**HTTP协议**

1. 学习路线



1. 什么是HTTP协议

HTTP超文本传输协议 (HTTP-Hyper text transfer protocol)，是一个属于应用层的面向对象的协议，由于其简捷、快速的方式，适用于分布式超媒体信息系统。它于1990年提出，经过几年的使用与发展，得到不断地完善和扩展。它是一种详细规定了浏览器和万维网服务器之间互相通信的规则，通过因特网传送万维网文档的数据传送协议。

1. **HTTP的主要特点**

1.支持客户端/服务器模式。

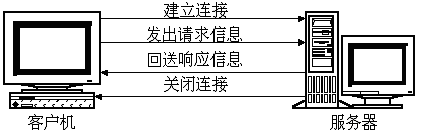
2.简单快速：客户向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有GET、POST、HEAD等。每种方法规定了客户与服务器联系的类型不同。由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快。

3.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。

4.无连接：无连接的含义是连接双方在完成交互后不维持连接状态，当客户端发送新的请求后则开启新的连接。

5.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。

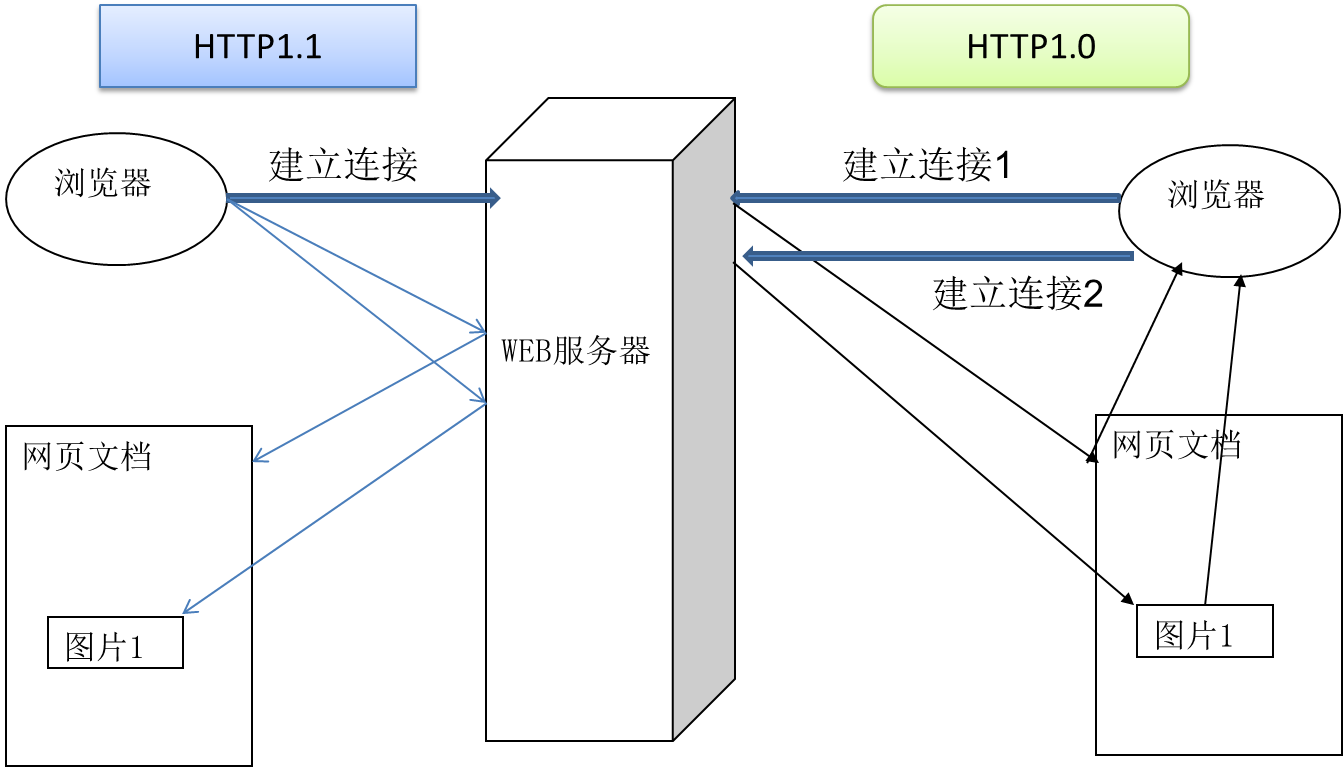
1. HTTP协议的会话方式
   1. 浏览器与服务器之间的通信过程要经历四个步骤



* 1. 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。
  2. 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。

1. HTTP1.0和HTTP1.1的区别

●在HTTP1.0版本中，浏览器请求一个带有图片的网页，会由于下载图片而与服务器之间开启一个新的连接；但在HTTP1.1版本中，允许浏览器在拿到当前请求对应的全部资源后再断开连接，提高了效率。



