



HTTP协议

尚硅谷: <http://www.atguigu.com/>

讲师: 张晓飞

主要内容

- 1 HTTP相关知识
- 2 HTTP请求
- 3 HTTP响应

从使用HttpWatcher工具开始

HTTP相关知识(1)

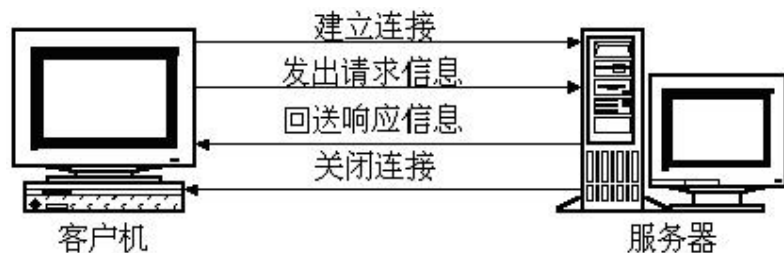
HTTP简介

- WEB浏览器与WEB服务器之间的一问一答的交互过程必须遵循一定的规则，这个规则就是HTTP协议。
- HTTP是 hypertext transfer protocol (超文本传输协议) 的简写，它是 TCP/IP 协议集中的一个应用层协议，用于定义WEB浏览器与WEB服务器之间交换数据的过程以及数据本身的格式。
- HTTP协议的版本 HTTP/1.0、HTTP/1.1、HTTP-NG

HTTP相关知识(2)

HTTP 的会话方式

- 四个步骤：

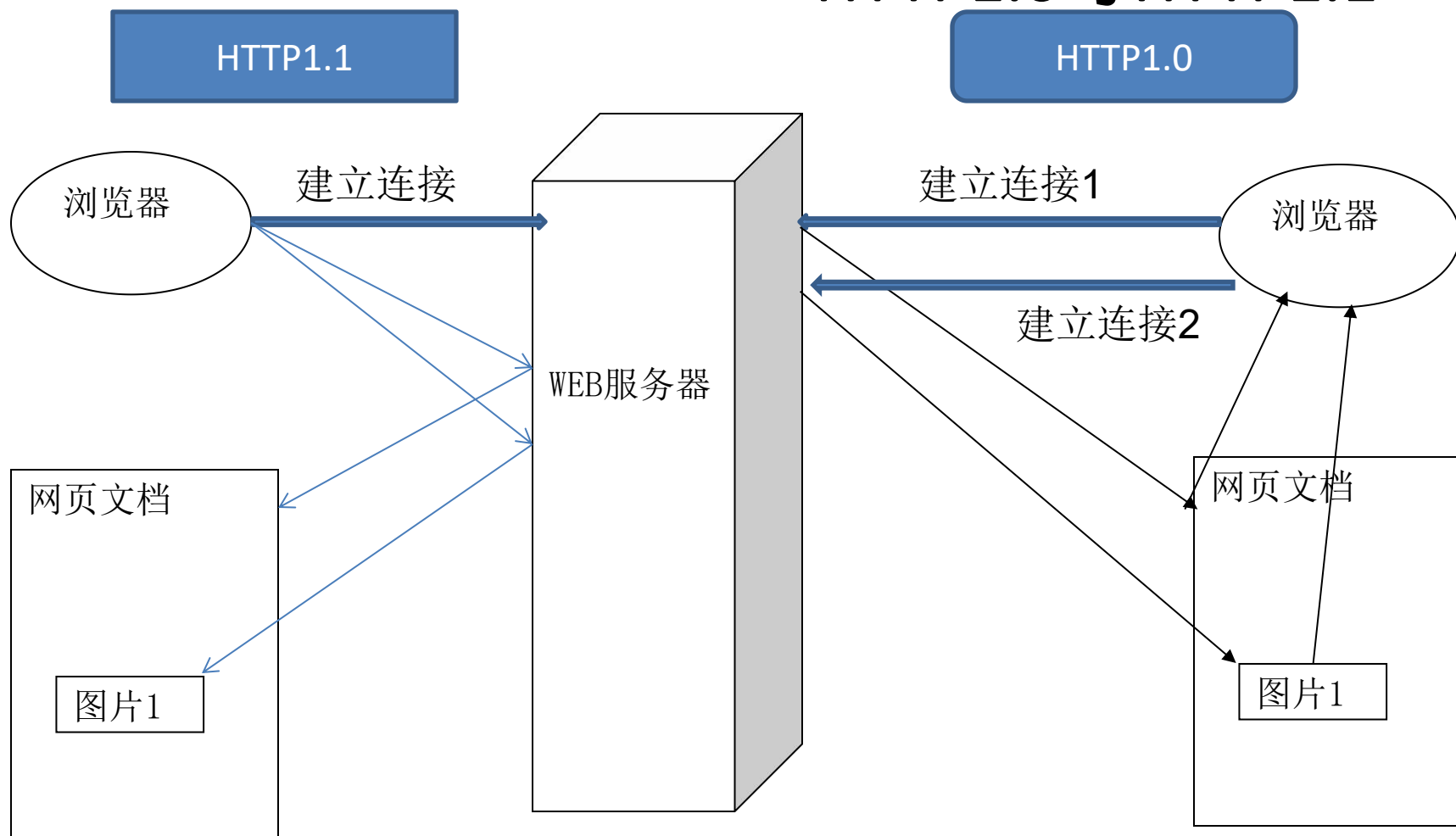


浏览器与WEB服务器的连接过程是短暂的，每次连接只处理一个请求和响应。对每一个页面的访问，浏览器与WEB服务器都要建立一次单独的连接。

浏览器到WEB服务器之间的所有通讯都是完全独立分开的请求和响应对。

HTTP相关知识(3)

