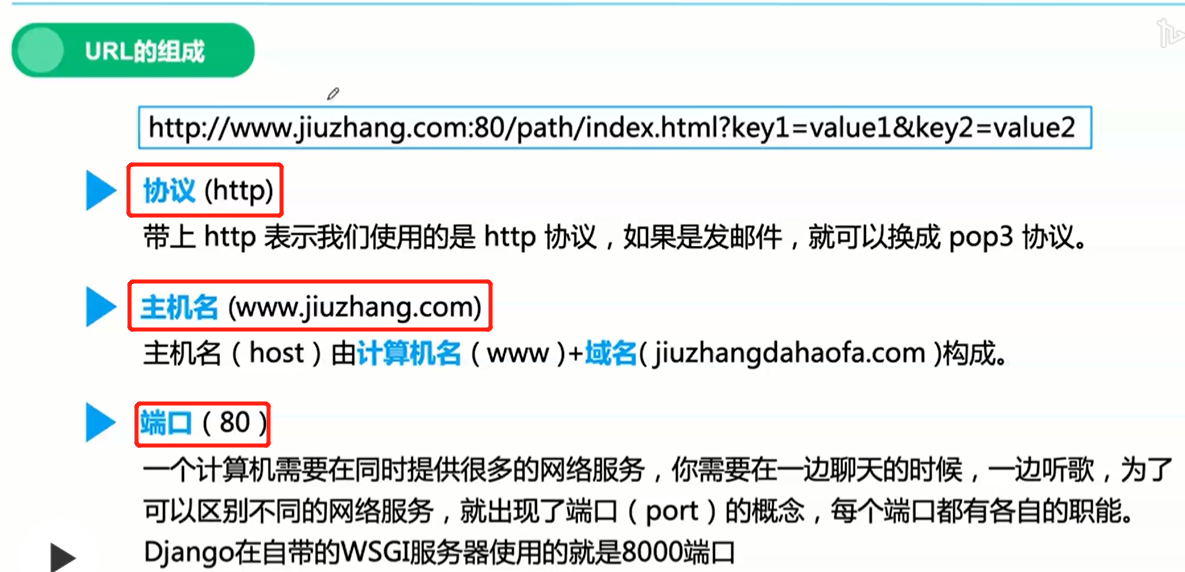
HTTP请求：客户端&服务端

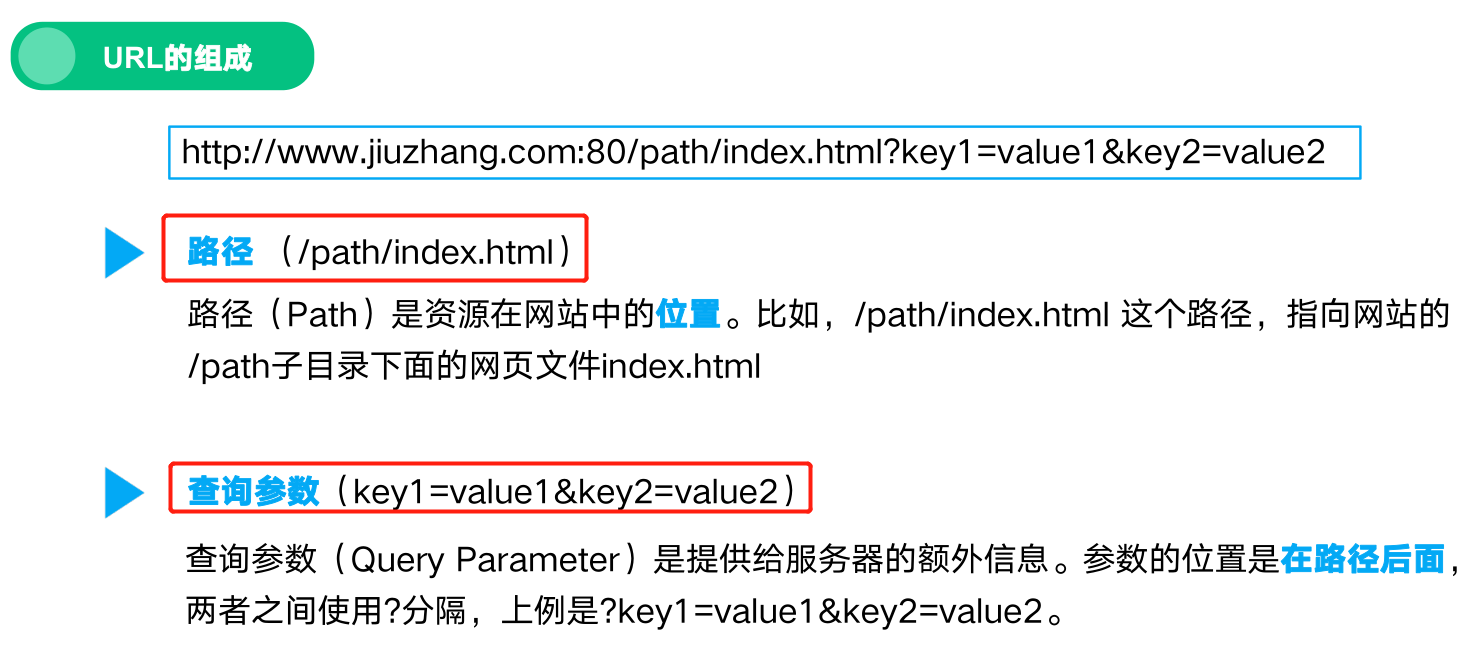
手机充当客户端，推特充当服务器的角色。

HTTP协议：超文本传输协议，本质上就是一种数据格式，用来沟通客户端和服务器。

URL：

例：下面是一个HTTP的GET请求



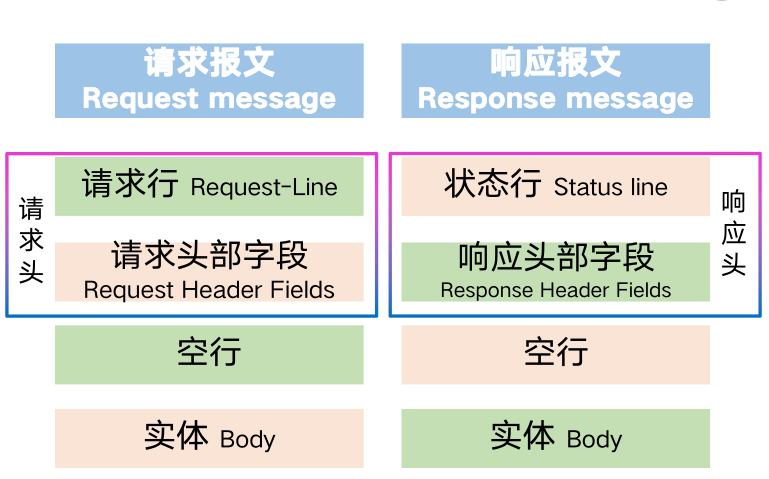


其中“路径”的概念通过直接访问服务器的文件，这是一种比较老的方式，现在一般不用了，而是通过router(路由)的方式来访问，不再指定文件路径，而是根据URL链接通过router来判断调用哪个函数来处理并返回Response，这样路径的命名更自由。

