## 响应对象 HttpServletResponse（接口）

### **响应行：**

**协议：HTTP1.1，当前请求、响应使用的网络传输协议为http的1.1版本**

响应状态码**(200成功 304重定向 401权限 404notfound资源路径错误**

**405方法的路径问题 500空指针异常、sql异常、数学异常、路径异常)**

### 响应头：

* **Content-Disposition：告诉浏览器以下载的形式打开数据(支持下载)**
* **Content-Length：响应数据的长度**
* **Content-Type：响应数据的MIME类型**
  + - **MIME类型：大类型/小类型**
    - **text/html;charset=utf-8**
    - **image/jpg png jpeg**
    - **video/mp4 avi wmv**
* **除了文本类型需要指定编码集之外，其余的都不需要指定，二进制数据没有编码！**
* **Date：响应的时间点**
* **refresh：定时刷新，应用场景居多，可以实现页面的延迟跳转！**

### 添加响应头的方法 addHeader setHeader

**response.setHeader(“头信息关键字”, “头信息的值”)**

**response.addHeader(“头信息关键字”, “头信息的值”)**

**区别：**

**setHeader是设置头信息的方法，那么同一个头信息关键字只能对应一个值，如果重复了，则将之前的值替换**

**addHeader是添加头信息的方法，可以保证一个头信息关键字对应多个值**

**但是在以后的开发场景中，只会接触到一个头信息关键字对应一个值，所以两个方法就没有区别！！！**

### 响应体：

**响应对象response在返回数据、响应数据时，会将html、图片、视频等数据通过响应主体返回，而绝大多数响应数据都是文本，我们可以将响应的数据理解为响应体！**

**自带了两个输出流：**

**response.getWriter() ==> PrintWriter**

**response.getOutputStream ==> ServletOutputStream**

**请求对象 HttpServletRequest**

**请求对象的作用：接收客户端(浏览器)的请求，获取请求中的信息。除了获取请求的数据之外，还可以获取比如主机地址、端口、请求方式、请求的参数列表、浏览器内核信息等**

**请求分为三部分：请求行、请求头、请求体**

### 请求行：

**request.getServerName()//主机地址**

**request.getServerPort()//tomcat端口号**

**request.getProtocol()//协议**

**request.getRequestURI()//完整的url主体路径**

**request.getContextPath()//项目工程名**

**request.getQueryString()//参数列表**

**request.getMethod()//请求方式**

**请求行：**

request.getHeader(“请求头的关键字”)

request.getHeaderNames()

### 请求体：

获取参数的两种方法：

**request.getParameter(“参数关键字”) ==> String**

**request.getParameterValues(“参数关键字”) ==> String[]**

**中文乱码：**

**tomcat8.0自动处理了get的中文乱码，**

**post需要设置参数的编码方式**

**request.setCharacterEncoding(“utf-8”)**

**在tomcat8.0之前，get请求处理中文乱码的代码：**

**String sname = request.getParameter(“sname”);**

**sname = new String(sname.getBytes(“iso8859-1”), “utf-8”);**

**但是不管在哪一个版本下，post请求需要手动设置中文的编码格式**

**request.setCharacterEncoding(“utf-8”)**

### 请求头：

**记住两个方法即可：**

**request.getHeader(“单个请求头的关键字”) ==> 得到对应请求头的值**

**request.getHeaderNames() ==> 所有头信息的关键字，遍历得到每一个关键字，再调用getHeader()既可以得到所有关键字对应的值！**

案例：

图片回显

使用a标签连接后台完成下载

上传：

改变表单的提交方式为post，修改enctype属性为multipart/form-data

为负责上传的servlet添加@MultipartConfig注解

通过request.getPart(“文件域的关键字”) ==> Part，保存着上传内容

从Part中获取文件名

将part中的内容写入到本地目录文件中