

HTTP 是 HyperText Transfer Protocol 的缩写，表示“超文本传输协议”，所有的 Web 服务都必须遵守这个标准。

我们上篇文章很“轻松”地完成了一个web服务：为用户提供一个登录注册的网站，在我们平时开发过程中也“理所当然”地使用HTTP协议来承载web服务，但是当我们回顾HTTP的历史后，我们就会知道在这“理所当然”的背后，是几代人共同的努力，而其中最璀璨的大师必是Tim Berners-Lee爵士。

本篇文章来看一看HTTP前世今生。

## 一、web服务都基于HTTP协议工作

上一篇我们搭建了一个前后端分离的项目，前端提供登录、注册界面，供用户输入用户名密码，随后前端向服务端发起请求，服务端tomcat负责相应的接收和响应，从而完成了一个用户登录注册的web服务。

注意，这个过程使用了请求-应答模式，客户端主动发起请求，服务器被动回复请求。

用户的所有操作都在浏览器上完成，背后是一个黑盒，普通用户不需要知道它们之间是如何交互的，但是我们需要。

我们来简单看下基于HTTP协议的网页大概是如何工作的。

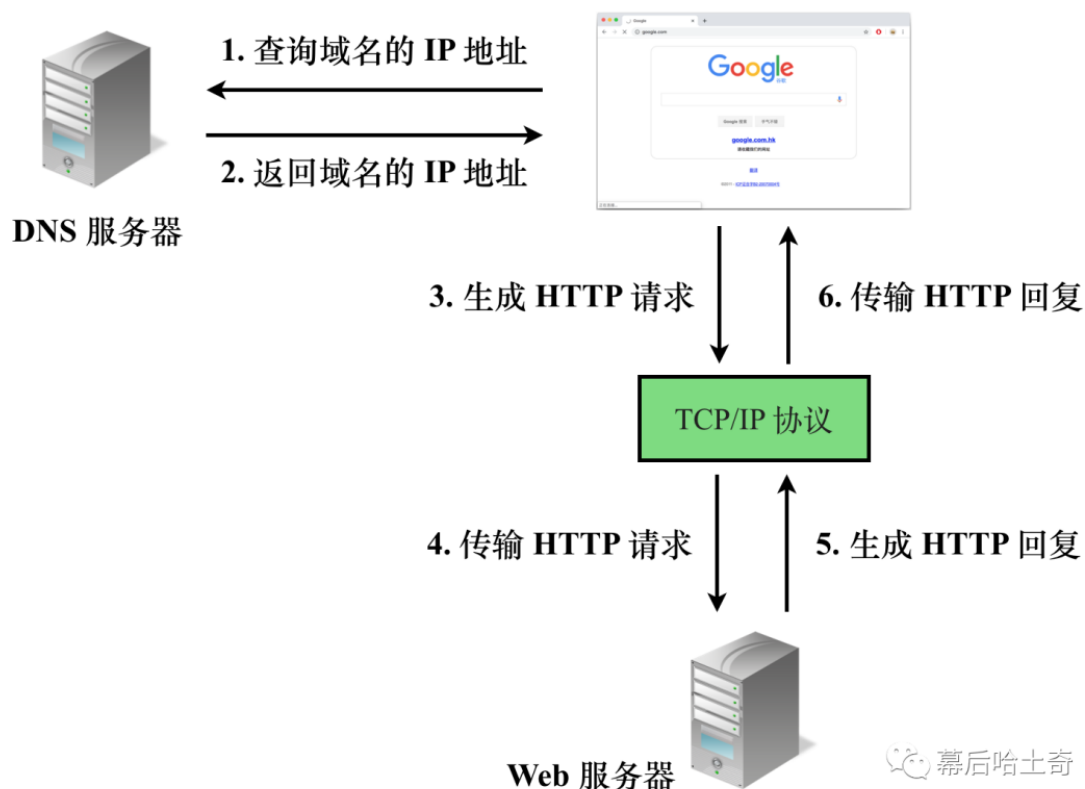
我们知道，依靠Web服务，我们才有了如今丰富多彩的网页，Web 的操作主要基于HTTP 协议，它是 OSI 第 7 层（应用层）协议。