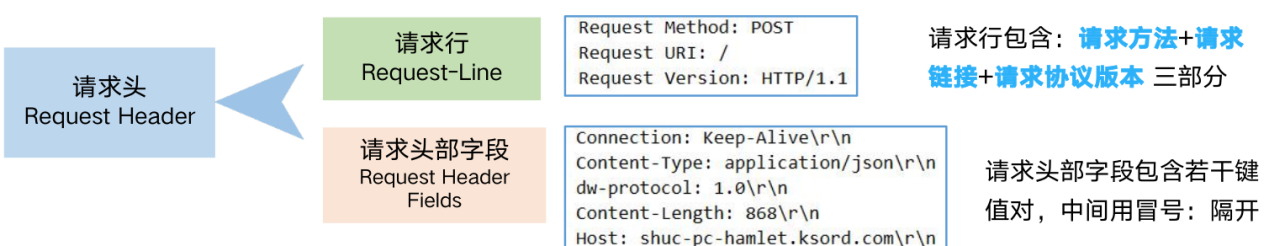
HTTP中GET和POST的区别。

HTTP报文：

HTTP请求数据被封装成请求报文(Request Message)，返回数据被封装成返回报文(Reponse Message)。

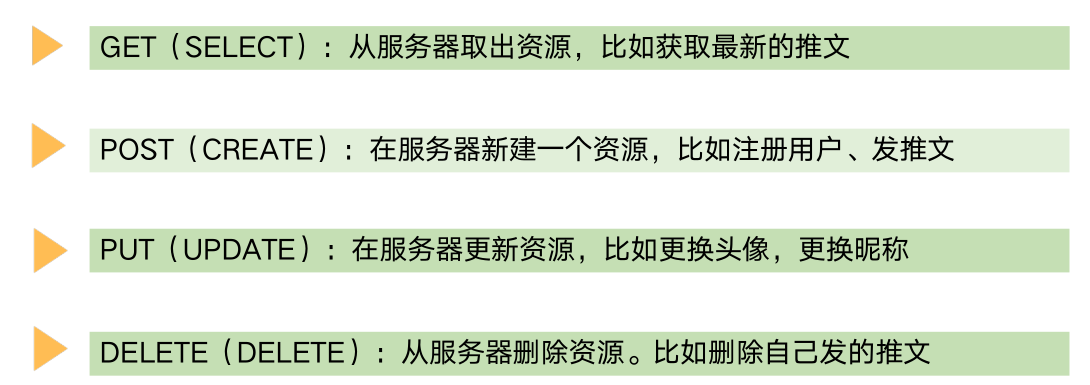


请求报文中的Request Header：



请求行：

HTTP/1.1(HTTP协议中最常用的一种)总共定义了8种方法，其中最常用的是GET和POST，其次会用到PUT和DELETE。



参考：<https://blog.csdn.net/eson_15/article/details/88083280>



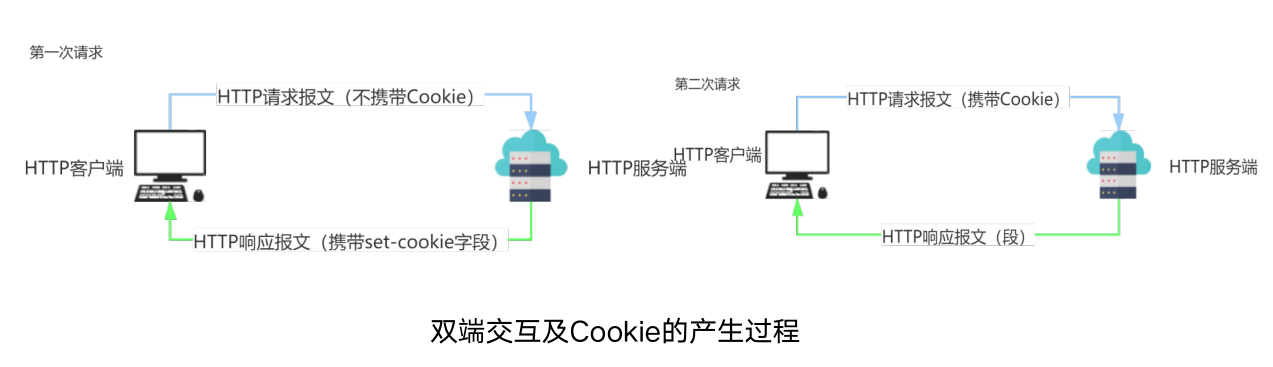
