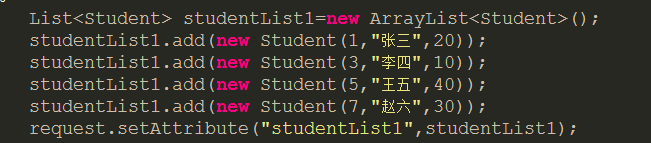
**Merge混合标签，结果是先添加一个studentList1的元素到studentList3，然后一个studentList2的元素，重复往返**

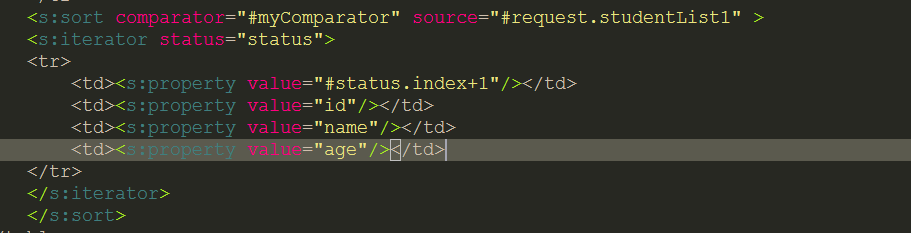


**Sort排序标签，先定义一个bean，然后在页面引用这个bean，然后再Sort标签的source的属性中写入要排序的list，以及在comparator属性中写入定义的bean，就可以使用迭代器输出显示**





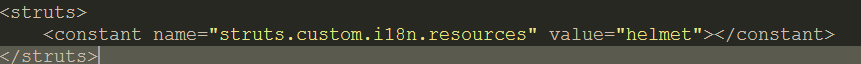


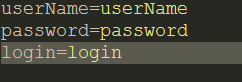
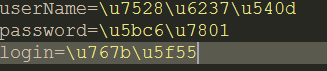
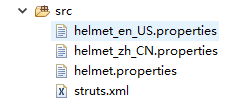


**Subset截取标签，例如把list从索引为2开始，截取长度为2**



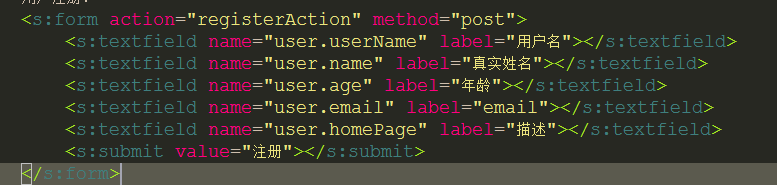
**国际化的使用，现在Struts.xml文件中定义，value值为一个属性文件，然后再另外添加一个代表中文的属性文件，一个代表英文的属性文件，最后再页面中通过标签去属性文件的值**



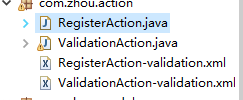


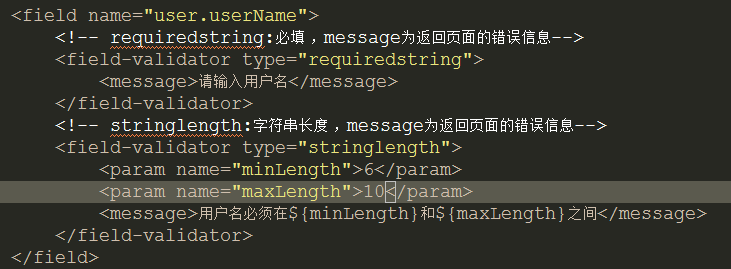


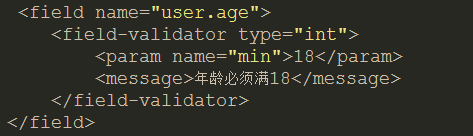
**Struts自带验证器的使用：**在jsp页面要使用Struts定义的form标签，textfiled为model定义的字段的名称，用户输入的值会自动调用model的set方法，把label相当于value值，显示在页面。

