一、准备

　先准备两样东西：两个 jar 包、一个页面。

1、两个 jar 包

　　commons-fileupload.jar：这是一个核心 jar包，实现文件上传的核心类及方法都在这里面。

　　commons-io.jar：这是一个依赖 jar包，是为了辅助 commons-fileupload 包，此包主要是进行 IO操作的，是 commons-fileupload 的依赖包。

2、一个页面

　　准备一个 jsp页面，此页面中要有一个 form表单，此表单的特征如下：

表单的 method 属性值必须是 post；

表单的 enctype 属性值必须是 multipart/form-data；

表单中必须提供至少一个 <input type="file"> 这样的文件上传项。

二、展示

我们需要建立一个 Servlet 用来处理 表单中的数据，在表单中将 type属性值为file 的数据在以下代码中称为“文件数据项”，其他的称为“普通数据项”。在此 Servlet 中需要涉及到三个类，分别是 DiskFileItemFactory类、ServletFileUpload类和 FileItem类。

　　DiskFileitemFactory类负责管理磁盘文件，ServletFileUpload类负责上传和解析文件，FileItem类负责保存每个表单数据项的信息。

1、Servlet 具体代码的实现：

@Override

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// 判断是普通表单，还是文件上传表单

if (!ServletFileUpload.isMultipartContent(request)) {

throw new RuntimeException("不是文件上传表单！");

}

// 创建上传所需要的两个对象

DiskFileItemFactory factory = new DiskFileItemFactory(); // 磁盘文件对象

ServletFileUpload sfu = new ServletFileUpload(factory); // 文件上传对象

// 设置解析文件上传中的文件名的编码格式

sfu.setHeaderEncoding("utf-8");

// 创建 list容器用来保存 表单中的所有数据信息

List<FileItem> items = new ArrayList<FileItem>();

// 将表单中的所有数据信息放入 list容器中

try {

items = sfu.parseRequest(request);

} catch (FileUploadException e) {

e.printStackTrace();

}

// 遍历 list容器，处理 每个数据项 中的信息

for (FileItem item : items) {

// 判断是否是普通项

if (item.isFormField()) {

// 处理 普通数据项 信息

handleFormField(item);

} else {

// 处理 文件数据项 信息

handleFileField(item);

}

}

}

2、两个处理“数据项”信息的方法

　　一个用来处理“普通数据项”信息的，另一个是用来处理“文件数据项”信息的。

　　1）处理“普通数据项”信息的方法

/\*\*

\* 处理 普通数据项

\* @param item

\*/

private void handleFormField(FileItem item) {

// 获取 普通数据项中的 name值

String fieldName = item.getFieldName();

// 获取 普通数据项中的 value值

String value = "";

try {

value = item.getString("utf-8"); // 以 utf-8的编码格式来解析 value值

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

}

// 输出到控制台

System.out.println("fieldName:" + fieldName + "--value:" + value);

}

　　2）处理“文件数据项”信息的方法

/\*\*

\* 处理 文件数据项

\* @param item

\*/

private void handleFileField(FileItem item) {

// 获取 文件数据项中的 文件名

String fileName = item.getName();

// 判断 此文件的文件名是否合法

if (fileName==null || "".equals(fileName)) {

return;

}

// 控制只能上传图片

if (!item.getContentType().startsWith("image")) {

return;

}

// 将文件信息 输出到控制台

System.out.println("fileName:" + fileName); // 文件名

System.out.println("fileSize:" + item.getSize()); // 文件大小

// 获取 当前项目下的 /files 目录的绝对位置

String path = this.getServletContext().getRealPath("/files");

File file = new File(path); // 创建 file对象

// 创建 /files 目录

if (!file.exists()) {

file.mkdir();

}

// 将文件保存到服务器上（UUID是通用唯一标识码，不用担心会有重复的名字出现）

try {

item.write(new File(file.toString(), UUID.randomUUID()+"\_"+fileName));

