

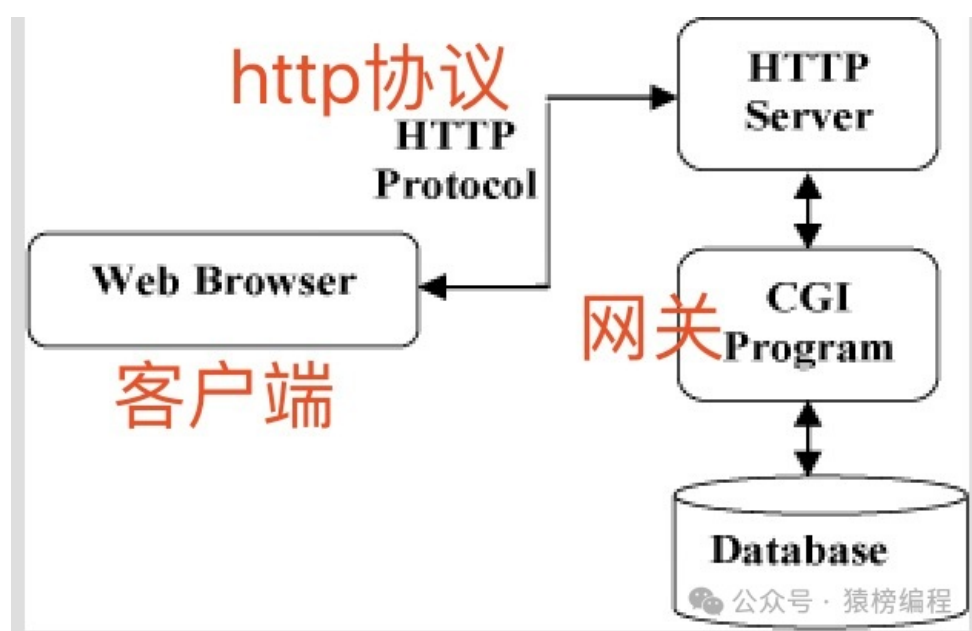
## HTTP协议是什么

1. HTTP协议（HyperText Transfer Protocol，超文本传输协议），是用于从万维网（WWW:World Wide Web）服务器传输超文本到本地浏览器的传送协议。它是因特网上应用最为广泛的一种网络传输协议，所有的WWW文件都必须遵守这个标准。
2. HTTP是一个基于TCP/IP通信协议来传递数据（HTML 文件, 图片文件, 查询结果等）。

## HTTP协议工作原理

1. HTTP协议工作于客户端-服务端架构上。浏览器作为HTTP客户端通过URL向HTTP服务端即WEB服务器发送所有请求。
2. Web服务器有：Apache服务器，IIS服务器（Internet Information Services）等。
3. Web服务器根据接收到的请求后，向客户端发送响应信息。
4. HTTP默认端口号为80，但是你也可以改为8080或者其他端口。

下图表展示了HTTP协议通信流程：



## HTTP协议三点注意事项：

- HTTP是无连接：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。
- HTTP是媒体独立的：这意味着，只要客户端和服务端知道如何处理的数据内容，任何类型的数据都可以通过HTTP发送。客户端以及服务器指定使用适合的MIME-type内容类型。
- HTTP是无状态：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。

