

# HTTP详解2

《HTTP详解1》中我们介绍了**URI**，并也说了在浏览器接收到 `http:` 开头的、格式正式的**URL**之后就开始根据**HTTP协议**拼接HTTP请求的消息。

关于**HTTP协议**我们在上一篇文章中也说明了，就是一种规则，那么什么叫做HTTP请求呢？

我们想要访问一个网页，就需要向对应的服务器请求这个网页。就像你想去你的邻居家借扳手，最好先打个电话问一下对方家里是否有你需要的扳手。我们可以将这个电话理解为这个请求，而电话的内容就是请求的消息。

整个互联网传输过程中，从宏观上看，基本上都是发送消息和接收消息。

而**HTTP协议**规定了一种消息格式，它告诉所有浏览器和服务端要按照它规定的格式来写这个消息，这样写出来的消息才是有效的，**HTTP协议**规定的消息格式也只有按照**HTTP协议**才能够解读。

简单来说，HTTP请求的消息就是一段文字。不过我们对这段文字的每个部分都做了详细的规定，以便于读取对应的内容。我们也把这段文字叫做**报文**（准确的说叫做**HTTP报文**）。

发送消息这个整体环节（即从解析**URL**到编写**HTTP报文**，再将**HTTP报文**发送出去的这个过程）我们一般称为“**请求（Request）**”，这个环节发送出去的内容称为**请求报文**。

**HTTP报文**格式如下图所示：

```
1  <方法><空格><URI><空格><HTTP版本>
2  <字段名>:<字段值>
3  <字段名>:<字段值>
4  <字段名>:<字段值>
5  .....
6  <字段名>:<字段值>
7  <空行>
8  <消息体>
```

上面的写法可能并不是很容易理解，那么我们就代入进去，看看具体的是一个什么样子的。比如说我们想要访问<http://www.fangqibiancheng.com/dir/index.html>这个网页，它的**HTTP报文**可能是这个样子的：

```
1 GET /dir/index.html HTTP/1.1
2 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng
  ,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
3 Accept-Encoding: gzip, deflate
4 Accept-Language: zh-CN,zh;q=0.9
5 Connection: keep-alive
6 Host: www.fangqibiancheng.com
7 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_3)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.113 Safari/537.36
8
```

首先第一行第一个部分（第一行的第一个空格之前的内容）为“**方法（Method）**”，在例子中，这部分写了“**GET**”，这是**HTTP协议**支持的方法之一，**HTTP协议**支持的方法都已经规定好了，有：**GET**、**POST**、**HEAD**、**OPTIONS**、**PUT**、**DELETE**、**TRACE**、**CONNECT**。

**GET** 的意思就是获得，我们访问这个网站就需要获得这个网站（准确的说是获得这个网站的代码），需要将这个网站的页面从服务器加载到我们电脑里面的浏览器上。

至于其他的方法都有什么作用，我们在《HTTP详解2【进阶】》中会有对应的说明。

第二个部分（第一行的第一个空格到第二个空格之间）是**URI**，这里是一个**请求URI**，表示需要请求的网站在服务器上的目录和文件。这里大家可能会有疑问，为什么在浏览器输入的时候还需要**域名**，到这里就只需要目录和文件名了呢？这是因为**HTTP协议**属于顶层协议，不负责具体的收发消息，只负责规定消息格式，所以这里并不需要**域名**来告知消息要发送到哪里。

什么叫做顶层协议呢？我们可以按照我们能够接触的深度来划分层级，可以想像竖着垒的一摞书，越在上面的越好拿，越在下面越不好拿，同理，我们容易接触到协议它的层级就越高，我们越难接触到的层级就越低。**HTTP协议**是解析**URL**之后遇到的第一个协议，所以我这里称之为顶层协议，下面还会有**TCP协议**、**IP协议**等等，而控制具体的收发消息的则是下面的这些协议，这就是之后的内容了。

第三个部分（第一行第二个空格后面）是**HTTP协议**的版本，现在对应的应该是有三个版本：**HTTP/1.0**、**HTTP/1.1**、**HTTP/2.0**（其实还有**HTTP/0.9**，这个太古老了，同样的还有**HTTP/3.0**，这个太新了）。我们在这里主要讲解的就是现在应用非常广泛的**HTTP/1.1**，顺带着讲讲**HTTP/1.0**。

