**文件上传**

**第一种上传方式（操作简单实用）：SmartUpload上传组件**

需要将SmartUpload的jar包放到Tomcot的安装目录中的lib文件夹中

如果要使用SmartUpload上传文件，则必须按照如下步骤进行：

1. 实例化SmartUpload对象

Eg:SmartUpload smart = new SmartUpload();

1. 初始化上传操作

Eg：smart.initialize(pageContext);

1. 准备上传

Eg:smart.upload();

1. 保存文件

Eg:smart.save(“upload”);

范例：混合表单的文件上传

<?xml version=*'1.0'* encoding=*'GBK'* ?>

<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Strict//EN' 'http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd'>

<html xmlns=*'http://www.w3.org/1999/xhtml'*>

<head>

<meta http-equiv=*'Content-Type'* content=*'text/html; charset=GBK'* />

<title>接收封装表单的文本数据</title>

</head>

<body>

<form action = *"smartupload\_demo02.jsp"* method = *"post"* enctype = *"multipart/form-data"*>

姓名：<input type = *"text"* name = *"uname"*></input><br></br>

照片：<input type = *"file"* name = *"pic"* ></input><br></br>

<input type = *"submit"* value = *"上传"*></input>

<input type = *"reset"* value = *"取消"*></input>

</form>

</body>

</html>

<?xml version=*"1.0"* encoding=*"GBK"* ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import=*"unit\_8.IPTimeStamp"*%>

<%@page import=*"org.lxh.smart.SmartUpload"*%>

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title>接收封装表单的文本数据,增加自动命名功能</title>

</head>

<body>

<% //request内置对象无法取得表单提交过来的参数，必须通过SmartUpload类中提供的getRequest()方法取得全部参数

request.setCharacterEncoding("GBK");

%>

<%

SmartUpload smart = **new** SmartUpload(); //实例化SmartUpload上传组件

smart.initialize(pageContext); //初始化上传操作

smart.upload(); //上传准备

String name = smart.getRequest().getParameter("uname"); //接收请求参数

IPTimeStamp its = **new** IPTimeStamp(request.getRemoteAddr()); //取得ip地址实例化对象

String ext = smart.getFiles().getFile(0).getFileExt(); //取得文件后缀

String fileName = its.getIPTimeRandom() + "." + ext; //拼凑文件名称

/\* smart.getFiles().getFile(0).saveAs(getServletContext().getRealPath("/") +

"upload" + java.io.File.separator + fileName); //保存文件 ？？？？？？？ \*/

smart.getFiles().getFile(0).saveAs( "//upload");

// smart.save("upload");

%>

<h2>姓名：<%=name %></h2>

<img src=*"../upload"*<%=fileName %>></img>

</body>

</html>

表单中使用了file控制进行文件的选择，而且在form上使用enctype进行了表单封装，表示表单将按照二进制方式提交，即所有的操作表单此时不再是分别提交而是将所有的内容都按照二进制方式提交。所以此时单纯靠request对象是无法取得提交参数的，必须依靠SmartUpload类中的getRequest().getParameter()方法才能取得请求的参数。

由于现在是通过SmartUpload完成参数接收，所以smart.getRequest()方法一定要在执行完upload()方法后才可以用。

1. 为上传文件自动命名（使用ip + 时间戳 + 三位随机数的方式作为文件名）

范例：

package unit\_8;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Random;

/\*\*

\* 定义一个取得ip+时间戳+随机数的类

\* 该类的作用是用来自动生成上传的文件名，以防止重名覆盖的情况

\* \*/

public class IPTimeStamp {

private SimpleDateFormat sdf = null; //定义SimpleDateFormat对象

private String ip = null ;

public IPTimeStamp(String ip){ //接收ip地址

this.ip = ip ;