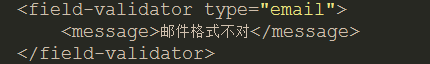


然后要定义一个actionName-validation.xml文件，当用户输入之后就会先到该文件之下去匹配是否符合要求，一些配置如下







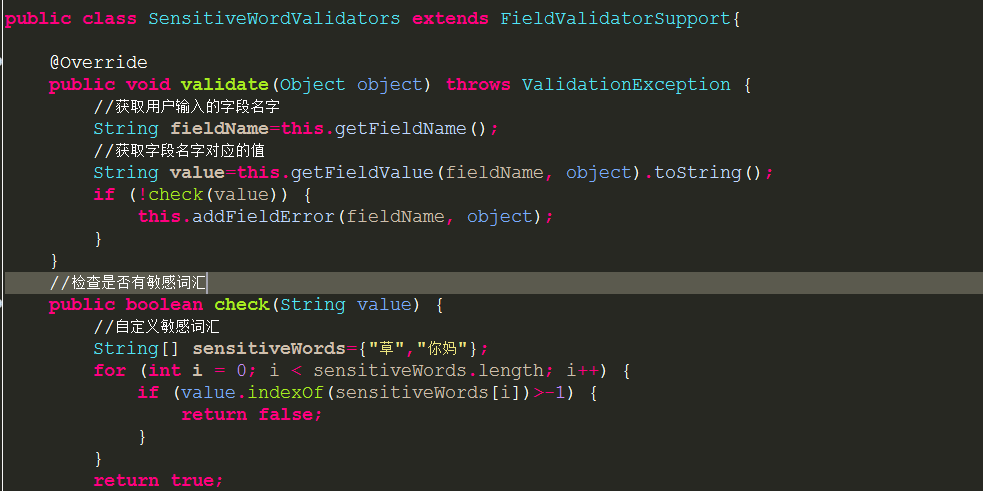


用来在页面显示输入字段的错误信息

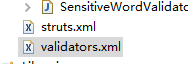


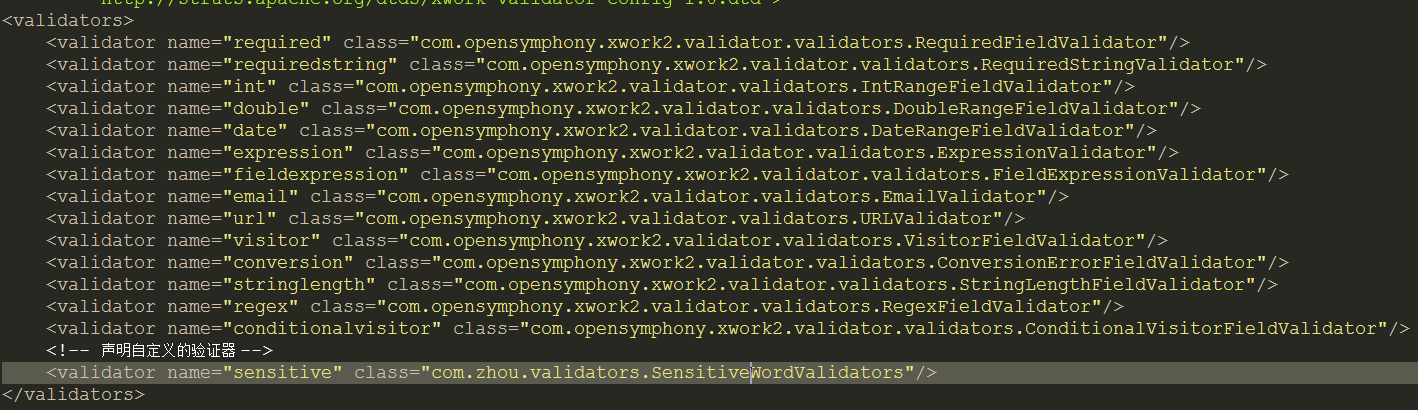
**自定义拦截器件，**

首先定义一个java类继承，并重写相应的方法

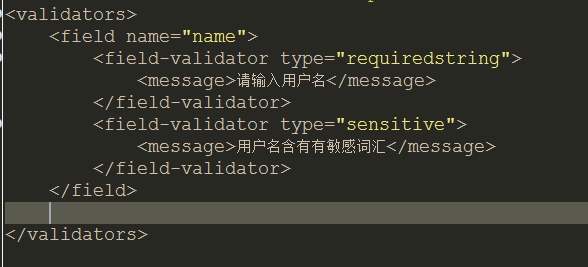


然后配置声明自定义的拦截器件

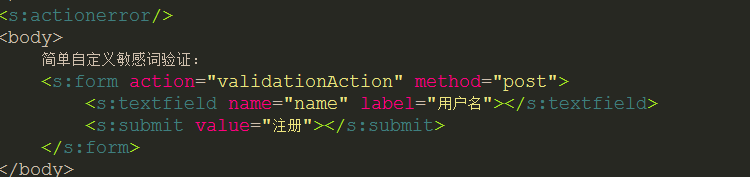




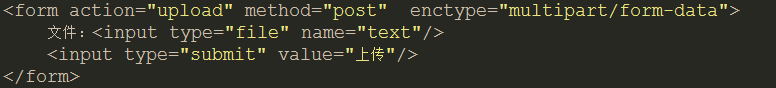
然后再actionName-validation.xml文件中调用



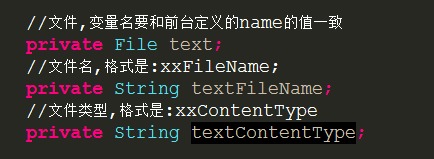
