

02 HTTP 初窥门径

更新时间：2020-08-12 10:42:22



“

读书而不思考，等于吃饭而不消化。——波尔克

”

历史总是存在着一些无法解释的现象，1955 年对于计算机行业来说，是一个伟大的年份，这一年，比尔·盖茨和乔帮主出生。前者一手缔造了MicroSoft，让计算机走进了千家万户。后者创办了Apple，创造了风靡全球的iPod、iPhone。这一年，英国诞生了一个小男孩 Tim Berners-Lee，此子天赋异禀，智商超群。



Tim 在大学毕业后，加入了欧洲核子研究组织(CERN)，成为了一名光荣的程序猿。这哥们在工作的时候可能觉得无聊，总想搞点大事情，他想把 CERN 所有的电脑的数据进行共享。这样的话，大家就可以偷偷看到其他人的各种八卦，会大大的丰富大家的业余生活~~~

这个理由是本人瞎猜臆造，如有巧合，纯属雷同.....

终于有一天，Tim 发明了一种叫做万维网(World Wide Web，也就是我们熟悉的 WWW)的东西，通过 WWW 技术可以将所有的电脑资源共享。

我们今天能够通过慕课网学习各种知识，也是多亏了 Tim 的万维网技术，请让我们对他表示崇高的敬意。

为了实现资源共享，Tim 发明了一种叫做 超文本(hypertext) 的东东。该技术包含了三部分内容：

- 定义资源路径的 URI；
- 传输资源的通信协议；
- 创建网页的语言。

他用这些技术在乔帮主发明的 NeXT 电脑作为服务器，创造了人类的第一个网站info.cern.ch，此网站现在还能正常访问。



为什么要说这些呢？因为我们今天要介绍的主角 HTTP 协议 就是上面提到的超文本(hypertext)中的传输资源的通信协议~

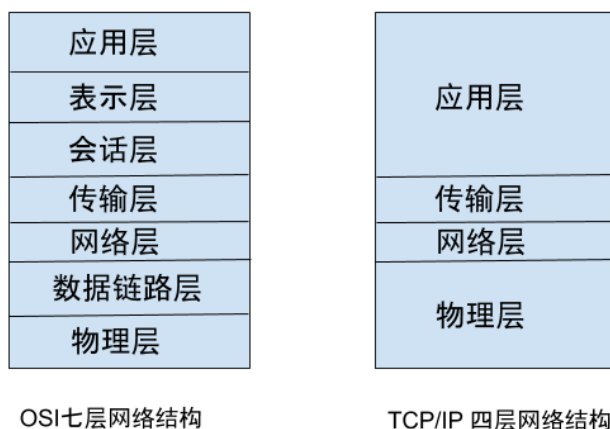
HTTP版本

自 HTTP 协议发明到现在，经过了几次版本修改，分别是 HTTP/0.9，HTTP/1/0，HTTP/1.1 以及 HTTP/2。现在市面上主要还是 HTTP/1.1，我们本文也主要介绍的是该版本。

TCP/IP协议

在学习 HTTP 协议之前我们先来了解下 TCP/IP 协议。可能大家要问了，我们在学习 HTTP，为什么要扯到 TCP/IP 协议呢？你是不是在凑字数？骗子！！！非也，非也，我们要理解 HTTP，就必须简单了解一下 TCP/IP 协议，它是 HTTP 的基础，地基打稳了房子才能结实嘛！

