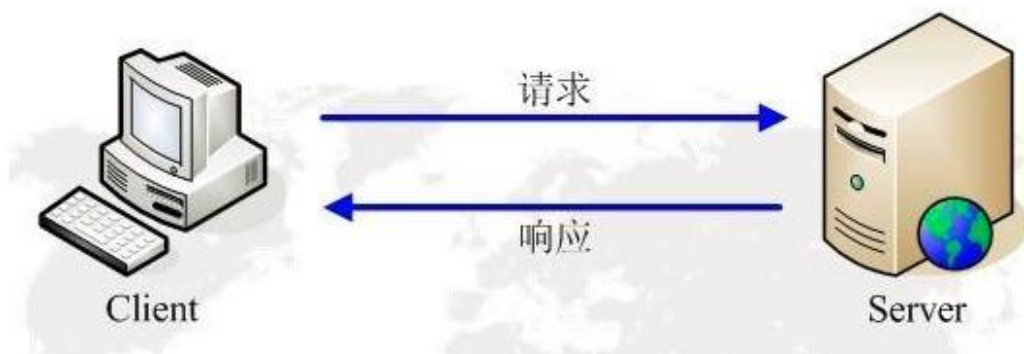


课程介绍

互联网的发展非常迅速，但是万变不离其宗。学习 *web* 开发，需要我们对互联网的交互机制有一定的了解。为了更好的理解并掌握 *Servlet*，在正式学习 *Servlet* 之前需要对 *web* 开发中客户端和服务器的交互机制以及所涉及的规范 (HTTP) 进行学习。

HTTP 协议详解

1、web 交互的基本流程



图片介绍：

客户端根据用户输入的地址信息请求服务器，服务器在接收到用户的请求后进行处理，然后将处理结果响应给客户端，客户端将响应结果展示给用户。

专业术语：

请求：客户端根据用户地址信息将数据发送给服务器的过程

响应：服务器将请求的处理结果发送给浏览器的过程

问题：

客户端也就是浏览器的版本是有很多的，服务器的版本也是有很多的，如何实现不同版本的浏览器和不同版本的服务器之间的数据交互呢？

解决：

规范浏览器和服务器的数据交互的格式。

实现：

HTTP 协议

2、 HTTP 的概念和介绍

概念：

超文本传输协议（Hyper Text Transfer Protocol）

作用：

规范了浏览器和服务器的数据交互

特点：

简单快速 :客户向服务器请求服务时 ,只需传送请求方法和路径。

请求方法常用的有 GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于 HTTP 协议简单，使得 HTTP 服务器的程序规模小，因而通信速度很快

灵活 :HTTP 允许传输任意类型的数据对象。正在传输的类型由 Content-Type 加以标记。

无连接 :无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。

.无状态 :HTTP 协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。

支持 B/S 及 C/S 模式。

HTTP1.1 版本后支持可持续连接

3、HTTP 的交互流程：

HTTP 的交互流程一般分为四个步骤(一次完整的请求)：

步骤一：

客户端和服务端建立连接

步骤二：

客户端发送请求数据到服务器端(HTTP 协议)

步骤三：

服务器端接收到请求后，进行处理，然后将

处理结果响应客户端(HTTP 协议)

步骤四：

关闭客户端和服务端连接(HTTP1.1 后不会立即关闭)

4、HTTP 协议之请求格式

请求格式的结构：

请求头：请求方式、请求的地址和 HTTP 协议版本

请求行：消息报头，一般用来说明客户端要使用的一些附加信息

空行：位于请求行和请求数据之间，空行是必须的。

请求数据：非必须。

请求方法	空格	URL	空格	协议版本	回车符	换行符	请求行
头部字段名	:	值	回车符	换行符	} 请求头部		
...							
头部字段名	:	值	回车符	换行符			
回车符	换行符						
						请求数据	

注意： 一张网页的内容是极其丰富的，浏览器会遵循 HTTP 请求的格式将有效数据发送给服务器。

示例(*get* 请求方式):

```
GET /562f25980001b1b106000338.jpg HTTP/1.1
Host: img.mukewang.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36
Accept: image/webp,image/*,*/*;q=0.8
Referer: http://www.imooc.com/
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
```

示例(*post* 请求方式):

```
POST / HTTP/1.1
Host: www.wrox.com
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)
Content-Type: application/x-www-form-urlencoded
Content-Length: 40
Connection: Keep-Alive

name=Professional%20Ajax&publisher=Wiley
```