Jsp如下：再写相应action

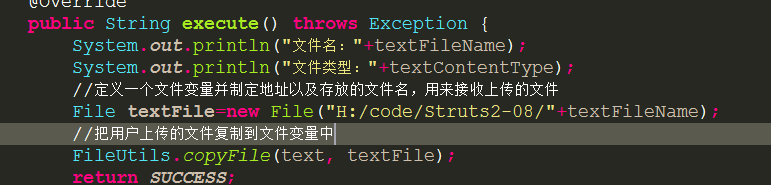


**Struts文件上传，例如jsp页面格式：**



Action中属性的定义及注意

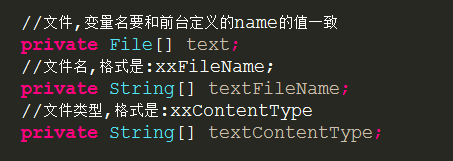


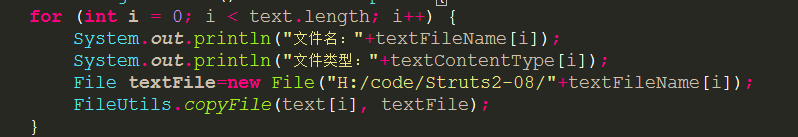


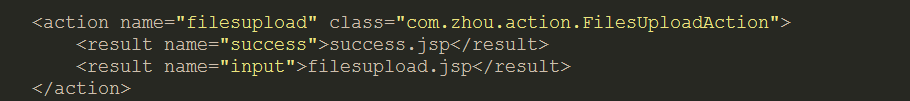
Struts.xml的配置，以及自定义上传的限制



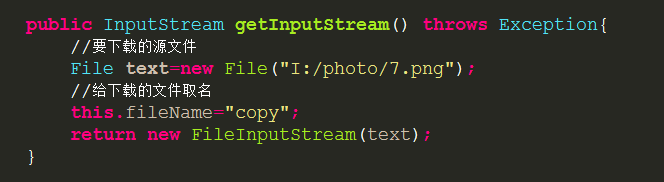
**多文件上传，主要是在后台改为用数组接收**

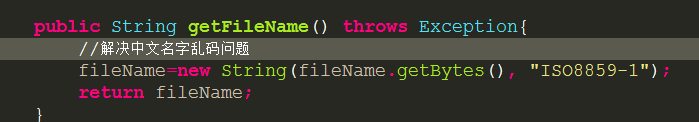




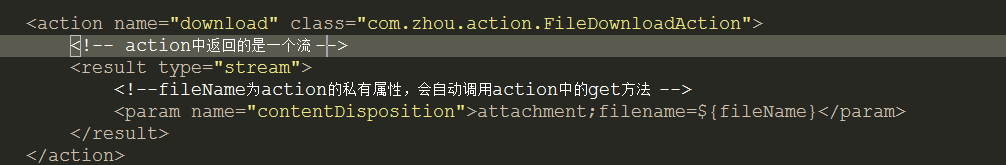


**文件下载，在fileName的get方法中要注意使用转码，不然容易出现中午乱码**





而在配置文件中要注意的是action中返回的是一个stream



**Struts防止反复提交**

