struts 和 servlet 上传文件用法

Servlet:

首先说一下 servlet 上传文件，由于大家对这个相对熟悉一点，所以这个就简要说明一下好了，有什么疑问可以参考之前实验室的代码。

Servlet 上传文件是使用的 smartUpload.jar 这个外部插件，它提供了很好的接口。这个很好下载，而且暂时没有发现有版本的限制，所以放心使用。

如果要上传文件，前台的<input>的属性需要修改，需要加上 entype=”multipart/form-data” 同时后台需要使用 smartUpload 提供的 request 来进行获取，否则会获取不到东西。下边我把后台的部分代码放上，以供参考：

```
//初始化
SmartUpload mySmartUpload=new SmartUpload();
mySmartUpload.initialize(config, request, response);
mySmartUpload.setMaxFileSize(1024*2048);
try
{
    mySmartUpload.upload();
    for (int i = 0; i < mySmartUpload.getFiles().getCount(); i++)
    {
        com.jspsmart.upload.File file =
mySmartUpload.getFiles().getFile(i);
        if (file.isMissing())
            continue;
        file.saveAs(filepath+"/test_file"+i+”.txt”);
    }
}catch(Exception err){}
```

Struts:

Struts相对来说比较麻烦一些，因为需要配置一些文件，然后配置好之后，就相对简

单了，因为这个写jsp后台跟写正常的java代码一样了。

首先讲一下struts文件，需要下载他的jar文件，但是测试了一下，最新的版本貌似不支持，总是出错误，但是2.0版本可以，这里都是使用2.0这个版本的。

其次，是配置文件，首先是 web.xml 文档，需要加入一个过滤器，因为 struts2 的入口是一个过滤器。配置 web.xml 的代码为：

```
<filter>
    <filter-name>struts2</filter-name>
    <filter-class>
        org.apache.struts2.dispatcher.FilterDispatcher
    </filter-class>
  </filter>
  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

然后就是 struts.xml 文档，这个是用来指导后台代码的文档。我截取一段代码进行说明一下：

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
    <package name="struts2" extends="struts-default">
        <action name="myupload" class="action.UploadAction">
            <result name="success">/result.jsp</result>
</action>
</package>
</struts>
```

其中每一个类都是一个 Action，name 是别名，class 指向具体的类，<result..> 是那个类返回的值，name 是返回的参数，要和后台返回的严格对应好，/result.jsp 是转向的 jsp 页面。

配置完毕，那下边就是上传文件了，有两种方法可以进行上传。这里我把代码放上，以供参考。

第一种：

这一种方法比较简单，struts2 提供了很好的接口，直接一句代码

```
private File upload;
```

```
    private javax.servlet.http.HttpServletRequest request;
    public String uploadFile()throws Exception
    {
        String
path=request.getSession().getServletContext().getRealPath("");
        System.out.println(path);
        File target=new File(path+"/test.txt");
        try {
            FileUtils.copyFile(upload, target);
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        request.setAttribute("result", "上传成功!");
        return "up";
    }
    public File getUpload() {
        return upload;
    }
    public void setUpload(File upload) {
        this.upload = upload;
    }
    @Override
    public void setServletRequest(HttpServletRequest req)
    {
        // TODO Auto-generated method stub
        this.request=req;
    }
```

这个类对应的 struts.xml 文档的内容为：

```
<action name="upfile" class="action.ReceiveAction">
    <result name="up">/result.jsp</result>
    <interceptor-ref name="fileUpload">
    <param name="allowedTypes">text/plain</param>
    <param name="maximumSize">4096</param>
</interceptor-ref>
<interceptor-ref name="defaultStack"></interceptor-ref>
</action>
```

<interceptor>是匹配器<param>是参数，提供允许上传的类型和大小，如果不匹配，就会出错。

第二种：

```
private java.util.List<File> uploads;
    private java.util.List<String> fileNames;
```

```
    private java.util.List<String> uploadContentTypes;
    public java.util.List<String> getUploadFileName()
    {
        return fileNames;
    }
    public void setUploadFileName(java.util.List<String> fileNames)
    {
        this.fileNames = fileNames;
    }
    public java.util.List<File> getUpload()
    {
        return uploads;
    }
    public void setUpload(java.util.List<File> uploads)
    {
        this.uploads = uploads;
    }
    public void setUploadContentType(java.util.List<String>
contentTypes)
    {
        this.uploadContentTypes = contentTypes;
    }
    public java.util.List<String> getUploadContentType()
    {
        return this.uploadContentTypes;
    }
    public String execute() throws Exception
    {
        if (uploads != null)
        {
            for (int i=0; i < uploads.size(); i++)
            {
                InputStream is = new FileInputStream(uploads.get(i));
                OutputStream os = new FileOutputStream("F:/test/upload/"
+ fileNames.get(i));
                byte buffer[] = new byte[8192];
                int count = 0;
                while ((count = is.read(buffer)) > 0)
                {
                    os.write(buffer, 0, count);
                }
                os.close();
                is.close();
```

```
            }
        }
        return SUCCESS;
    }
```

这个相对比较麻烦，是使用最常用的 InputStream 和 OutStream 来写入的。这个类对应的 struts.xml 文档内容为：

```
<action name="myupload" class="action.UploadAction">
    <result name="success">/result.jsp</result>
</action>
```

这个里边没有添加类型匹配的，也可以加，和上边的一样。