大家在工作过程中可能经常听到 OSI 七层网络结构以及 TCP/IP 四层网络结构等，但是这些都是什么呢？你平时工作中是否被搞晕过呢？



左边的是大名鼎鼎的国际标准化组织 **ISO** 指定的网络结构模型。但是实际上大家使用的都是右边的 **TCP/IP** 四层网络结构。

ISO 指定网络标准的时候，**TCP/IP** 已经成为了事实上的标准，所以造成了一种奇葩现象。国际协会指定了标准，但是大家都不用~~

下面我们简单的聊一下 TCP/IP 四层结构中每一层的作用。

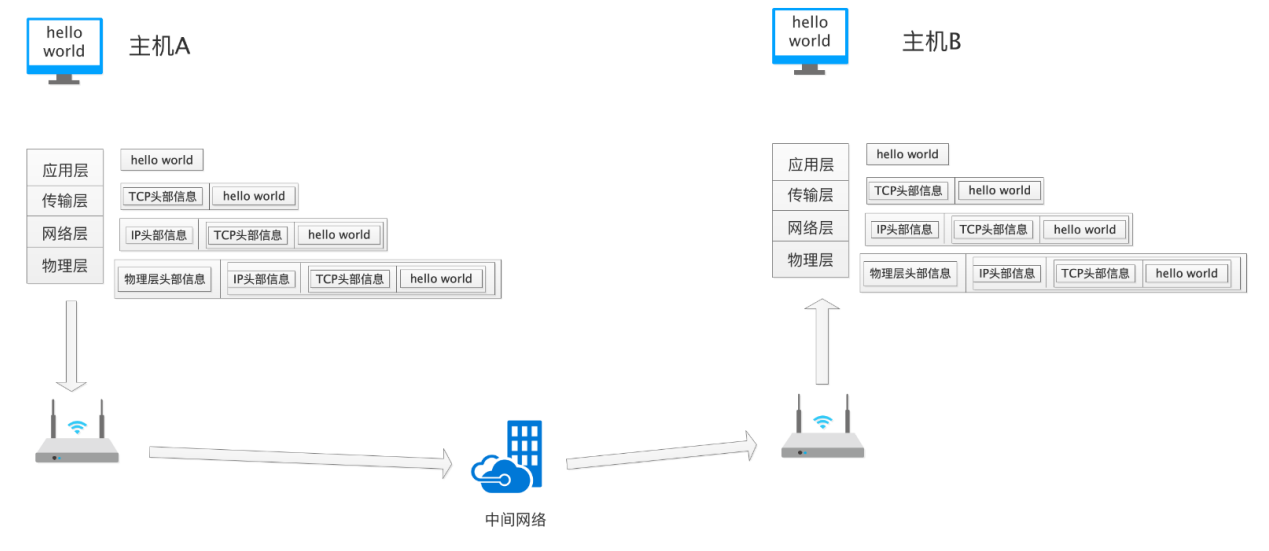
应用层： 这一层就是上层应用使用的协议。比如 **HTTP** 协议，**FTP** 协议，**SMTP** 协议等。

传输层： 从物理层到传输层这三层都是负责建立网络连接，发送数据。他们并不关心应用层使用什么协议。传输层使用的就是 **TCP** 和 **UDP** 协议。

网络层： 这一层负责选择 **路由路径**。条条大路通罗马嘛，所以从相同的 **出发地** 到相同的 **目的地** 也会有很多种路径，网络层就负责选择一条数据通行的路径。

物理层： 这一层负责数据的发送。这是最底层的网卡负责的。网卡把数据转换成 **高/低电平**，然后通过网线发送出去。

下面我们以一个实际的例子来说明这个过程：



上图是 主机A 向 主机B 发送 **hello world** 的过程。用户发出一个请求之后，从应用层开始，一直到物理层，每一层都会被加上盖层所属的附加信息；在接收端，每经过一层都会去掉该层的附加信息，然后交给上层处理。

是不是很像洋葱，一层又一层.....

应用层把待发送的信息 **hello world** 使用自己的协议进行封装，然后调用传输层的接口，以此类推，最终数据传递给了物理层。每一层都会加上自己特有的一些标志信息。物理层最后把要发送的数据(包含应用层真正想发送的**hello world** 以及每个层自己增加的标识信息)转换为高低电平发送出去，接收端收到之后进行一个**逆向**的解析过程，最后主机B收到了 **hello world**。