}

public String getIPTimeRandom(){ //得到ip地址+时间戳+三位随机数

StringBuffer buf = new StringBuffer(); //实例化StringBuffer对象

if(this.ip != null){

String s[] = this.ip.split("\\."); //拆分ip

for(int i = 0;i <s.length;i++){

buf.append(this.addZero(s[i], 3)); //不够三位的要补0

}

}

buf.append(this.getTimeStamp()); //取得时间戳

Random r = new Random(); //定义Random对象，产生随机数

for(int i= 0;i < 3;i++){ //循环三次

buf.append(r.nextInt(10)); //增加一个随机数

}

return buf.toString();

}

private String addZero(String str,int len){ //补0操作

StringBuffer buf = new StringBuffer();

buf.append(str);

if(str.length() < len){

buf.insert(0, "0"); //若不够三位则在前面补0

}

return buf.toString();

}

public String getDate(){ //取得当前系统时间

this.sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");

return this.sdf.format(new Date());

}

public String getTimeStamp(){ //取得时间戳

this.sdf = new SimpleDateFormat("yyyyMMddHHmmssSSS");

return this.sdf.format(new Date());

}

}

<?xml version=*"1.0"* encoding=*"GBK"* ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import=*"unit\_8.IPTimeStamp"*%>

<%@page import=*"org.lxh.smart.SmartUpload"*%>

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title>接收封装表单的文本数据,增加自动命名功能</title>

</head>

<body>

<% //request内置对象无法取得表单提交过来的参数，必须通过SmartUpload类中提供的getRequest()方法取得全部参数

request.setCharacterEncoding("GBK");

%>

<%

SmartUpload smart = **new** SmartUpload(); //实例化SmartUpload上传组件

smart.initialize(pageContext); //初始化上传操作

smart.upload(); //上传准备

String name = smart.getRequest().getParameter("uname"); //接收请求参数

IPTimeStamp its = **new** IPTimeStamp(request.getRemoteAddr()); //取得ip地址实例化对象

String ext = smart.getFiles().getFile(0).getFileExt(); //取得文件后缀

String fileName = its.getIPTimeRandom() + "." + ext; //拼凑文件名称

/\* smart.getFiles().getFile(0).saveAs(getServletContext().getRealPath("/") +

"upload" + java.io.File.separator + fileName); //保存文件 ？？？？？？？ \*/

smart.getFiles().getFile(0).saveAs( "//upload");

// smart.save("upload");

%>

<h2>姓名：<%=name %></h2>

<img src=*"../upload"*<%=fileName %>></img>

</body>

</html>

1. 批量上传

在以上程序中可以发现，有 smart.getFiles().getFile(0).saveAs(getServletContext().getRealPath("/") +

"upload" + java.io.File.separator + fileName); //保存文件 这样一个语句，也就证明一次可以上传多个文件的操作

范例：批量上传

<?xml version=*'1.0'* encoding=*'GBK'* ?>

<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Strict//EN' 'http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd'>

<html xmlns=*'http://www.w3.org/1999/xhtml'*>

<head>

<meta http-equiv=*'Content-Type'* content=*'text/html; charset=GBK'* />

<title>批量上传</title>

</head>

<body>

<form action = *"smartupload\_demo03.jsp"* method = *"post"* enctype = *"multipart/form-data"*>

照片1：<input type = *"file"* name = *"pic1"*></input><br></br>

照片2：<input type = *"file"* name = *"pic2"*></input><br></br>

照片3：<input type = *"file"* name = *"pic3"*></input><br></br>

<input type = *"submit"* value = *"上传"*></input>

<input type = *"reset"* value = *"取消"*></input>

</form>

</body>

</html>

<?xml version=*"1.0"* encoding=*"GBK"* ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<%@ page import = *"org.lxh.smart.\*"* %>

<%@ page import = *"unit\_8.IPTimeStamp"* %>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title>批量上传</title>

</head>

<%request.setCharacterEncoding("GBK"); %>

<body>

<%

SmartUpload smart = **new** SmartUpload() ; //实例化SmartUpload上传组件

smart.initialize(pageContext) ; //初始化上传操作

smart.upload() ; //上传准备

IPTimeStamp its = **new** IPTimeStamp(request.getRemoteAddr()) ; //实例化IPTimeStamp对象

