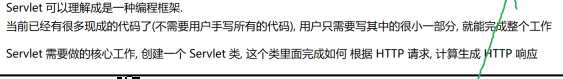
理解Servlet:(根据之前的Socket）

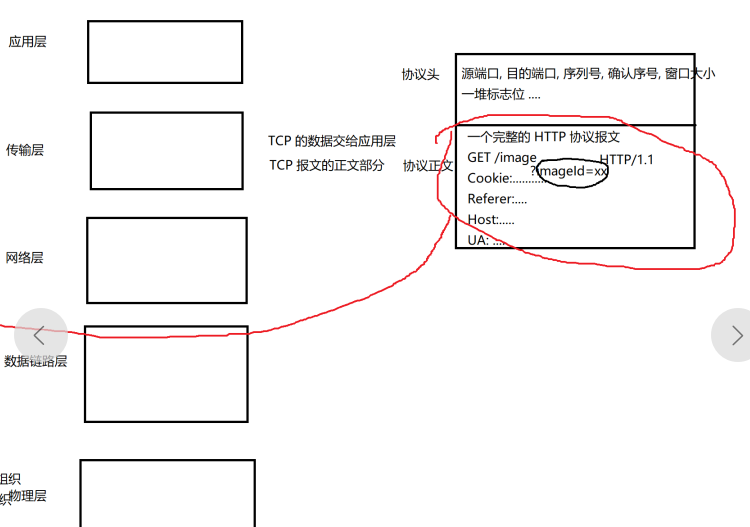


severStart{
Socket socket对象
socket.bind(服务器端+端口号)
while（true）{
Socket new Socket=socjet.accpet();
//读取客户端发来的数据，这个数据相当于一份完整的HTTP请求
Byte[]buffer=new Socket.read()；
HttpServletRequest req=parse(buffer);
//将HTTP的请求数据读入buffer中，进行解析得到request对象req
//调用用户自己编写的代码ImageServlet
HttpServlet h=bulid(req.getUrl())
//根据req对象中URL的path路径，决定创建的对象
//然后根据URL在web.xml中找到对应的类，创建对象，在这里根据/image找到类时httpServlet
if(req.getMethod().equals("Get")){
h.doGet(req,resp);
}else if(req.getMethod().equals("Post")){
h.doPost(req,resp);
}else{
......
}
//执行完上面的代码后，再讲resp对象转为字符串，写回socket
new Socket.write(resp.toString());
}
}

Request对象包含的参数：



HTTP协议内容（HTTP存在于应用层）



**部署(linux windows都是一样的）：**

**使用maven打包(war包)**

**把war包拷贝到Tomcat的webapps下部署完成之后 进行访问浏览器**

47.98.116.42是老师的本机ip

注意：自己本地的tomcat只有内部可以进行访问 别人不能用



**ImageServlet:**

上传图片 POST

查看指定图片 GET

查看所有图片 GET

删除图片 GET

**ImageShow**

查看图片内容(因为之前规定的api的path路径不同）GET

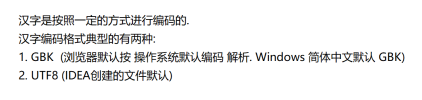
**上传图片（fileupload）：**

**1.获取图片的属性信息，存在数据库**

**2.获取图片的内容信息，在磁盘文件中**

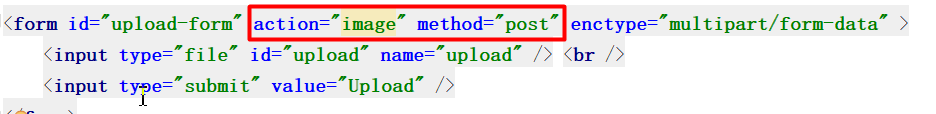
**3.给客户端返回一个结果数据**

出现乱码原因：



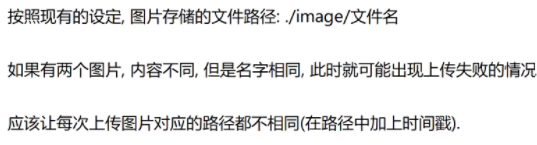
如何改正：（修改main——webapp——upload.html中的字符集编码）

**Upload.html作用：**



action代表路径 method代表上传图片是请求的方法

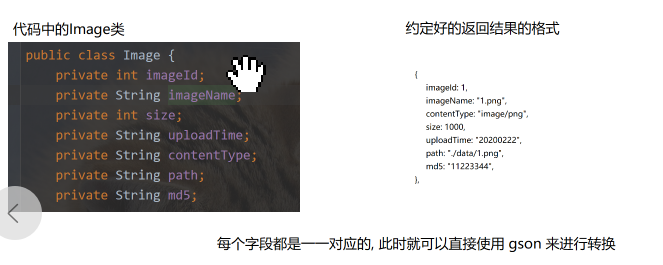
**解决名称相同，内容不同的图片插入失败的情况：**



**查看图片：**

**根据ImageDao获取图片信息，在利用json进行转换为字符串**

注意：Image的属性应该之前规定好的一致



// 1. 创建一个 ImageDao 对象, 并查找数据库
ImageDao imageDao = new ImageDao();
List<Image> images = imageDao.selectAll();
// 2. 把查找到的结果转成 JSON 格式的字符串, 并且写回给 resp 对象
Gson gson = new GsonBuilder().create();
//jsonData 就是一个 json 格式的字符串了, 与之前约定的格式是一样的.
String jsonData = gson.toJson(images);
resp.getWriter().write(jsonData);

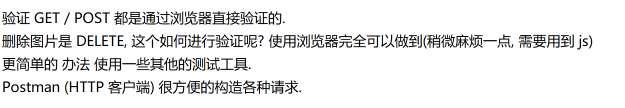
**删除图片：**

**1.获取到imageId**

**2.查找图片相关属性，主要是路径**

**3.删除数据库记录**

**4.删除磁盘内容**



**查看图片内容：**

**1.解析出imageId**

**2.根据imageId查找数据库，得到相关图片信息，得到图片存储路径**

**3.根据路径打开文件，读取图片内容，写入响应中**