上面是一个简单的描述，但是整体的原理就是这样的~

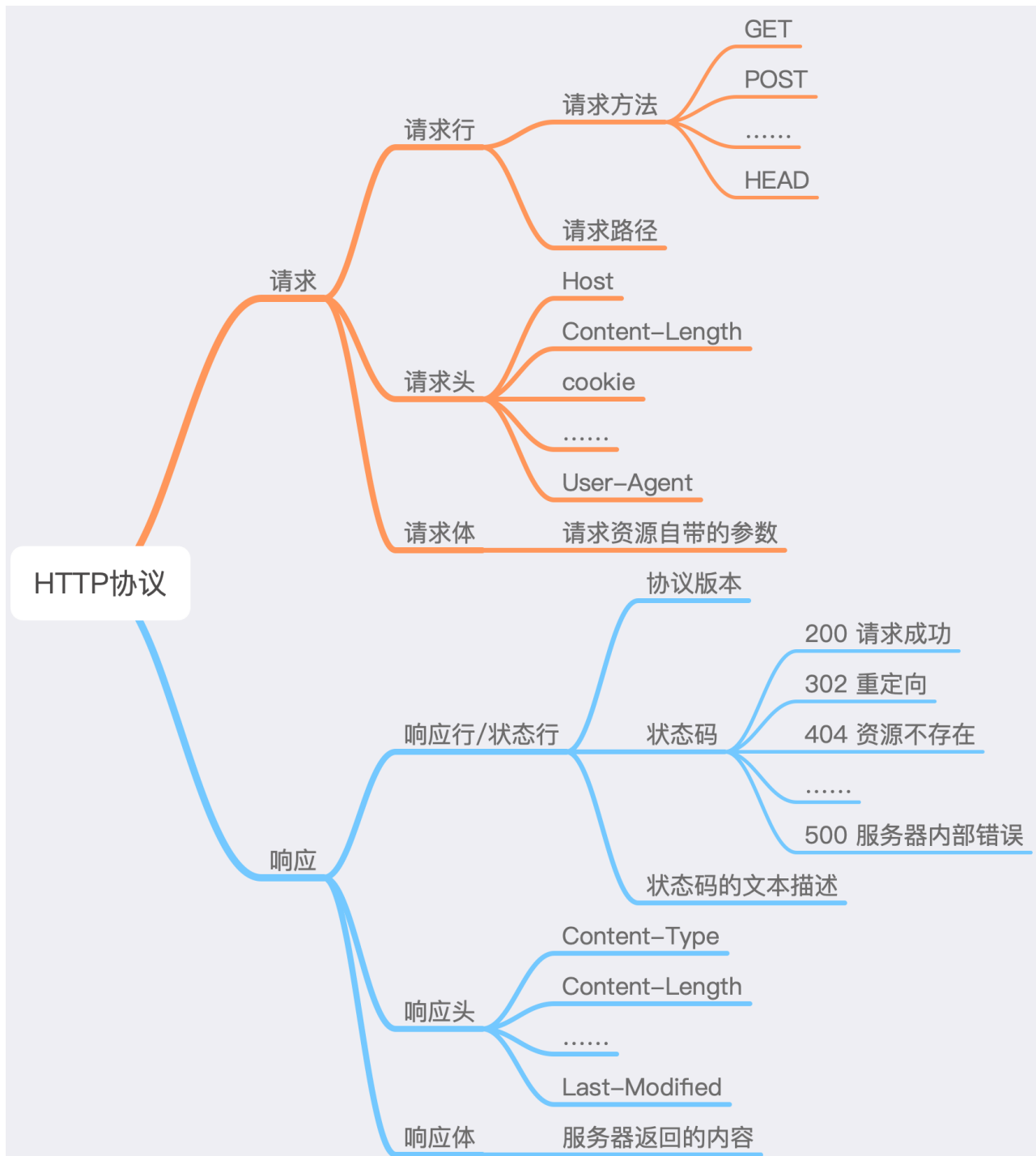
大家不要被每一层的概念所迷惑。这些层是人为划分出来的概念。就是为了写代码实现的时候比较方便。只需要定义每个层的接口，上一层调用下一层的接口就行了，不必关心具体实现。这也是软件设计中的一种理念。

HTTP协议

上面简单的说了一下网络的结构，这些东西是 **HTTP** 运行的**基础**。下面我们聊一下本文的重点：**HTTP** 协议相关的内容。

HTTP协议总览

我们先看一下 HTTP 协议的总览图，如下：



HTTP协议结构



HTTP报文格式

图中可以看到，**HTTP** 协议分为三个部分，其中报文头部和报文主体之间使用空行进行分隔。

HTTP 是一个 **Request-Response** 的协议。



1. 客户端发出 HTTP 请求(request);
2. 服务器接收请求，处理请求，然后返回响应结果(response);
3. 客户端接收响应结果，进行其它处理。