**for**(**int** i = 0;i < smart.getFiles().getCount();i++){

String ext = smart.getFiles().getFile(i).getFileExt() ; //取得文件后缀

String fileName = its.getIPTimeRandom() + "." + ext; //拼凑文件名称

smart.getFiles().getFile(i).saveAs(getServletContext().getRealPath("/") +

"upload" + java.io.File.separator + fileName) ; //保存文件

}

%>

</body>

</html>

可以通过正则进行限制文件上传的类型和合法性：

**If(smart.getFiles().getFile(0).getFileName().matches(“^\\w+\\.(jpg|gif)$”)){}表示只允许上传jpg和GIF格式文件。**

在以后普通的web开发中，使用SmartUpload是一个明智的选择。

**FileUpload接收上传内容**

FileUpload上传的基本步骤：

1. 创建磁盘工厂：DiskFileItemFactory factory = new DiskFileItemFactory();
2. 创建处理工具：ServletFileUpload upload = new ServletFileUpload(factory);
3. 设置上传文件大小：upload.setFileSizeMax(3145728);
4. 接收全部内容：List<FileItem> items = upload.parseRequest(request);

**使用FileUpload上传文件时，会将所有的上传内容一起接收，所以需要依次判断每一次上传的内容是普通文本还是文件。**

**范例：接收上传文件**

<?xml version=*'1.0'* encoding=*'GBK'* ?>

<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Strict//EN' 'http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd'>

<html xmlns=*'http://www.w3.org/1999/xhtml'*>

<head>

<meta http-equiv=*'Content-Type'* content=*'text/html; charset=GBK'* />

<title>使用FileUpload接收上传内容</title>

</head>

<body>

<form action = *"fileupload\_demo01.jsp"* method = *"post"* enctype = *"multipart/form-data"*>

姓名：<input type = *"text"* name = *"uname"*></input><br></br>

照片：<input type = *"file"* name = *"pic"*></input><br></br>

<input type = *"submit"* value = *"上传"*></input>

<input type = *"reset"* value = *"重置"*></input>

</form>

</body>

</html><?xml version=*"1.0"* encoding=*"GBK"* ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<%@ page import = *"java.util.\*"* %>

<%@ page import = *"org.apache.commons.fileupload.FileItem.\*"* %>

<%@ page import = *"org.apache.commons.fileupload.RequestContext.\*"* %>

<%@ page import = *"org.apache.commons.fileupload.disk.DiskFileItemFactory.\*"* %>

<%@ page import = *"org.apache.commons.fileupload.disk.\*"* %>

<%@ page import = *"org.apache.commons.fileupload.servlet.\*"* %>

<%@ page import = *"org.apache.commons.fileupload.\*"* %>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title>FileUpload接收上传文件</title>

</head>

<%request.setCharacterEncoding("GBK"); %>

<body>

<%

DiskFileItemFactory factory = **new** DiskFileItemFactory(); //创建磁盘工厂

ServletFileUpload upload = **new** ServletFileUpload(factory); //创建处理工具

upload.setFileSizeMax(3145728); //设置上传大小为3MB

List<FileItem> items = upload.parseRequest(request); //接收全部内容

Iterator<FileItem> iter = items.iterator(); //将全部的内容变为iterator的实例

**while**(iter.hasNext()){

FileItem item = iter.next(); //取出每一个上传的文件

String fieldName = item.getFieldName(); //得到表单控件名称

%>

<ul><h3>表单控件名称：<%=fieldName %>--><%=item.isFormField() %></h3>

<%

**if**(!item.isFormField()){ //不是普通的文本数据，是上传文件

String fileName = item.getName(); //取得文件名称

String contentType = item.getContentType(); //取得文件类型

**long** sizeInBytes = item.getSize(); //取得文件大小

%>

<li>上传文件名称：<%=fileName %></li>

<li>上传文件类型：<%=contentType %></li>

<li>上传文件大小：<%=sizeInBytes %></li>

<%

}**else**{

String value = item.getString(); //取得表单内容

%>

<li>普通参数：<%=value %></li>

<%

}

%>

</ul>

<%

}

%>

</body>

</html>

但是可以发现以上的程序只是对文件进行了上传，但是没有进行文件的保存。

**范例：利用FileUpload保存上传文件**

<?xml version=*'1.0'* encoding=*'GBK'* ?>

<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Strict//EN' 'http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd'>

