request

2019年5月17日 星期五 9:05

1. 回顾：HTTP协议
   a. HTTP数据交互模型：
      请求响应模型。
      一次请求对应一次响应。
      请求只能由浏览器发出，服务器可以根据请求做出响应。
2. HTTP请求
   a. 继承结构
      ServletRequest 提供servletrequest对象必要的方法。
      |
      |---HttpServletRequest 包含和HTTP协议操作的一些API，可以通过这些API来开发浏览器和服务器之间的功能。
   b. HTTP请求的组成
      一个请求行 请求方式 请求资源的路径 HTTP协议版本
      多个请求头 Host：虚拟主机名称
      一个空行
      请求实体内容 用户发送请求时携带的请求参数
3. Request当中的API操作：
   - 获取客户端相关的信息

     getRequestURL方法 -- 返回客户端发出请求完整URL

     getRequestURI方法 -- 返回请求行中的资源名部分

     getQueryString方法 -- 返回请求行中的参数部分

     getRemoteAddr方法 -- 返回发出请求的客户机的IP地址

     getMethod -- 得到客户机请求方式

     !!getContextPath -- 获得当前web应用虚拟目录名称 -- 在写路径时不要将web应用的虚拟路径的名称写死，应该在需要写web应用的名称的地方通过getContextPath方法动态获取
   - 获取请求头信息

     getHeader(name)方法 --- String

     getHeaders(String name)方法 --- Enumeration<String>

     getHeaderNames方法 --- Enumeration<String>

     //快捷键获取返回值：Alt+Shift+L；注意：光标放在最后。

     getIntHeader(name)方法 --- int

     getDateHeader(name)方法 --- long(日期对应毫秒)
   - 获取请求参数

     getParameter(String name) --- String 通过name获得值

     getParameterValues(String name) --- String[ ] 通过name获得多值 checkbox

     getParameterMap() --- Map<String,String[ ]> key :name value: 多值 将查询的参数保存在一个Map中

getParameterNames() --- Enumeration<String> 获得所有name

i. 获取请求参数乱码问题：

乱码出现的原因：编解码不一致。

ii. 浏览器端发送的数据字符集使用的UTF-8，原因是在浏览器打开的时候使用的是什么字符集发送数据的时候就会使用什么字符集进行编码操作。

iii. tomcat服务器接收数据字符集ISO8859-1，因为服务器的默认字符集是ISO8859-1，所以在收到浏览器发送的以UTF-8编码的数据时解码过程会出现乱码。

iv. 统一浏览器端和服务器的字符集为UTF-8.

1） 修改服务器接收数据时使用的字符集。

a） request.setCharacterEncoding("utf-8")；
当前语句仅对post请求生效，因为用户使用post提交时，请求参数会存储在请求实体内容中。语句可以将请求实体内容中的中文参数乱码做出处理。

b） 由于get提交参数在地址栏中拼接传输，没有进入请求实体内容，所以此处无法处理get提交的乱码。

c） 注意：post乱码处理一定要放在获取参数的代码之前。

2） get提交参数乱码处理

a） 先对获取到的中文进行编码操作，使用字符集iso8859-1，获取到真正的二进制数据，将二进制数据使用正确是utf-8字符集重新解码。

```
byte[] bytes = ??????.getBytes("iso8859-1");
```

b） 使用utf-8字符集解码二进制数据。

```
username = new String(bytes,"utf-8");
```

c) 这种处理方式其实就是讲获取到的参数重新编码、解码各一次，所以不管get提交还是post提交参数都会重新编解码，所以它既能解决get提交乱码也可以解决post提交乱码。

```
username = new String(????.getBytes("iso8859-1"),"utf-8");
```