

1. 上传:

a. 引入2个jar包

apache: commons-fileupload.jar组件

commons-fileupload.jar依赖于commons-io.jar

b.

代码:

前台JSP:

```
<input type="file" name="spicture"/><br/>
```

表单上传方式必须是post, 因为用get会把信息放到地址栏上面, 容量太小

在表单中必须增加一个属性 enctype="multipart/form-data"//如果没有这个字段后台就没有办法解析这个文件

后台Servlet:

注意问题:

上传目录 upload

1. 如果修改代码则在tomcat重新启动的时候被删除

原因: 当修改代码的时候, tomcat会重新编译一份class并且重新部署(重新创建各种目录)

2. 如果不修改代码, 则不会被删除

原因: 1. 没有修改代码, class仍然是之前的class,
因此为了防止上传目录丢失: a. 虚拟路径 b. 直接更换上传目录到非tomcat目录

限制上传:

类型、

大小

注意 对文件的限制条件写在parseRequest之前

二、文件下载

1. 不需要依赖任何jar包

2. 请求(地址 超链接或者表单)

1、请求Servlet 2、Servlet通过文件地址将文件转为输入流, 读到Servlet中

3、通过输出流将刚才已经转为输入流的文件输出给用户

注意下载文件需要设置两个响应头

```
response.setHeader("content-Type", "application/octet-stream");//123类型为二进制（任意文件）
```

```
response.setHeader("content-  
Disposition","attachment;filename="+fileName);//包含了文件后缀
```

3. 下载时，文件名乱码问题：

edge:

```
URLEncoder.encode(fileName,"utf-8")
```

firefox:

给文件名加：

前缀 =?UTF-8?B?

中间：String构造方法

```
Base64.encode
```

后缀

```
response.setHeader("content-Disposition","attachment;filename=?  
UTF-8?B?" + new String(  
org.apache.tomcat.util.codec.binary.Base64.encodeBase64(fileName.getBytes("utf-  
8")))+"?=");//包含了文件后缀
```