设计 HTTP 最初的目的是为了提供一种发布和接收 HTML（超文本标记语言）页面的方法。

HTTP：超文本传输协议，是一种通信协议，它允许将超文本标记语言(HTML)文档从Web服务器传送到客户端浏览器。

简单来说，就是传输超文本的一种协议，什么是超文本呢？

超文本大概就是，不仅仅是文字，还有多媒体：视频、图片、动画等。还有就是有超链接，点击一个按钮可以调到另一个页面去，每个页面都可以这样跳来跳去，就形成一个网络。

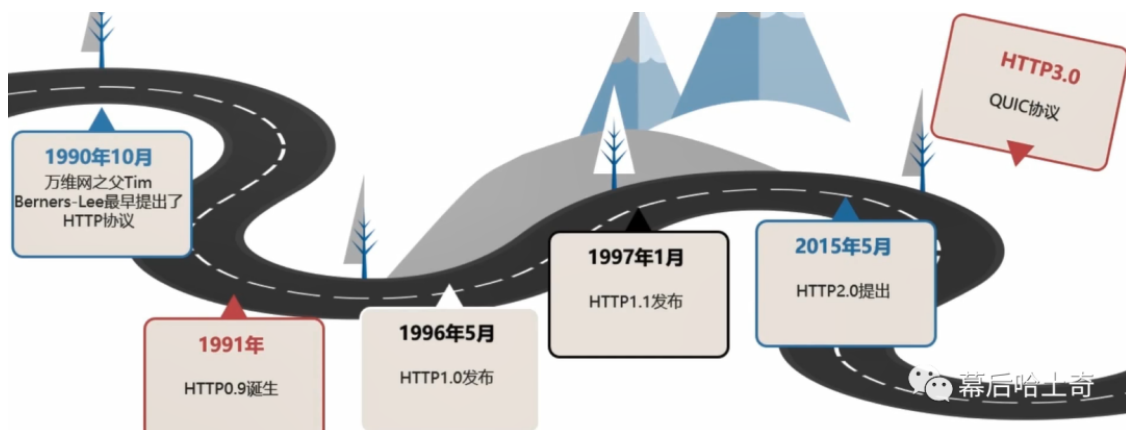


HTTP 基于 TCP 协议。当我们在浏览器的地址栏中输入一个 URL (Uniform Resource Locator 的缩写，表示“统一资源定位符”)，按下回车后浏览器会分析出 URL 上面的域名，然后通过 DNS 服务器查询出域名映射的 IP 地址，浏览器根据查询到的 IP 地址与 Web 服务器进行通信，而通信的协议就是 HTTP 协议。

说来说去，最关键的就是HTTP协议，HTTP协议的特点是什么，以及HTTP的发展史。

## 二、HTTP前世今生概述

HTTP版本迭代：



- HTTP 协议始于三十年前蒂姆·伯纳斯 - 李的一篇论文
- HTTP/0.9 是个简单的文本协议，只能获取文本资源；
- HTTP/1.0 确立了大部分现在使用的技术，但它不是正式标准；

- HTTP/1.1 是目前互联网上使用最广泛的协议，功能也非常完善；
- HTTP/2 基于 Google 的 SPDY 协议，注重性能改善，但还未普及；
- HTTP/3 基于 Google 的 QUIC 协议，是将来的发展方向。

我们来一个一个来唠叨唠叨。

### 三、HTTP/0.9版本

首先就是诞生于1991年的HTTP/0.9版本。

1990年11月，CERN 成功研制出世界上第一台 Web 服务器和 Web 浏览器。1991年，Tim 发表了一篇 HTTP 协议文章，后来被称为 HTTP/0.9。实际上这篇文章比较简短，很难被称为标准的协议，只是 Tim 个人的作品，解释了程序的实现过程。不过鉴于 Tim 对互联网的开创性贡献，后人将其HTTP的文章作为HTTP的第一版标准。

Tim 完成了3项 WWW 构建技术，分别是：

- URI，统一资源标识符，作为互联网上的唯一标识。
- HTML，超文本标记语言，描述超文本。
- HTTP，超文本传输协议，传输超文本。

最早的版本协议非常简单，特性只有如下：

- 内容非常简单，只有一个命令GET
- 没有HEADER等描述数据的信息
- 服务器发送完毕，就关闭TCP连接（一个HTTP请求在一个TCP连接中完成）
- 协议规定，服务器只能回应HTML格式的字符串，不能回应别的格式。也没有状态码来区分正确和错误消息

