



本节主要讲解http协议相关的内容，作为后期讲述的一个预备知识。深入了解Http协议，对你今后的JavaSE，JavaEE学习都大有裨益，本节难度不是很大，却很重要。



早期的Http协议是1.0版本的，基于TCP协议。Http协议是一种基于请求、响应模式的无状态的文件传输协议。什么叫做基于请求、响应模式呢？举一个例子，比如我访问 www.xiaotublog.com，就在浏览器输入这一个字符串，然后敲一下回车。



网站首页：



刷了一下，博客首页就被打开了，而且地址栏自动给我加上了http//，虽然浏览器上可能看不到，但实际上肯定是有。你有意识到刚才发生了什么吗？没错，我通过浏览器这一个客户端，向某个远程服务器发送了一个http请求，因为我是通过域名来访问的。所以，为了解析这个域名，我刚才敲的这一下回车肯定是先走到了某一个DNS服务器，然后这个DNS服务器将域名解析后拿到一个IP地址，最终才传送到对应的服务器上。博客网站的服务器只有IP地址，没有域名的说法。实际上，在这个服务器上，也就是跑了一个Tomcat，Tomcat里面有一个博客的web项目，服务器原则上一直在开机状态，tomcat也一直在运行着。它处于一种等待状态，一直在等着有外部的Http请求过来。比如，我刚才就敲了一下回车，一个http请求就被发送过去了，tomcat软件接收到这个请求，并且解析这个请求。经过解析，就把其中的一个页面资源找到，并且通过响应的方式回执给我的客户端 - 浏览器。

让我们右键查看源代码，就能看到服务器给我们返回的其实就是html文本。

[illegible]

利用浏览器的调试工具，我们可以看到更多关于本次http请求的信息。

▼ **General**

Request URL: http://www.xiaotublog.com/index.html
Request Method: GET
Status Code: 200 OK
Remote Address: 122.114.160.91:80

▼ **Response Headers** [view source](#)

Content-Language: zh-CN
Content-Type: text/html; charset=UTF-8
Date: Wed, 25 Jan 2017 04:07:13 GMT
Server: Apache-Coyote/1.1
Transfer-Encoding: chunked

再来说说无状态，什么叫无状态呢？就是说，我刚才访问了一个网页，如果我马上再次刷新一下，是不是又对服务器发送了一次Http请求啊？是的，可是，对服务器而言，它并不知道两次的请求是否是同一个人，或者说同一台电脑发出来的，他不知道。这个就叫做无状态。

在早期的时候，Http请求还是1.0版本，他会给每一次的请求都建立一个连接，当请求结束，连接也就是自动关闭了。如果用户所请求的资源仅仅是一些纯文本，包括嵌入在页面中的css，和js代码，那么问题不是很大。可是在现在，网页内容越做越丰富，大量的外部css被引用，比如bootstrap.min.css。大量的外部js被引入，比如jQuery.min.css，还有就是有一个站，往往会引入大量的图片。比如，我现在有一个网站的首页是这样子的。

```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<meta name="Generator" content="EditPlus@">
<meta name="Author" content="">
<meta name="Keywords" content="">
<meta name="Description" content="">
<title>Document</title>
<link type="text/css" rel="stylesheet" href="1.css" />
<link type="text/css" rel="stylesheet" href="2.css" />
<link type="text/css" rel="stylesheet" href="3.css" />
</head>
<body>

</body>
</html>
```

在html.0版本的时候，它是这样的，首先，和服务器建立连接，请求html文本，然后服务器把上面的这段代码发过来，本地的浏览器接收，OK，连接关闭。接下来，有趣的事情发生了，浏览器一行一行解析这些代码，当解析到这一句的时候，再次向服务器发送一条请求，专门请求1.css这个文件。然后，服务器把1.css的纯文本发送给浏览器，浏览器缓存下来，接着，第二次连接关闭。同理，对于2.css和3.css，都会再分别建立一个连接，获取资源，然后又把连接关闭。不但是外部的css文件会这样，就连js文件，还有哪怕是张图片，img标签等等，都会如此。

我们知道，随着网站的内容越来越丰富，图片动辄几十张，显然这种方式会持续不断的浪费性能，不可取。所以，这才有了http2.0。http2.0最重要的一点，就是新增了一个持续连接的功能，就如上面那个HTML文件，正常情况下，它只会连接一次，在本次连接的过程中，会负责所有文件的请求和返回操作，当一段时间用户没有任何操作的话，这个连接才会自动断开。这就是所谓的http2.0