本次课程的知识点：

掌握smartupload组件的作用

使用smartupload上传文件

为上传的文件重命名

Smartupload的作用：

主要功能就是完成上传操作，如果想要上传的话，则直接使用表单中的file元素即可

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <form>  <input type=*"file"*/>  </form> |

网络上常见的上传操作有两种组件，Smartupload,Apache FileUpload

Smartupload组件需要一个组件架档

直接复制到WEB-INF 的 lib目录中即可

使用Smartupload上传文件：

·基本上传

建立表单：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <form action=*"Demo02.jsp"* method=*"post"*>  选择文件：<input type=*"file"*/><br>  <input type=*"submit"* value=*"提交"*/>  </form> |

表单建立好了，之后就要建立jsp文件接收上传的内容，上传的步骤分为以下几步：

·初始化上传文件

·准备上传

·保存文件

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <jsp:useBean id=*"smartupload"* class=*"org.lxh.smart.SmartUpload"* scope=*"page"*>  </jsp:useBean>  <html>  <head>  <title>smartupload</title>  </head>  <body>  <%  smartupload.initialize(pageContext);//初始化上传文件  smartupload.upload();//准备上传  smartupload.save("upload");//保存文件    %>  </body>  </html> |

此时程序执行没有出错，但是upload文件夹中并没有我们上传的文件，为什么呢？

因为正常情况下表单提交的都是文本数据，而如果现在想要上传文件，此时需要明确的表示提交并不是简单的文本，此时表单需要封转，如果不封装肯定无法上传文件

现在进行表单封转：在表单中加入 enctype属性，表示对表单进行封装

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <form action=*"Demo02.jsp"* method=*"post"* enctype=*"multipart/form-data"*>  选择文件：<input type=*"file"*/><br>  <input type=*"submit"* value=*"提交"*/>  </form> |

提交表单，再次提交，发现已经可以上传，而且保存的文件名称和上传前的完全一致

表单封装之后，request无法取得参数了

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <form action=*"Demo04.jsp"* method=*"post"* enctype=*"multipart/form-data"*>  选择文件：<input type=*"file"* name=*"pic"*/><br>  姓名：<input type=*"text"* name=*"name"*/><br>  <input type=*"submit"* value=*"提交"*/>  </form> |
| <%@page contentType=*"text/html; charset=GBK"*%>  <jsp:useBean id=*"smartupload"* class=*"org.lxh.smart.SmartUpload"* scope=*"page"*>  </jsp:useBean>  <html>  <head>  <title>smartupload</title>  </head>  <body>  <%  request.setCharacterEncoding("GBK");  String name = request.getParameter("name");  smartupload.initialize(pageContext);//初始化上传文件  smartupload.upload();//准备上传  smartupload.save("upload");//保存文件    %>  <h1><%= name%></h1>  </body>  </html> |

表单封装之后，但是request接收不到参数了，但是不封装呢，又无法保存上传文件，因为表单封装之后，里面传递的就不是文本数据，所以根本无法通过request.getParameter()方法取得参数的值，此时就要依靠smartupload中的request对象完成，但是在使用smartupload中的request对象之前，必须保证smartupload必须已经初始化。

修改Demo04.jsp:

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <jsp:useBean id=*"smartupload"* class=*"org.lxh.smart.SmartUpload"* scope=*"page"*>  </jsp:useBean>  <html>  <head>  <title>smartupload</title>  </head>  <body>  <%  request.setCharacterEncoding("GBK");  smartupload.initialize(pageContext);//初始化上传文件  smartupload.upload();//准备上传  String name = smartupload.getRequest().getParameter("name");//通过smartupload中的request对象完成  smartupload.save("upload");//保存文件    %>  <h1><%= name%></h1>  </body>  </html> |

为上传图片改名：

因为图片上传可能存在同名的情况，所以在上传图片的时候需要改名

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <form action=*"Demo06.jsp"* method=*"post"* enctype=*"multipart/form-data"*>  选择文件：<input type=*"file"* name=*"pic"*/><br>  文件名称：<input type=*"text"* name=*"picname"*/><br>  <input type=*"submit"* value=*"提交"*/>  </form> |
| <%@page contentType=*"text/html; charset=GBK"*%>  <jsp:useBean id=*"smartupload"* class=*"org.lxh.smart.SmartUpload"* scope=*"page"*>  </jsp:useBean>  <html>  <head>  <title>smartupload</title>  </head>  <body>  <%  request.setCharacterEncoding("GBK");  smartupload.initialize(pageContext);//初始化上传文件  smartupload.upload();//准备上传  String name = smartupload.getRequest().getParameter("picname");//取得文件名称  String end\_name = smartupload.getFiles().getFile(0).getFileExt();//取得文件的后缀  String filename = getServletContext().getRealPath("/")+"upload/"+name+"."+end\_name;//文件的绝对路径  smartupload.getFiles().getFile(0).saveAs(filename);//保存文件  %>  <h1><%= name%></h1>  </body>  </html> |