比如发起一个GET请求：

```
GET /index.html
```

服务器响应：

```
<html>
  <body>Hello World</body>
</html>
```

那时候是互联网初期，计算机的处理能力包括网速等等都很弱，所以 HTTP 也逃脱不了那个时代的约束，因此设计的非常简单，而且也是纯文本格式。

李老当时的想法是文档存在服务器里面，我们只需要从服务器获取文档，因此只有“GET”，也不需要啥请求头，并且拿完了就结束了，因此请求响应之后连接就断了。

这就是为什么 HTTP 设计为文本协议，并且一开始只有“GET”、响应之后连接就断了的原因了。

在我们现在看来这协议太简陋了，但是在当时这是互联网发展的一大步！一个东西从无到有是最困难的。

这时候的 HTTP 还没有版本号的，之所以称之为 HTTP / 0.9 是后人加上去了，为了区别之后的版本。

## 四、HTTP/1.0版本

人们的需求是无止尽的，随着图像和音频的发展，浏览器也在不断的进步予以支持。需求促使添加各种特性来满足用户的需求，经过了一系列的草案，HTTP/1.0 于 1996 年正式发布。HTTP 1.0 扩展了0.9版，其中主要增加了几个变化：

- 在请求中加入了HTTP版本号，如：GET /coolshell/index.html HTTP/1.0
- HTTP 开始有 header了，不管是request还是response 都有header了
- 增加了HTTP Status Code 标识相关的状态码
- 还有 Content-Type 可以传输其它类型的文件了

可以看到引入了新的方法，填充了操作的语义，像 HEAD 还可以只拿元信息不必传输全部内容，提高某些场景下的效率。

引入的响应状态码让请求方可以得知服务端的情况，可以区分请求出错的原因，不会一头雾水。

引入了头部，使得请求和响应更加的灵活，把控制数据和业务实体进行了拆分，也是一种解耦。

新增了版本号表明这是一种工程化的象征，说明走上了正途，毕竟没版本号无法管理。

引入了 Content-Type，支持传输不同类型的数据，丰富了协议的载体，充实了用户的眼球。

举个例子，请求如下：

```
GET / HTTP/1.0
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5)
Accept: */*
```

可以看到，这个格式与0.9版有很大变化。第一行是请求命令，必须在尾部添加协议版本（HTTP/1.0）。后面就是多行头信息，描述客户端的情况。

客户端请求的时候，可以使用Accept字段声明自己可以接受哪些数据格式。上面代码中，/表示客户端声明自己可以接受任何格式的数据。

服务器的回应如下：

```
HTTP/1.0 200 OK
Content-Type: text/plain
Content-Length: 137582
Expires: Thu, 05 Dec 1997 16:00:00 GMT
Last-Modified: Wed, 5 August 1996 15:55:28 GMT
Server: Apache 0.84

<html>
  <body>Hello World</body>
</html>
```

回应的格式是"头信息 + 一个空行（\r\n） + 数据"。其中，第一行是"协议版本 + 状态码（status code） + 状态描述"。

关于字符的编码，1.0版规定，头信息必须是 ASCII 码，后面的数据可以是任何格式。因此，服务器回应的时候，必须告诉客户端，数据是什么格式，这就是Content-Type字段的作用。