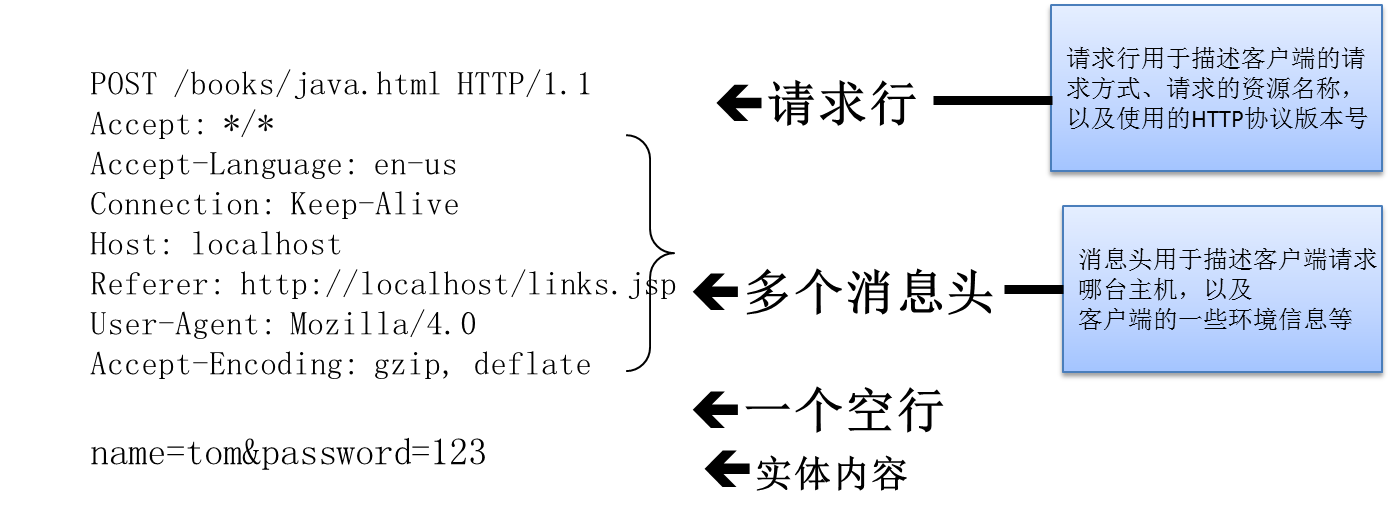
1. HTTP请求
   1. 客户端连上服务器后，向服务器发出获取某个Web资源的消息，称之为客户端向服务器发送了一个HTTP请求。一个完整的HTTP请求包括如下内容：

①请求行

URL地址中如果包含中文，浏览器会自动对中文字符进行编码之后再发送

②若干消息头(请求头)

③实体内容(请求体) 有可能没有



* 1. 请求方式

①HTTP中定义了7种请求方式：POST、GET、HEAD、OPTIONS、DELETE、TRACE、PUT。其中最常用的是GET和POST

②GET请求

[1]从字面意思来说，GET请求是用来向服务器端获取信息而发送的请求。

[2]没有特殊设置，默认情况下浏览器发送的都是GET请求，例如点击超链接、在浏览器地址栏直接输入地址访问。

[3]GET请求也可以向服务器端发送请求参数，形式是在URL地址后面加上?，请求参数名和值用=连接，多个请求参数之间使用&分隔。例如：

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

需要注意的是：GET方式所能够携带的数据是有限制的，其数据大小通常不能超过4K，不适于提交大量表单数据，故而在表单的提交方式中首选POST方式。

③POST请求

[1]POST请求的字面含义是向服务器端发送数据，仅在表单中设置method=”post”时，请求方式为POST方式[另外在Ajax应用中，可以指定请求方式为POST]。

[2]POST请求会将请求参数放在请求体中，而不是URL地址后面，并且发送数据的大小是没有限制的。

* 1. 请求消息头

①请求消息头简称请求头，用来向服务器报告浏览器端的一些基本信息。

②常用请求头

|  |  |
| --- | --- |
| 代码 | 说明 |
| 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.it315.org/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 | 对具体客户端所做的标记 |