下面我们针对 HTTP 的请求和响应分别学习他们的报文结构。

HTTP 请求报文

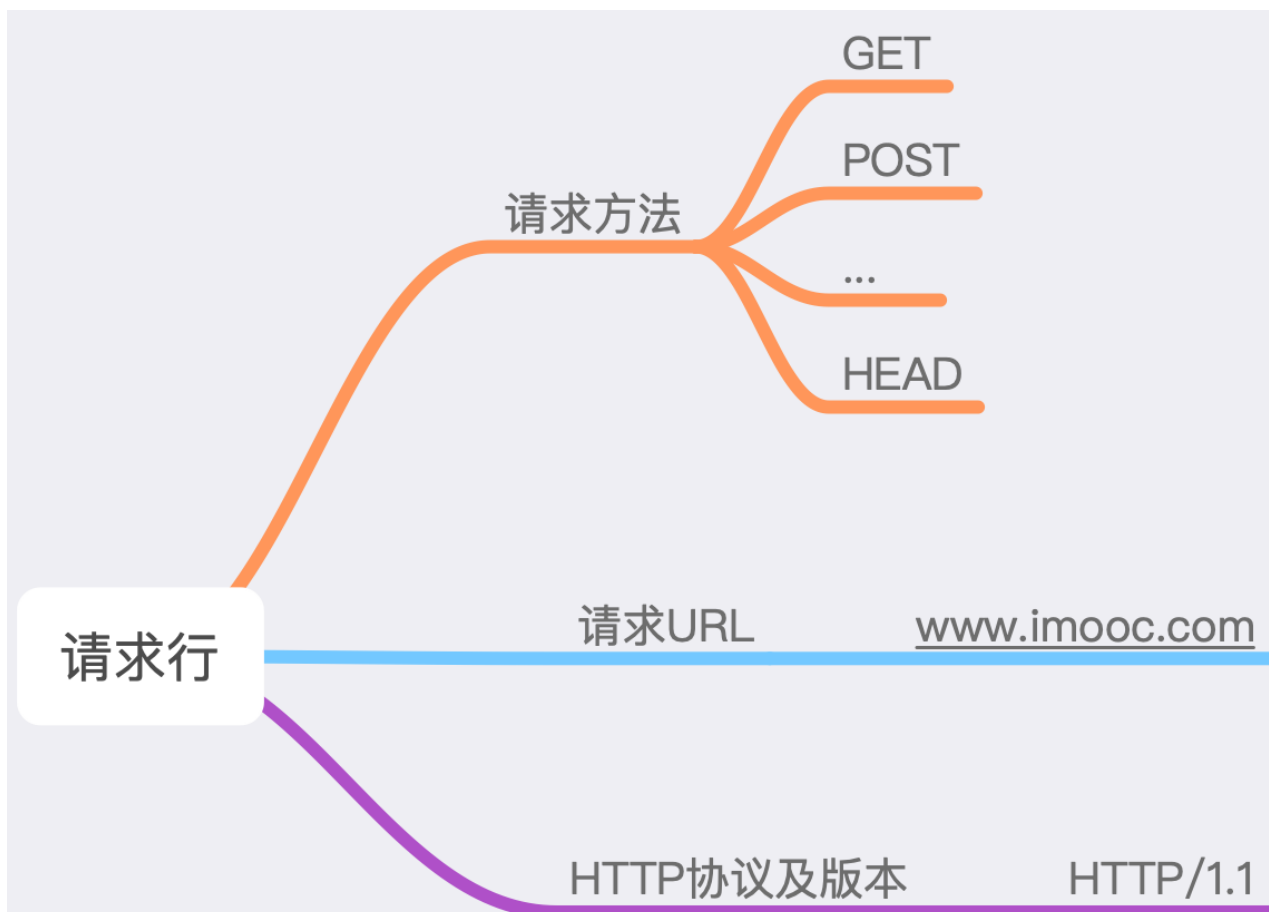
报 文 头 部	状态行	
	响 应 头	响应首部字段1
		响应首部字段2
		响应首部字段n
空行		
响应内容		