} catch (Exception e) {

e.printStackTrace();

}

}

三、扩展

1、文件上传的常见限制要求

　　1）限制上传文件类别。如：只能上传图片

// 控制只能上传图片

if (!item.getContentType().startsWith("image")) {

return;

}

　　2）限制上传文件大小。如：单个文件大小不能超过 10KB，总文件大小不能超过 100KB

// 创建上传所需要的两个对象

DiskFileItemFactory factory = new DiskFileItemFactory(); // 磁盘文件对象

ServletFileUpload sfu = new ServletFileUpload(factory); // 文件上传对象

// 限制单个文件的大小

sfu.setFileSizeMax(1024\*10);

// 限制上传的总文件大小

sfu.setSizeMax(1024\*100);

// 创建 list容器用来保存 表单中的所有数据信息

List<FileItem> items = new ArrayList<FileItem>();

// 将表单中的所有数据信息放入 list容器中

try {

items = sfu.parseRequest(req);

} catch (FileUploadException e) {

System.out.println("单个文件大小不能超过10KB，总文件大小不能超过100KB");

e.printStackTrace();

}

2、common-fileupload 包中三大核心类

　　1）DiskFileItemFactory类

作用：磁盘文件工厂类，它还可以设置缓冲区大小以及设置临时保存位置。

setSizeThreshold( int sizeThreshold )

// 设置缓冲区大小，默认sizeThreshold的大小为：10240（10KB）。

setRepository( File repository );

// 设置临时文件的保存位置，如果不设置，repository为系统的临时目录。

DiskFileItemFactory();

// 构造方法，缓冲区大小为默认sizeThreshold和临时文件为目录为默认repository的磁盘文件工厂类。

DiskFileItemFactory( int SizeThreshold , File repository );

// 构造方法，指定缓冲区大小和指定临时文件的磁盘文件工厂类。12345678

　　2）ServletFileUpload类

作用：文件上传类，实现上传的一些实用方法的集合。

public List<FileItem> parseRequest( HttpSevletRequest request );

// 解析request对象，获取表单中的所有数据信息，并返回一个List<FileItem>集合，

// 其中每个FileItem就是一个表单数据项信息。

boolean isMultipartContext( HttpServletRequest request );

// 根据请求对象来判断是普通表单，还是文件上传表单，如果是文件上传表单，则返回true，否则false。

setFileSizeMax( long fileSizeMax );

// 设置单个文件的上传的大小上限。

setSizeMax( long sizeMax );

// 设置总文件上传的大小上限。

setHeaderEncoding( Charset charset);

// 解决上传文件中的文件名中文乱码的问题。

ServletFileUpload( DiskFileItemFactory factory );

// 构造方法，指定磁盘文件工厂类，并使用factory指定的缓冲区大小和临时文件。

　　3）FileItem类

作用：表单数据项类，保存了表单数据项的所有信息，以下是完整的 FileItem类对象信息。

普通数据项： name=null, StoreLocation=XXX.tmp, size=5 bytes, isFormField=true, FieldName=username

文件数据项： name=要要.jpg, StoreLocation=XXX.tmp, size=8739 bytes, isFormField=false, FieldName=file2

isFormField();

// 判断是否为普通数据项。

String getFieldName();

// 获取该表单数据项的名称。即：<input>标签中的name属性。

String getString( String encoding );

// 以指定的编码格式解析该表单数据项的值。即：<input>标签中的value属性。

String getName();

// 获取上传文件中的文件名。

getInputStream()；

// 获取上传文件的内容的输入流，使用文件复制就能完成文件的上传。

delete();

// 删除临时文件。

3、解决中文乱码问题

　　1）上传的文件名乱码

　　使用 ServletFileUpload类 中的 setHeaderEncoding( String encoding ) 方法。

　　2）普通表单数据项的内容乱码

　　使用 FileItem 中的 getString( String encoding ) 方法。