

HTTP协议详解

1.HTTP协议重要性(能干啥?):

(先记住吧, 后面回来看也许就懂了, 好多东西没听说过)

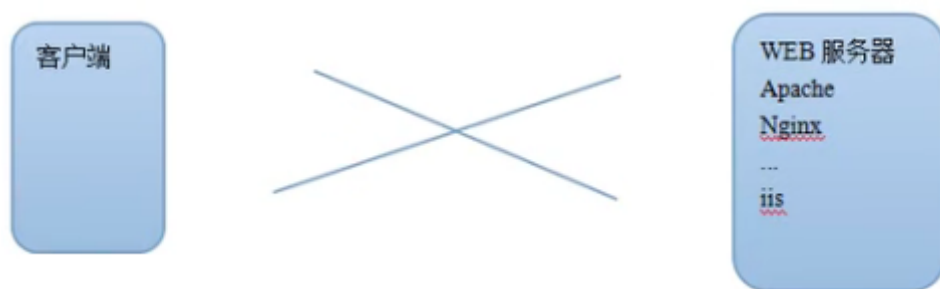
- 无论是以后用webservice,还是用rest做大型架构,都离不开对HTTP协议的认识.
甚至可以简化的说:
webservice = http协议 + XML
Rest= HTTP协议 + json
各种API(Application Programming Interface)应用程序接口也一般是用http+XML/json来实现的
[什么是API, 点击这里观看视频了解](#)
- 做采集,小偷站,也需要对HTTP协议有所了解。
- 以及Ajax,对HTTP协议有了解之后学习Ajax是非常容易理解的

2. 什么是协议

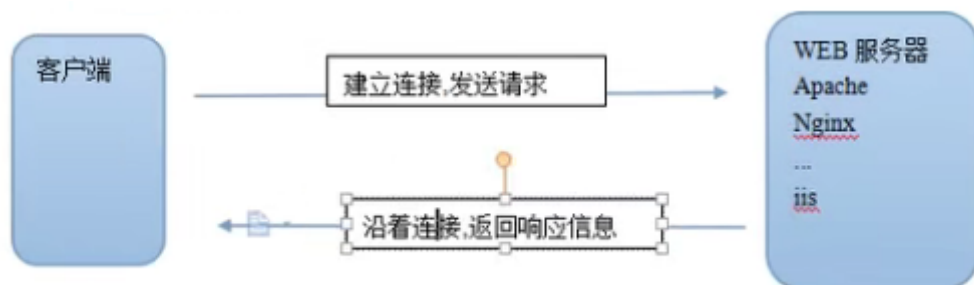
- 计算机中的协议和现实中的协议是一样的。一式双份/多份.双方多方都遵从共同的一个规范,这个规范就可以称为协议。
- 计算机之所以能全世界互通,协议是功不可没, 如果没有协议,计算机各说各话,根本谁也听不懂谁。

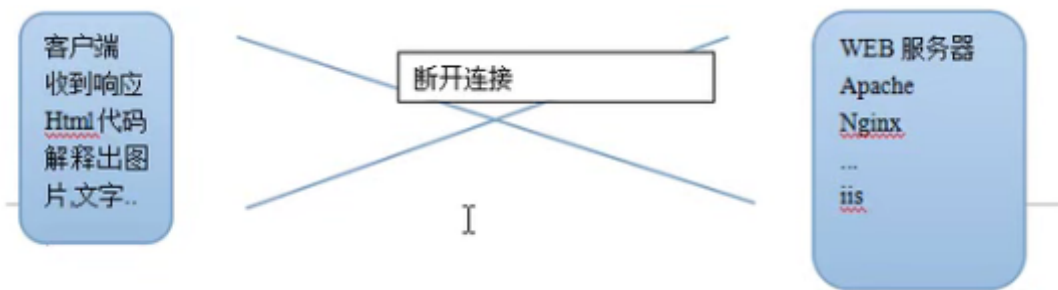
3. HTTP协议工作流程

1. 原始状态: 客户端和服务端之间没有关系



2. (什么是连接: 连接就是网络上的虚拟电路)