# HTTP协议工作流程

## 一.请求流程

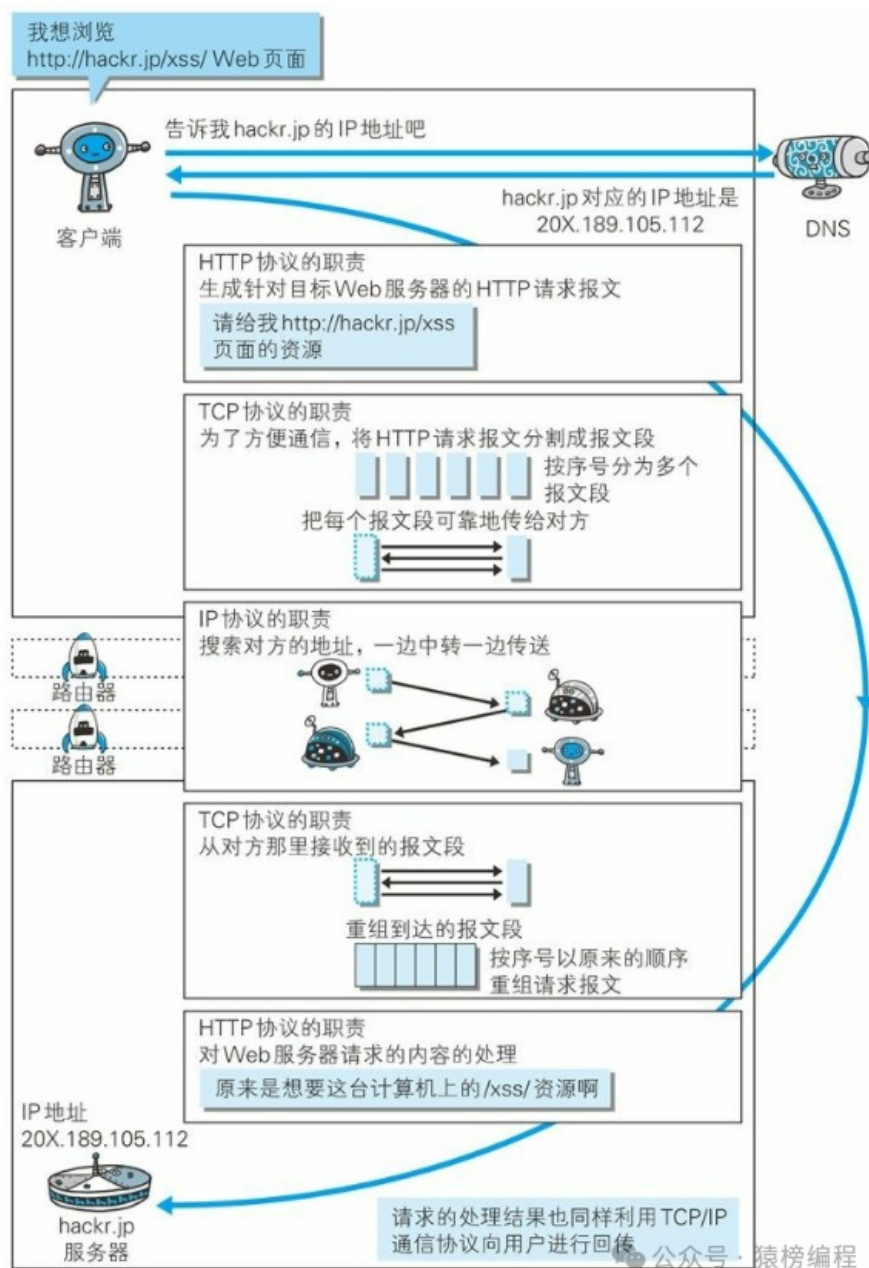
一次HTTP请求，其工作过程大概如下：

1. 用户在浏览器中键入需要访问网页的URL或者点击某个网页中链接；
2. 浏览器根据URL中的域名，通过DNS解析出目标网页的IP地址；

浏览器请求这个页面：`http://hackr.ip/index.html`  
在这一步，需要域名系统DNS解析域名hackr.ip,得主机的IP地址 20X.189.105.112。  
然后将上面结合本机自己的信息，封装成一个http请求数据包

3. 在HTTP开始工作前，客户端首先会通过TCP/IP协议来和服务端建立链接（TCP三次握手）
4. 建立连接后，客户机发送一个请求给服务器，请求方式的格式为：统一资源标识符（URL）、协议版本号，后边是MIME信息包括请求修饰符、客户机信息和内容。
5. 服务器接到请求后，给予相应的响应信息，其格式为一个状态行，包括信息的协议版本号、一个成功或错误的代码，后边是MIME信息包括服务器信息、实体信息和可能的内容。
6. 一般情况下，一旦Web服务器向浏览器发送了请求数据，它就要关闭TCP连接，然后如果浏览器或者服务器在其头信息加入了这行代码：`Connection:keep-alive`，TCP连接在发送后将仍然保持打开状态，于是，浏览器可以继续通过相同的连接发送请求。保持连接节省了为每个请求建立新连接所需的时间，还节约了网络带宽。

过程如下图



## 二.消息结构

HTTP是基于客户端/服务端 (C/S) 的架构模型, 通过一个可靠的链接来交换信息, 是一个无状态的请求/响应协议。HTTP使用统一资源标识符 (Uniform Resource Identifiers, URI) 来传输数据和建立连接。一旦建立连接后, 数据消息就通过类似Internet邮件所使用的格式[RFC5322]和多用途Internet邮件扩展 (MIME) [RFC2045]来传送。

### 2.1 URI和URL

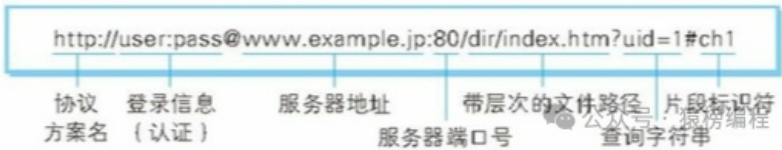
- URI

URI(Uniform Resource Identifier)是统一资源标识符,在某个规则下能把这个资源独一无二标示出来,比如人的身份证号

- 1. Uniform 不用根据上下文来识别资源指定的访问方式
  - 2. Resource 可以标识的任何东西
  - 3. Identifier 表示可标识的对象
- URL

统一资源定位符，表示资源的地点，URL时使用浏览器访问WEB页面时需要输入的网页地址

- 1. Uniform 不用根据上下文来识别资源指定的访问方式
  - 2. Resource 可以标识的任何东西
  - 3. Location 定位
- URL的格式



三.请求报文

客户端发送一个HTTP请求到服务器的请求消息包括以下格式：请求行（request line）、请求头部（header）、空行和请求数据四个部分组成，下图给出了请求报文的一般格式。