以上程序虽然可以改变上传文档的名称，但是文件名称还是可能重复，一旦重复，新文件就会覆盖就的文件

为上传文件自动进行更名：

如果想要自动进行更名，必须要考虑如何起名字，课可以让文件名称不重复

可以采用以下方式进行命名：

·客户端IP+时间戳+三位随即数

因为计算还是有点复杂，此时编写一个javaBean，此javaBean自动生成文件的后缀

|  |
| --- |
| **package** com.wanczy.jjm;  **import** java.text.DateFormat;  **import** java.text.SimpleDateFormat;  **import** java.util.Date;  **import** java.util.Random;  **public** **class** Img\_name {  **private** String ip;  **public** String getIp() {  **return** ip;  }  **public** **void** setIp(String ip) {  **this**.ip = ip;  }  **public** Img\_name(){  }  **public** Img\_name(String ip){  **this**.ip = ip;  }  **public** String getTime(){  String temp = **null**;  DateFormat df = **new** SimpleDateFormat("yyyyMMddHHmmssSSS");  temp = df.format(**new** Date());  **return** temp;  }  **public** String getImg\_name(){  StringBuffer buf = **new** StringBuffer();  **if**(**this**.ip!= **null**){  String ip[] = **this**.ip.split("\\.");  System.*out*.println("====");  **for**(**int** i = 0;i<ip.length;i++){  System.*out*.println("\_\_\_\_\_");  buf.append(**this**.addZero(ip[i],3));  System.*out*.println("<<<<<<<<");  }  }  buf.append(**this**.getTime());    Random r = **new** Random();  **for**(**int** i = 0;i<3;i++){  buf.append(r.nextInt(10));  }  **return** buf.toString();    }  **public** String addZero(String str,**int** length){  StringBuffer sb = **new** StringBuffer(str);  **while**(sb.length()<length){  sb.insert(0, '0');  }  **return** sb.toString();  }    } |

数据库操作：

一般在开发中，需要将一些图片信息和文档保存在数据库里面，此时有两种做法：

·直接将图片保存在数据库中，使用BLOB字段保存

|-如果上传文件比较大，数据库性能会降低，存储、读取、更新都非常的麻烦

|-如果数据库需要备份，直接备份数据量

·将图片的地址保存在数据库里面，使用varchar2保存

|-不管文件内容多大，多少，都非常好存储和读取

|-既要备份数据库，也要备份文件

我们主要是接收第二种方式来进行数据量操作：

数据库创建脚本：

|  |
| --- |
| *--drop table jt39\_jjm\_person;*  create table jt39\_jjm\_person(  p\_name varchar2(20),  p\_photo varchar2(50),  p\_age number(3)  ); |

新增表单：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <form action=*"Demo08.jsp"* method=*"post"* onsubmit=*"return onChe(this)"* enctype=*"multipart/form-data"*>  姓名：<input type=*"text"* name=*"p\_name"*/><br>  年龄：<input type=*"text"* name=*"p\_age"*><br>  头像：<input type=*"file"* name=*"p\_photo"*/><br>  <input type=*"submit"* value=*"提交"*/>  </form>  <script type=*"text/javascript"*>  **function** **onChe**(f){  **var** age = f.p\_age.value;  **if**(!(/^\d{1,3}$/.test(age))){  alert("年龄必须为数字");  f.p\_age.focus();  **return** **false**;  }  **return** **true**;  }  </script> |