5、HTTP 协议之请求方式

根据 HTTP 标准, HTTP 请求可以使用多种请求方法。

HTTP1.0 定义了三种请求方法: **GET**, **POST** 和 **HEAD** 方法。

HTTP1.1 新增了五种请求方法: **OPTIONS**, **PUT**, **DELETE**, **TRACE** 和 **CONNECT** 方法。

GET	请求指定的页面信息,并返回实体主体。
HEAD	类似于get请求,只不过返回的响应中没有具体的内容,用于获取报头。
POST	向指定资源提交数据进行处理请求(例如提交表单或者上传文件)。数据被包含在请求体中。POST请求可能会导致新的资源的建立和/或已有资源的修改。
PUT	从客户端向服务器传送的数据取代指定的文档的内容。
DELETE	请求服务器删除指定的页面。
CONNECT	HTTP/1.1协议中预留给能够将连接改为管道方式的代理服务器。
OPTIONS	允许客户端查看服务器的性能。
TRACE	回显服务器收到的请求,主要用于测试或诊断。

get 和 *post* 请求方式的区别:

get 请求方式:

请求数据会以?的形式隔开拼接在请求头中,不安全,没有请求实体部分。

HTTP 协议虽然没有规定请求数据的大小,但是浏览器对 URL 的长度是有限制的,所以 *get* 请求不能携带大量的数据。

post 请求方式:

请求数据在请求实体中进行发送, 在 *URL* 中看不到具体的请求数据, 安全。适合数据量大的数据发送。

6、HTTP 协议之响应

响应格式的结构:

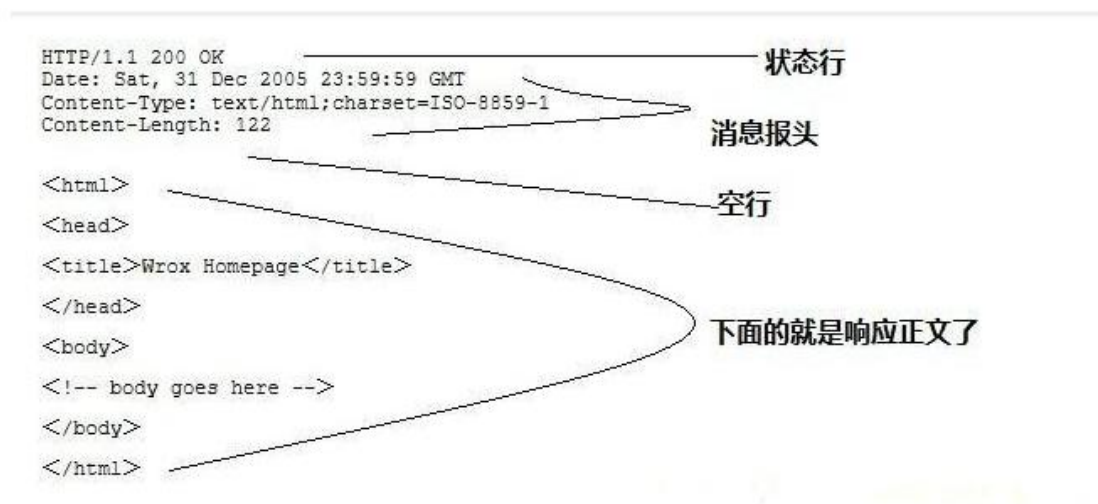
响应行(状态行): *HTTP* 版本、状态码、状态消息

响应头: 消息报头, 客户端使用的附加信息

空行: 响应头和响应实体之间的, 必须的。

响应实体: 正文, 服务器返回给浏览器的信息

示例:



HTTP 常见响应状态码含义:

HTTP 状态码由三个十进制数字组成, 第一个十进制数字定义了状态码的类型, 后两个数字没有分类的作用。*HTTP* 状态码共分为 5 种类型:

HTTP状态码分类

分类	分类描述
1**	信息，服务器收到请求，需要请求者继续执行操作
2**	成功，操作被成功接收并处理
3**	重定向，需要进一步的操作以完成请求
4**	客户端错误，请求包含语法错误或无法完成请求
5**	服务器错误，服务器在处理请求的过程中发生了错误

常见状态码：

200 OK //客户端请求成功

400 Bad Request //客户端请求有语法错误，不能被服务器所理解

401 Unauthorized //请求未经授权，这个状态代码必须和WWW-Authenticate 报头域一起使用

403 Forbidden //服务器收到请求，但是拒绝提供服务

404 Not Found //请求资源不存在，eg：输入了错误的 URL

500 Internal Server Error //服务器发生不可预期的错误

503 Server Unavailable //服务器当前不能处理客户端的请求，一段时间后可能恢复正常