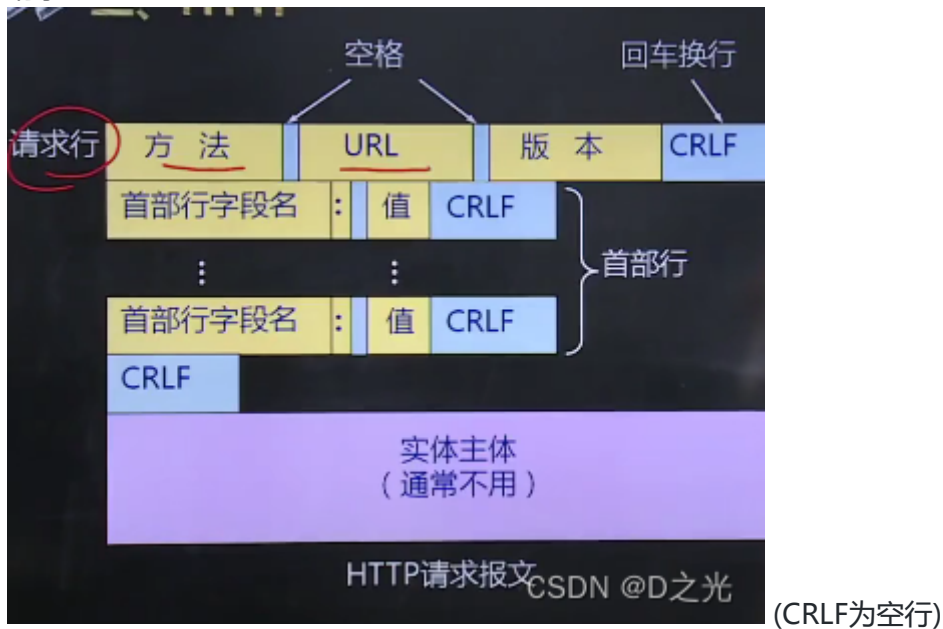




3. 断开连接

4. HTTP请求信息和响应信息的格式

1. 请求:



- 请求行
- 请求头信息
- 请求主体信息(可以没有)
- 头信息结束后和主题信息之间要空一行
- 请求行又分为三部分
 - 请求方法
 - 请求路径: URL的一部分
 - 所用的协议:

- 请求方法：GET POST ...

表 6-1 HTTP 请求报文的一些方法

方法（操作）	意义
OPTION	请求一些选项的信息
GET	请求读取由 URL 所标志的信息
HEAD	请求读取由 URL 所标志的信息的首部
POST	给服务器添加信息（例如，注释）
PUT	在指明的 URL 下存储一个文档
DELETE	删除指明的 URL 所标志的资源
TRACE	用来进行环回测试的请求报文
CONNECT	用于代理服务器

CSDN @D之光

- 注意:这些请求方法虽然HTTP协议里规定的,但WEB SERVER未必允许或支持这些方法. HEAD:和GET基本一致.只是返回内容. 比如我们只是确认一个内容,比如照片,还正常存在不需要返回照片的内容, 这时用 HEAD比较合适

例子:

POST /0606/02.php HTTP/1.1

Host: localhost

Content-type: application/x-www-form-urlencoded

Content-length: 5

Age=3

- 问:浏览器能发送HTTP协议, HTTP协议一定要浏览器来发送吗?
答:不是,HTTP既然是一种协议,那么只要满足这种协议,什么工具都可以发.



2. 响应:

- 状态码,状态文字:
状态码是用来反应服务器响应情况的.
最常见的如200 OK, 404 NOT FOUND
状态文字是用来描述状态码的, 便于人观察

HTTP响应报文：

HTTP状态码分类

状态码	作用	描述
1xx	信息提示	通告信息，可能还需要进一步交互
2xx	成功	成功完成客户请求的操作，并进行响应
3xx	重定向	表示资源已移走，需要向新URL发请求
4xx	客户端错误	由于客户端请求错误，无法成功响应
5xx	服务器端错误	由于服务器端错误，无法成功响应

- 最常用
 - 200 服务器成功返回网页
 - 301/2-永久/临时重定向(header 默认重定向是302)
 - 304 Not Modified—未修改
 - 307 重定向中保持原有的请求数据
 - 失败的状态码:
 - 404-请求的网页不存在
 - 503—服务器暂时不可用
 - 500-服务器内部错误

例子

```
HTTP/1.1 200 OK
Content-type: text/html
Content-length: 5
```

hello

5. socket编程发送GET请求

6. socket编程批量发帖

7. http协议模拟登陆发帖

8. referer头与防盗链

9. http缓存详解

10. http内容压缩

11. comet反向ajax