保存文件和数据库插入操作：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"* import=*"java.sql.\*"*%>  <jsp:useBean id=*"smartupload"* class=*"org.lxh.smart.SmartUpload"* scope=*"page"*>  </jsp:useBean>  <jsp:useBean id=*"getName"* class=*"com.wanczy.jjm.Img\_name"* scope=*"page"*>  </jsp:useBean>  <jsp:useBean id=*"conn"* class=*"com.wanczy.jjm.DateBaseConnection"* scope=*"page"*>  </jsp:useBean>  <html>  <head>  <title>smartupload</title>  </head>  <body>  <%  request.setCharacterEncoding("GBK");  smartupload.initialize(pageContext);//初始化上传文件  smartupload.upload();//准备上传  String p\_name = smartupload.getRequest().getParameter("p\_name");  String p\_age = smartupload.getRequest().getParameter("p\_age");  String ip = request.getRemoteAddr();//获取客户端IP  getName.setIp(ip);  String name = getName.getImg\_name();//取得文件名称  String end\_name = smartupload.getFiles().getFile(0).getFileExt();//取得文件的后缀  String filename = getServletContext().getRealPath("/")+"upload/"+name+"."+end\_name;//文件的绝对路径  smartupload.getFiles().getFile(0).saveAs(filename);//保存文件  %>  <%  PreparedStatement pstamt = **null**;  **try** {  String sql = "insert into jt39\_jjm\_person(p\_name,p\_age,p\_photo) values(?,?,?)";  pstamt = conn.getConn().prepareStatement(sql);  pstamt.setString(1,p\_name);  pstamt.setInt(2,Integer.parseInt(p\_age));  pstamt.setString(3,name+"."+end\_name);  pstamt.execute();  %>  <h2>新增成功</h2>  <%  }**catch**(Exception e){  %>  <h2>新增失败</h2>  <%  }**finally**{  pstamt.close();  conn.close();  }  %>  </body>  </html> |

以上代码运行结果显示，新增已经没有问题了

接下来进行查询操作：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%@page import=*"java.sql.\*"* %>  <jsp:useBean id=*"conn"* class=*"com.wanczy.jjm.DateBaseConnection"* scope=*"page"*></jsp:useBean>  <%  PreparedStatement pstamt = **null**;  ResultSet res = **null**;  **try** {  String sql = "select p\_name,p\_age,p\_photo from jt39\_jjm\_person";  pstamt = conn.getConn().prepareStatement(sql);  res = pstamt.executeQuery();  %>  <center>  <h1>  人员查询  </h1>  <table border=*"1"* bordercolor=*"green"* width=*"70%"*>  <tr>  <th>  姓名  </th>  <th>  年龄  </th>  <th>  头像  </th>    </tr>  <%  **while** (res.next()) {  String p\_name = res.getString(1);  **int** p\_age = res.getInt(2);  String p\_photo = res.getString(3);  %>  <tr>  <td><%=p\_name%></td>  <td><%=p\_age%></td>  <td><img src=*"*<%=getServletContext().getRealPath("/")+"upload/"+ p\_photo%>*"* width=*"50"* height=*"70"*/></td>    </tr>  <%  }  } **catch** (Exception e) {  } **finally** {  **try** {  res.close();  pstamt.close();  conn.close();  } **catch** (Exception e) {  }  }  %>  </table>  </center> |

条件查询：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%@page import=*"java.sql.\*"* %>  <jsp:useBean id=*"conn"* class=*"com.wanczy.jjm.DateBaseConnection"* scope=*"page"*></jsp:useBean>  <center>  <form action=*"Demo09.jsp"* method=*"post"*>  姓名：<input type=*"text"* name=*"p\_name\_q"*>  <input type=*"submit"* value=*"查询"*>  </form>  </center>  <hr>  <%  request.setCharacterEncoding("GBK");  String p\_name\_q = request.getParameter("p\_name\_q");  System.out.print(p\_name\_q);  PreparedStatement pstamt = **null**;  ResultSet res = **null**;  **try** {  String sql = "select p\_name,p\_age,p\_photo from jt39\_jjm\_person where p\_name like ?";  pstamt = conn.getConn().prepareStatement(sql);  pstamt.setString(1,"%"+p\_name\_q+"%");  res = pstamt.executeQuery();  %>  <center>  <h1>  人员查询  </h1>  <table border=*"1"* bordercolor=*"green"* width=*"70%"*>  <tr>  <th>  姓名  </th>  <th>  年龄  </th>  <th>  头像  </th>    </tr>  <%  **while** (res.next()) {  String p\_name = res.getString(1);  **int** p\_age = res.getInt(2);  String p\_photo = res.getString(3);  %>  <tr>  <td><%=p\_name%></td>  <td><%=p\_age%></td>  <td><img src=*"*<%=getServletContext().getRealPath("/")+"upload/"+ p\_photo%>*"* width=*"50"* height=*"70"*/></td>    </tr>  <%  }  } **catch** (Exception e) {  } **finally** {  **try** {  res.close();  pstamt.close();  conn.close();  } **catch** (Exception e) {  }  }  %>  </table>  </center> |