HTTP响应报文格式

我们可以看到， HTTP 请求报文分为三部分：

- 请求行
- 请求头
- 请求体

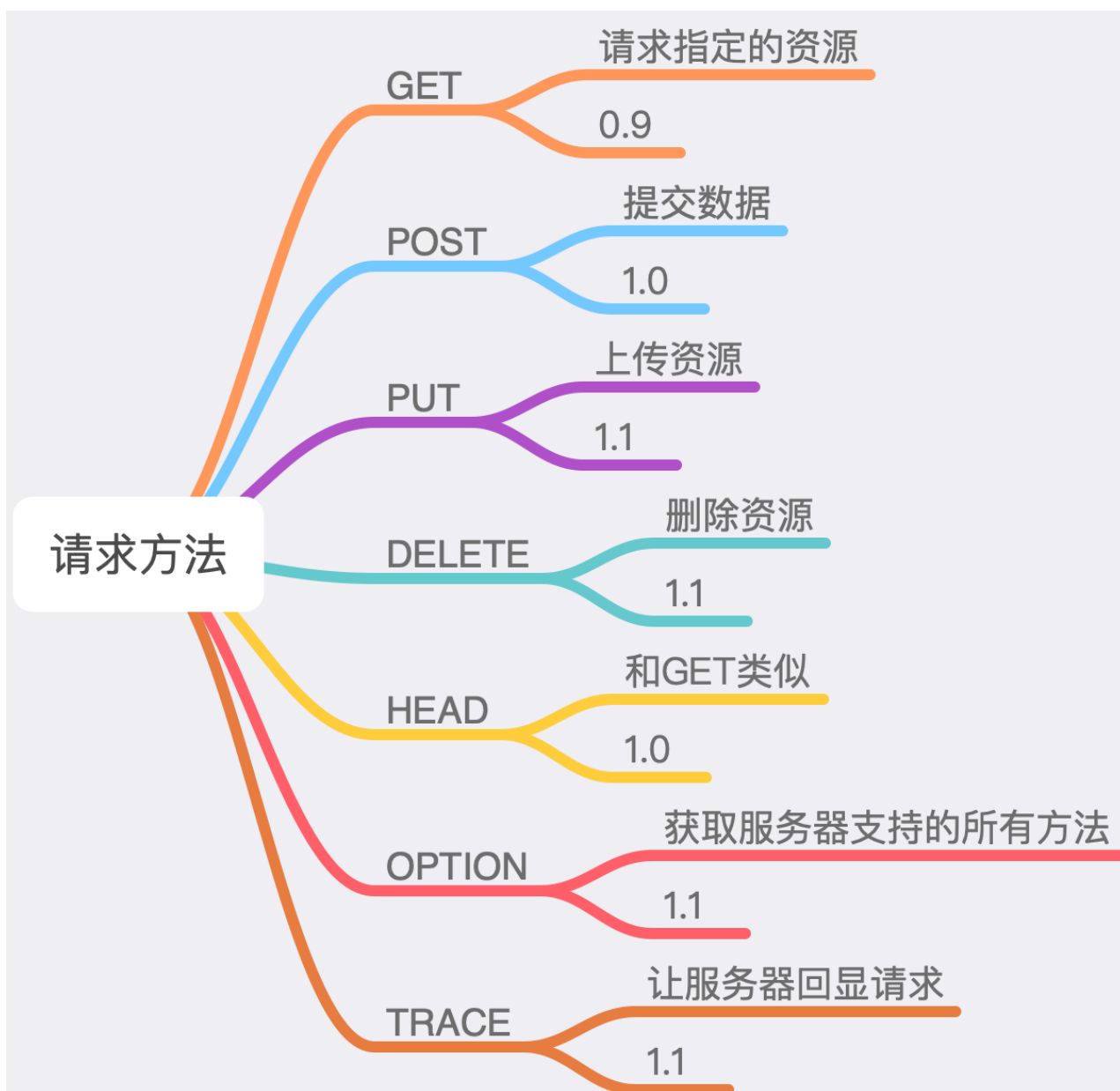
请求行



请求行包含了三部分，分别是请求方法，请求的 **URL** 以 **HTTP** 协议版本。

请求方法

HTTP/1.1 支持多种请求方法 (**method**)，如下图：



图中的数字表示的是支持该方法的最低 **HTTP** 协议版本，例如我们从图中可以看到 **HTTP/0.9** 只支持 **GET** 方法。在工作中我们最常用到的是 **GET** 和 **POST** 方法。下面我们简单来聊一下这两种方法的区别。

}