* 1. 请求体

1. GET请求没有请求体

②POST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

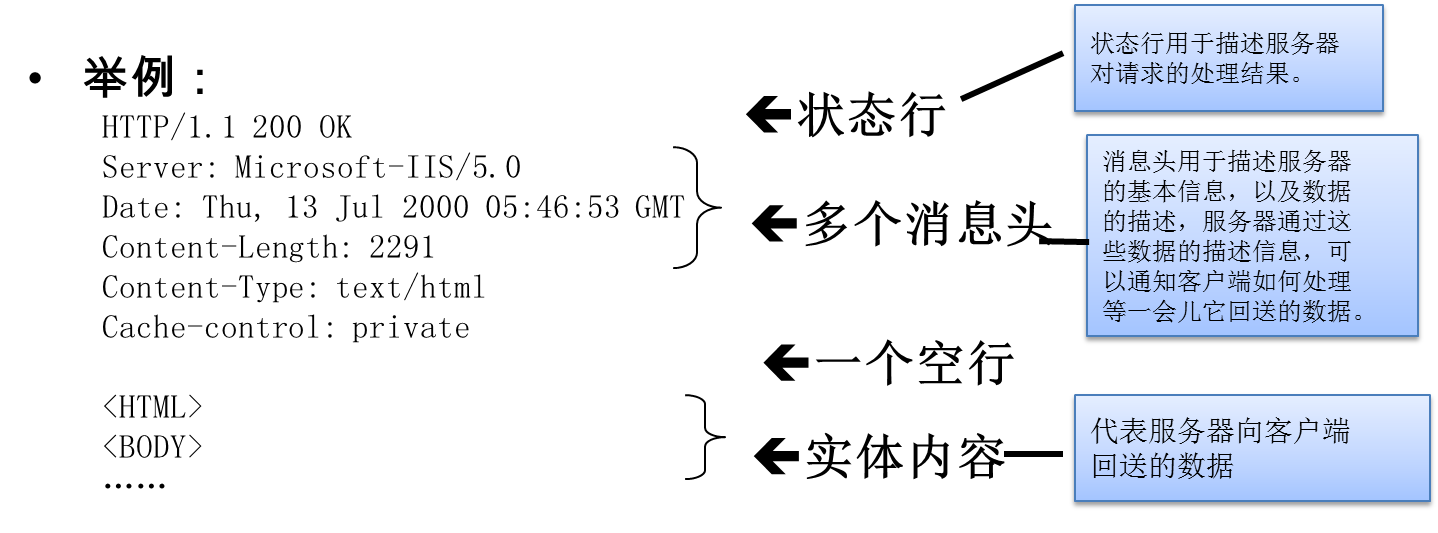


1. HTTP响应
   1. 所谓响应其实就是服务器对请求处理的结果，或者如果浏览器请求的直接就是一个静态资源的话，响应的就是这个资源本身。
   2. HTTP响应的组成

①响应状态行：包括协议版本、响应状态码、响应状态信息

②响应消息头：响应头

③实体内容：响应体



* 1. 最常见的响应状态码

|  |  |  |
| --- | --- | --- |
| 代码 | 表示 | 说明 |
| 200 | 成功 | 服务器已成功处理了请求。通常，这表示服务器提供了请求的网页。 |
| 302 | 重定向 | 代表让浏览器重新请求另一个资源 |
| 404 | 找不到 | 找不到请求的资源，但有时请求路径正确也返回404往往是由于Web应用有配置方面的问题，例如按照配置文件中指定的组件的全类名找不到指定的类。 |
| 500 | 错误 | 服务器内部错误，例如服务器端程序运行时抛出异常。 |

●响应状态码以2开头的通常表示成功。

●响应状态码以3开头的通常表示转移。

●响应状态码以4开头的通常表示无法访问，其中包括找不到资源或没有权限等。

●响应状态码以5开头的通常表示服务器端程序运行出错。

* 1. 响应消息头：简称响应头

●服务器发送给浏览器的数据，为了告诉浏览器一些情况。

|  |  |
| --- | --- |
| 代码 | 说明 |
| 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 |
| 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 | 文件下载 |
| 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 | 当HTTP1.1服务器指定 CacheControl = no-cache时，浏览器就不会缓存该网页 |
| Connection: close/Keep-Alive | 是否保持连接 |
| Date: Tue, 11 Jul 2000 18:23:51 GMT | 响应的时间 |

* 1. 响应体：服务器给出的响应结果的主体，通常是用来在页面上显示的。
  2. HTTP内容类型：当前响应体的数据类型。

①需要指出的是在浏览器和服务器之间传输的数据类型并非都是文本类型，还包括图片、视频、音频等多媒体类型。这些多媒体类型是使用MIME类型定义的。

②MIME的英文全称是"Multipurpose Internet Mail Extensions" 多功能Internet 邮件扩充服务。MIME类型的格式是“大类型/小类型”，并与某一种文件的扩展名相对应。

③常见的MIME类型

|  |  |
| --- | --- |
| 文件 | MIME类型 |
| 超文本标记语言文本 | .html,.html text/html |
| 普通文本 | .txt text/plain |
| RTF文本 | .rtf application/rtf |
| GIF图形 | .gif image/gif |
| JPEG图形 | .jpeg,.jpg image/jpeg |
| au声音文件 | .au audio/basic |
| MIDI音乐文件 | mid,.midi audio/midi,audio/x-midi |
| RealAudio音乐文件 | .ra, .ram audio/x-pn-realaudio |
| MPEG文件 | .mpg,.mpeg video/mpeg |
| AVI文件 | .avi video/x-msvideo |
| GZIP文件 | .gz application/x-gzip |
| TAR文件 | .tar application/x-tar |