HTTP1.0与HTTP1.1



HTTP请求(1)

概述

客户端连上服务器后，向服务器请求某个web资源，称之为客户端向服务器发送了一个HTTP请求。一个完整的HTTP请求包括如下内容：

1. 请求行
2. 若干消息头
3. 实体内容(可能没有)，如下所示：

```
POST /books/java.html HTTP/1.1
Accept: */*
Accept-Language: en-us
Connection: Keep-Alive
Host: localhost
Referer: http://localhost/links.jsp
User-Agent: Mozilla/4.0
Accept-Encoding: gzip, deflate
```

```
name=tom&password=123
```

← 请求行

请求行用于描述客户端的请求方式、请求的资源名称，以及使用的HTTP协议版本号

← 多个消息头

消息头用于描述客户端请求哪台主机，以及客户端的一些环境信息等

← 一个空行

← 实体内容

HTTP请求(2)

请求行

请求行中的**GET**称之为请求方式，请求方式有：

POST、GET、HEAD、OPTIONS、DELETE、TRACE、PUT

常用的有：GET、POST

用户如没有设置，默认情况下浏览器向服务器发送的都是get请求，例如在**浏览器直接输地址访问，点超链接访问等都是get**，用户如想把请求方式改为post，可通过更改表单的提交方式实现。

不管POST或GET，都用于向服务器请求某个WEB资源，这两种方式的**区别主要表现在数据传递上：**

如请求方式为GET方式，则可以在请求的URL地址后**以?的形式**带上交给服务器的数据，多个数据之间以&进行分隔，例如：

GET /mail/1.html?name=abc&password=xyz HTTP/1.1

GET方式的特点：在**URL**地址后附带的参数是有限制的，其数据容量通常不能超过**1K**。

如请求方式为POST方式，则可以在请求的实体内容中向服务器发送数据，

Post方式的特点：传送的数据量无限制。

HTTP请求(3)

请求消息头(请求头)

常用请求头(浏览器发送给服务器的数据,为了告诉服务器一些情况)

Accept: text/html,image/* 我支持的数据类型

Accept-Charset: utf-8 支持的数据的编码字符集

Accept-Encoding: gzip 支持的压缩方式

Accept-Language: en-us,zh-cn 支持的语言

Host: localhost:8888 请求的主机名

Referer: http://www.atguigu.com/index.jsp 发送请求的界面对应的url 防盗链

User-Agent: Mozilla/4.0 浏览器的相关信息

Connection: keep-Alive 请求完成后, 到是断开呢还是连接着

Date: Tue, 11 Jul 2000 18:23:51 GMT 发送请求的时间

Cookie: tt=123 //后面专门讲

HTTP请求(4)

请求体

- **GET请求**：没有请求体
- **POST请求**：如果form表单提交的方式为post,则表单项的数据以请求体的形式发送给服务器，没有大小限制

HTTP响应(1)

概述

一个**HTTP**响应代表服务器向客户端回送的数据，它包括：

一个状态行

若干消息头（响应头）

实体内容（响应体）

- **举例：**

HTTP/1.1 200 OK

Server: Microsoft-IIS/5.0

Date: Thu, 13 Jul 2000 05:46:53 GMT

Content-Length: 2291

Content-Type: text/html

Cache-control: private

<HTML>

<BODY>

.....

← 状态行

状态行用于描述服务器对请求的处理结果。

← 多个消息头

消息头用于描述服务器的基本信息，以及数据的描述，服务器通过这些数据的描述信息，可以通知客户端如何处理等一会儿它回送的数据。

← 一个空行

← 实体内容

代表服务器向客户端回送的数据

HTTP响应(2)

响应状态行

- 状态行

格式：HTTP版本号 状态码 原因叙述

举例：HTTP/1.1 200 OK

- 状态码用于表示服务器对请求的处理结果，它是一个三位的十进制数。响应状态码分为5类，如下所示：

| 状态码 | 含义 |
|---------|--|
| 100~199 | 表示成功接收请求，但要求客户端继续提交下一次请求才能完成整个处理过程 |
| 200~299 | 表示成功接收请求并已完成整个处理过程，常用200 |
| 300~399 | 为完成请求，客户需进一步细化请求。例如，请求的资源已经移动一个新地址，常用302 |
| 400~499 | 客户端的请求有错误，常用404 |
| 500~599 | 服务器端出现错误，常用 500 |

200 代表成功

302 代表让浏览器重新请求另一个资源

404 找不到请求的资源

500 服务器出问题了

HTTP响应(3)

响应消息头(响应头)

响应头(浏览器发送给服务器的数据,为了告诉服务器一些情况):

Location: /day05/index.jsp 告诉浏览器重新定向到指定的路径

Server:apache tomcat 使用的什么web服务器

Content-Encoding: gzip 告诉浏览器我传给你的数据用的压缩方式

Content-Length: 80 响应体的字节数

Content-Language: zh-cn 响应体数据的语言

content-type: text/html; charset=GB2312 响应体内容的类型 html/ css / image

Last-Modified: Tue, 11 Jul 2000 18:23:51 GMT 资源最后被修改的时间

Refresh: 1定时刷新

Content-Disposition: attachment; filename=aaa.zip 提示用户下载

Set-Cookie:SS=Q0=5Lb_nQ; path=/search 将cookie数据回送给ie

Expires: -1 告诉浏览器不要缓存起来

Cache-Control: no-cache

Pragma: no-cache

Connection: close/Keep-Alive 是否保持连接

Date: Tue, 11 Jul 2000 18:23:51 GMT 响应的时间

HTTP响应(4)

响应体

浏览器真正显示的数据，一般情况下为一个html格式的文本,也可以是其它类型的数据