1.请求行（request line）

- 请求方法

GET 获取资源  
POST 向服务器端发送数据，传输实体主体  
PUT 传输文件  
HEAD 获取报文首部  
DELETE 删除文件  
OPTIONS 询问支持的方法  
TRACE 追踪路径  
协议/版本号  
URL

- 请求头

通用首部(General Header)  
请求首部(Request Header)  
响应首部(Response Header)

- 请求体

## 四.响应报文

HTTP响应也由四个部分组成，分别是：状态行、消息报头、空行和响应正文。

```
HTTP/1.1 200 OK
Date: Sat, 31 Dec 2005 23:59:59 GMT
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 122

<html>
<head>
<title>Wrox Homepage</title>
</head>
<body>
<!-- body goes here -->
</body>
</html>
```

状态行

消息报头

空行

下面的就是响应正文了

公众号·猿榜编程

## HTTP协议状态码

状态码负责表示客户端请求的返回结果、标记服务器端是否正常、通知出现的错误

### 状态码类别

类别	原因短语
1XX	Informational(信息性状态码)
2XX	Success(成功状态码)
3XX	Redirection(重定向)
4XX	Client Error(客户端错误状态码)
5XX	Server Error(服务器错误状态码)

### 2XX 成功

- 200(OK 客户端发过来的数据被正常处理)
- 204(Not Content 正常响应，没有实体)
- 206(Partial Content 范围请求，返回部分数据，响应报文中由Content-Range指定实体内容)

### 3XX 重定向

- 301(Moved Permanently) 永久重定向
- 302(Found) 临时重定向，规范要求方法名不变，但是都会改变

- 303(See Other) 和302类似，但必须用GET方法
- 304(Not Modified) 状态未改变 配合(If-Match、If-Modified-Since、If-None\_Match、If-Range、If-Unmodified-Since)
- 307(Temporary Redirect) 临时重定向，不该改变请求方法

4XX 客户端错误

- 400(Bad Request) 请求报文语法错误
- 401 (unauthorized) 需要认证
- 403(Forbidden) 服务器拒绝访问对应的资源
- 404(Not Found) 服务器上无法找到资源

5XX 服务器端错误


- 500(Internal Server Error)服务器故障
- 503(Service Unavailable) 服务器处于超负载或正在停机维护

HTTP请求方法

根据HTTP标准，HTTP请求可以使用多种请求方法。

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

序号	方法	描述
1	GET	请求指定的页面信息，并返回实体主体。
2	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
3	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。
4	PUT	从客户端向服务器传送的数据取代指定的文档的内容。
5	DELETE	请求服务器删除指定的页面。
6	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。
7	OPTIONS	允许客户端查看服务器的性能。
8	TRACE	回显服务器收到的请求，主要用于测试或诊断。

 公众号 · 猿榜编程

首部

通用首部字段

首部字段名	说明
Cache-Control	控制缓存行为
Connection	链接的管理
Date	报文日期
Pragma	报文指令
Trailer	报文尾部的首部
Transfer-Encoding	指定报文主体的传输编码方式
Upgrade	升级为其他协议
Via	代理服务器信息
Warning	错误通知

 公众号 · 猿榜编程

请求首部字段

首部字段名	说明
Accept	用户代理可处理的媒体类型
Accept-Charset	优先的字符集
Accept-Encoding	优先的编码
Accept-Language	优先的语言
Authorization	Web认证信息
Expect	期待服务器的特定行为
From	用户的电子邮箱地址
Host	请求资源所在的服务器
If-Match	比较实体标记
If-Modified-Since	比较资源的更新时间
If-None-Match	比较实体标记
If-Range	资源未更新时发送实体Byte的范围请求
If-Unmodified-Since	比较资源的更新时间(和If-Modified-Since相反)
Max-Forwards	最大传输跳数
Proxy-Authorization	代理服务器需要客户端认证
Range	实体字节范围请求
Referer	请求中的URI的原始获取方
TE	传输编码的优先级
User-Agent	HTTP客户端程序的信息

👤 公众号 · 猿榜编程

响应首部字段



首部字段名	说明
Accept-Ranges	是否接受字节范围
Age	资源的创建时间
ETag	资源的匹配信息
Location	客户端重定向至指定的URI
Proxy-Authenticate	代理服务器对客户端的认证信息
Retry-After	再次发送请求的时机
Server	服务器的信息
Vary	代理服务器缓存的管理信息
www-Authenticate	服务器对客户端的认证

公众号 · 猿榜编程

实体首部字段

首部字段名	说明
Allow	资源可支持的HTTP方法
Content-Encoding	实体的编码方式
Content-Language	实体的自然语言
Content-Length	实体的内容大小(字节为单位)
Content-Location	替代对应资源的URI
Content-MD5	实体的报文摘要
Content-Range	实体的位置范围
Content-Type	实体主体的媒体类型
Expires	实体过期时间
Last-Modified	资源的最后修改时间