图片上传功能:

前端注意点：

前端使用POST请求提交表单

并设置enctype="multipart/form-data"（注意：设置该参数将导致request对象获取不到请求信息）

|  |
| --- |
| <form action=*"servlet/AddServlet2"* method=*"post"* enctype=*"multipart/form-data"*>  用户名:<input type=*"text"* name=*"name"*/><br>  年龄:<input type=*"text"* name=*"age"*/><br>  分数:<input type=*"text"* name=*"score"*/><br>  照片:<input type=*"file"* name=*"photo"*/><br>  <input type=*"submit"* value=*"提交"*/>  </form> |

图片上传功能分析：

图片存储在服务器上，通过图片在服务器上的名称进行保存，方便获取

图片上传功能步骤实现：

1、创建工厂类FileItemFactory对象

2、创建ServletFileUpload对象，把工厂对象作为参数传递进去

3、限制上传文件大小，上传时如果因文件过大而导致上传不成功，可以手动调大

4、通过ServletFileUpload对象的parseRequest方法，把request对象传进去，获取请求信息，并返回一个FileItem集合

5、遍历FileItem集合，获取请求信息，对请求信息进行处理

6、判断上传文件类型，设置上传文件类型必须为指定的文件类型

7、获取Tomcat服务器地址

|  |
| --- |
| String realPath = **this**.getServletContext().getRealPath("/upload"); |

8、使用该地址创建文件夹，并服务器中是否有该文件夹，没有则创建

9、获取上传文件的真实名称，并获取服务器名称，上传文件类型

10、把文件写进服务器的指定位置

11、接下来就是数据保存工作及相关业务处理了

在查询页显示图片：

由于图片被存储到服务器里面，所以可以通过地址访问直接拿到图片

实现代码:

通过Servlet转发实现：

|  |
| --- |
| <td><img alt=*""* src=*"upload/*${stu.photoName}*"* width=*"100px"*></td> |

文件下载功能：

1、前端下载请求图片id，后端查找相应对象

2、通过对象获取图片存储位置，把路径传到File对象里面

3、获取图片真实名称

4、获取请求头信息，判断浏览器类型，由于不同浏览器对下载的处理方式不同

|  |
| --- |
| String userAgent = request.getHeader("User-Agent").toLowerCase(); |

5、注意点：前端获取到图片名称可能出现乱码，要先设置成utf-8

|  |
| --- |
| **if**(userAgent.indexOf("mozilla")>=0){//如果是IE  realName = URLEncoder.*encode*(realName,"utf-8");  }**else**{//其他  realName = **new** String(realName.getBytes("utf-8"), "iso-8859-1");  } |

6、设置响应（response）的文件大小，文件类型，并设置响应头，以附件形式下载

|  |
| --- |
| response.setContentLength((**int**)file.length());  response.setContentType(stu.getPhotoType());  response.setHeader("Content-disposition","attachment;filename="+realName); |

7、创建文件读取流，从服务器端读取文件

8、创建响应输出流，把图片响应给浏览器

9、使用apache下的commons.io中的工具类IOUitls进行拷贝

10、关闭流，先开后关原则