第四个部分我们将第一行以下，下面空行以上的部分合并叫做**首部字段**，这部分是每一行代表一个字段，每个字段分为**字段名**和**字段值**，由冒号（**:**）分隔。**字段名**有一部分是在**HTTP协议**中规定好的，对于规定好的**字段名**，也有对应规定好的**字段值**，不同的**字段名**可用的**字段值**也不同。**字段名**的另一部分是自定义的，也就是由发送者和接收者约定好，自己添加进去的。例子中的**Accept**、**Accept-Encoding**、**Accept-Language**、**Connection**、**Host**、**User-Agent**都是**HTTP协议**中规定好的**字段名**，对应的在它们后面的冒号后面的就是它们的**字段值**了。

其他所有规定好的**字段名**我们将在《HTTP详解2【进阶】》中说明。

第五个部分（空行以下部分）是**消息体**，存放一些具体的消息内容，因为我们请求的是加载并显示网页，用的是**GET**，所以用不到这第五部分，这里我们就不再说明了。（其他类型的**HTTP请求**有可能会用到这一部分，《HTTP详解2【进阶】》中见。）

其实，整个**HTTP请求**的消息是以中间那行空行来区分的，空行上面部分叫做**头部（或首部，Header）**，空行下面部分叫做**主体（Body）**。

空行一般由 `CR+LF` 组成。`CR` 是回车符，16进制的 `0x0d`。`LF` 是换行符，16进制的 `0x0a`。

为了方便之后的理解，这里也说明一下，发送出去的消息（**请求报文**）是上面的格式，但是接收方（也就是服务器）返回的消息回执的格式就有一点区别了，下面是消息回执的格式：

消息回执我们一般称为“**响应（Response）**”，对应的内容称为**响应报文**。

```
1 <HTTP版本><空格><状态码><空格><响应短语>
2 <字段名>:<字段值>
3 <字段名>:<字段值>
4 <字段名>:<字段值>
5 .....
6 <字段名>:<字段值>
7 <空行>
8 <消息体>
```

带入具体内容为：

```
1 HTTP/1.1 200 OK
2 Accept: /*/*
3 Accept-Language: zh
4 Content-Type: text/html
5 Connection: keep-alive
6 Host: www.fangqibiancheng.com
7
8 <html>
9 <head>
10 ...
11 </head>
12 <body>
13 ...
14 </body>
15 </html>
```

这里我们就可以看到和上面发送消息（**请求报文**）的不同了，主要体现在第一行和最下面的**消息体**部分。

第一行第一部分（第一行第一个空格前）变成了HTTP版本，是发送消息（**请求报文**）里面第一行的最后一个部分，并且同一个发送消息（**请求报文**）和消息回执（**响应报文**）的HTTP版本保持一致。

第一行第二部分（第一行第一个空格和第二个空格之间）是**状态码**，这代表你刚刚发送的这条消息对于服务器来说是不是有效的，如果是有效的一般会返回的状态码为 `200`。如果你想要请求的网页不存在，这里就是你非常熟悉的 `404`。至于其他的常用的状态码，我们在《HTTP详解2【进阶】》中会列举出来。

第一行第三部分（第二个空格后面部分）是**响应短语**，这个其实就是对前面的**状态码**的一个文字说明。`200`一般对应的就是 `OK`。

最后的不同在于最底下的**消息体**，前面发送的 `GET` 请求中我们并没有**消息体**，但是在消息回执（**响应报文**）中我们能够看到有**消息体**返回，**消息体**这一部分保存了你想要看到的页面的具体的代码（HTML/CSS/JavaScript.....）。

以上就是关于消息的发送和接收的格式了。从上面复杂的**消息头部**我们可以看出，**头部**是HTTP请求非常重要的部分，它会提供**消息体**的大小，使用的语言，认证信息等等。

在下面我们将会进入到更低层的学习，首先要学习的就是如何查找服务器的在互联网中的地址。