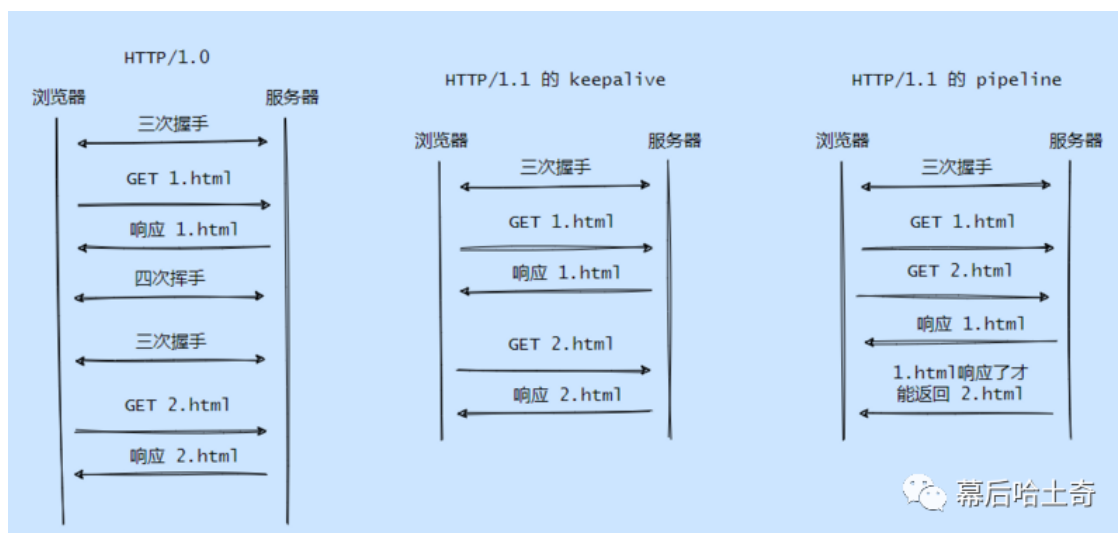
## 五、HTTP/1.1版本

HTTP/1.1 是目前正在使用的主流版本。

HTTP/1.1 主要解决了HTTP 1.0的网络性能的问题，以及增加了一些新的东西：

- 新增了连接管理即 keepalive，允许HTTP重用TCP链接。重用TCP链接可以省了每次请求都要进行的TCP三次握手的巨大开销。
- 支持 pipeline，无需等待前面的请求响应，即可发送第二次请求。可以减少整体的响应时间。（注：非幂等的POST 方法或是有依赖的请求是不能被pipeline化的）
- 允许响应数据分块（chunked），即响应的时候不标明Content-Length，客户端就无法断开连接，直到收到服务端的 EOF，利于传输大文件。
- 新增缓存的控制和管理。
- 协议头增加了 Language, Encoding, Type 等等头，让客户端可以跟服务器端进行更多的协商。

- 还正式加入了一个很重要的头—— HOST，这样的话，服务器就知道你要请求哪个网站了。因为可以有多个域名解析到同一个IP上，要区分用户是请求的哪个域名，就需要在HTTP的协议中加入域名的信息，而不是被DNS转换过的IP信息。



## 六、HTTP/2和HTTP/3版本

HTTP协议的一大缺点是：明文传输，无法验证通信双方的身份，也不能判断报文是否被篡改，总之是一个不安全的协议。

而HTTPS基于HTTP实现了数据加密、身份认证、防篡改等机制，实现了安全可靠的传输，所以现在绝大多数的网站都已迁移至HTTPS。

由于 HTTPS 已经在安全方面做的非常好了，所以 HTTP/2 的唯一目标就是改进性能，比如有以下改进：

- 兼容HTTP/1，这一点是前提
- 利用 HPACK 算法压缩头部，减少“大头儿子”的出现
- 不再使用肉眼可见的ASCII码，而是向下层的TCP/IP协议“靠拢”，全面采用二进制格式
- 等等.....

现在Top 1000 的网站中已经有超过 40% 运行在了 HTTP/2 上，包括知名的 Apple、Facebook、Google、Twitter 等等。仅用了四年的时间，HTTP/2 就拥有了这么大的市场份额和巨头的认可，足以证明它的价值。

而 HTTP/3 更是在 HTTP/2 的基础上更进一步，彻底解决底层TCP带来的根本问题

\*\*： \*\*

- HTTP/3让 HTTP 跑在 QUIC 上而不是 TCP 上
- 真正“完美”地解决了“队头阻塞”问题，这一点在HTTP/2中未得到根本解决，根本原因是HTTP/2底层TCP协议固有特性
- QUIC 是一个新的传输层协议，建立在 UDP 之上，实现了可靠传输

不过 HTTP/3 目前还处于草案阶段，正式发布前可能会有变动。

值得注意的是，主流的浏览器 Chrome、Firefox 等都公开宣布只支持加密的 HTTP/2，因此实际落地时，**加密是\*\*HTTP/2和HTTP/3基本配置\*\***，HTTP的明文传输不复存在。

本篇文章由于所存储的知识十分有限，还不能展开对 HTTP/2 和 HTTP/3 学习，后续我们在学习完HTTPS后再来学习。

所以，当前的我们，先把目光留在 HTTP/1.1 上，这是目前绝大多数网站的主流选择。