<html xmlns=*'http://www.w3.org/1999/xhtml'*>

<head>

<meta http-equiv=*'Content-Type'* content=*'text/html; charset=GBK'* />

<title>同时上传多个文件</title>

</head>

<body>

<form action = *"fileupload\_demo02.jsp"* method = *"post"* enctype = *"multipart/form-data"*>

姓名：<input type = *"text"* name = *"uname"*></input><br></br>

照片：<input type = *"file"* name = *"pic1"*></input><br></br>

照片：<input type = *"file"* name = *"pic2"*></input><br></br>

照片：<input type = *"file"* name = *"pic3"*></input><br></br>

<input type = *"submit"* value = *"提交"*></input>

<input type = *"reset"* value = *"重置"*></input>

</form>

</body>

</html>

<?xml version=*"1.0"* encoding=*"GBK"* ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import=*"unit\_8.IPTimeStamp"*%>

<%@page import=*"org.apache.commons.fileupload.disk.DiskFileItemFactory"*%>

<html xmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<%@ page language=*"java"* contentType=*"text/html; charset=GBK"*

pageEncoding=*"GBK"*%>

<%@ page import = *"org.apache.commons.fileupload.disk.\*"* %>

<%@ page import = *"org.apache.commons.fileupload.servlet.\*"* %>

<%@ page import = *"org.apache.commons.fileupload.\*"* %>

<%@ page import = *"java.util.\*,java.io.\*"* %>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=GBK"* />

<title>保存上传内容</title>

</head>

<body>

<%

DiskFileItemFactory factory = **new** DiskFileItemFactory(); //创建磁盘工厂

factory.setRepository(**new** File(**this**.getServletContext().getRealPath("/") + "uploadtemp")); //创建临时文件夹

ServletFileUpload upload = **new** ServletFileUpload(factory); //创建处理工具

upload.setFileSizeMax(3145728); //设置上传文件大小

List<FileItem> items = upload.parseRequest(request); //接收全部内容

Iterator<FileItem> iter = items.iterator(); //将全部内容变成iterator实例

IPTimeStamp its = **new** IPTimeStamp(request.getRemoteAddr()); //实例化ip日期时间戳对象

**while**(iter.hasNext()){ //依次取出每一个内容

FileItem item = iter.next();

String fieldName = item.getFieldName(); //取得表单控件的名称

**if**(!item.isFormField()){ //不是普通的文本数据，是上传文件

File savaFile = **null**; //定义保存的文件

InputStream input = **null**; //定义文件输入流，用于读取源文件

OutputStream output = **null**; //定义文件输出流，用于保存文件

input = item.getInputStream(); //取得上传文件的输入流

output = **new** FileOutputStream(**new** File(**this**.getServletContext().getRealPath("/") + "upload" +

File.separator + its.getIPTimeRandom() + "." + item.getName().split("\\.")[1]));

**byte** data[] = **new** **byte**[512]; //分块保存

**int** temp = 0;

**while**((temp = input.read(data, 0, 512)) != -1){ //依次读取内容

output.write(data); //保存内容

}

input.close();

output.close();

}**else**{

String value = item.getString(); //取得表单内容

%>

普通参数：<%=value %>

<%

}

String address = **this**.getServletContext().getRealPath("/");

%>

<h3>真实地址：<%=address %></h3>

<%

}

%>

此方法需要在存放文件的目录中手动创建文件夹。

**建议尽量使用SmartUpload进行文件的上传，FileUpload只是在框架中才使用，但是在框架中使用也没有这么麻烦了。**