GET和POST本质上都是TCP链接，只是HTTP的规定和浏览器/服务器的限制，才会导致差别。

请求头部字段：

比较重要的几个字段：

Cookie：服务器给客户端的令牌或者通行证，为了辨别用户进行会话跟踪而存储在用户本地的数据，主要功能是维持当前访问会话。当登陆了网站并获取到Cookie后，浏览器的每次访问都会带上Cookie。

Cookie本质是一个哈希表，存储和用户身份有关的信息(也可以存放其它信息，例如偏好等，但不推荐)，。



Third Party Cookie(第三方Cookie)

Cookie有三种主要用途：会话管理，个性化和跟踪。使用Cookie进行个性化时，主要与用户偏好有关，如语言，主题，颜色等。

Cookie第三方应用的小例子：

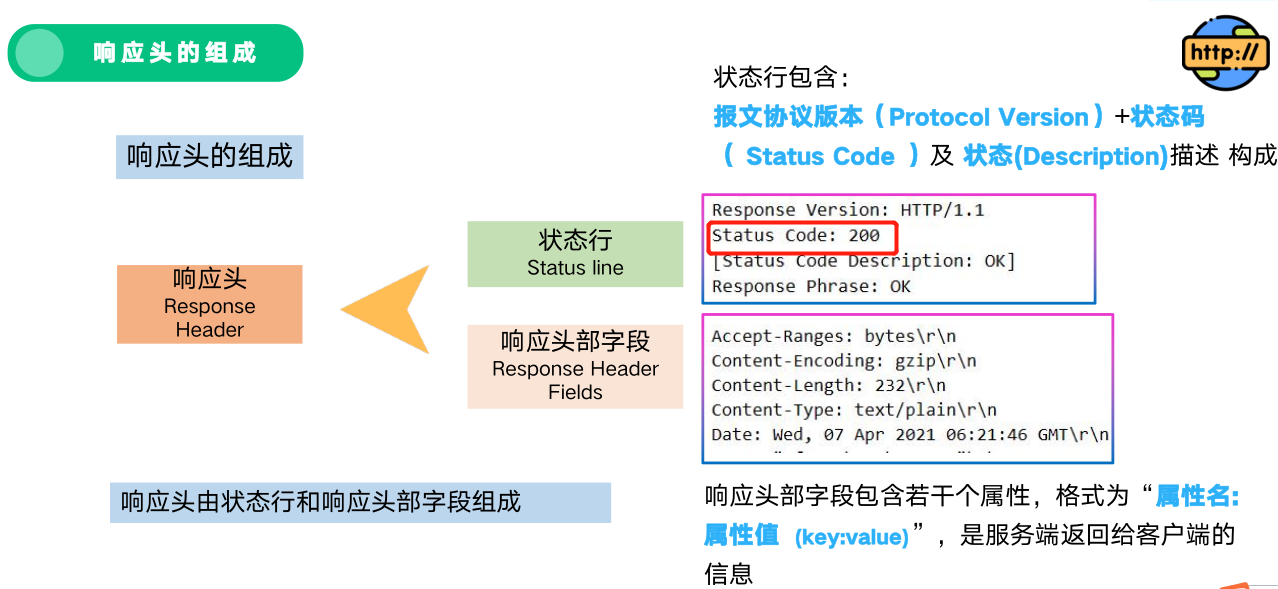
生活中如果你登陆了一家旅游网站，查询了机票和酒店信息后，再去登陆一家电商网站，如果这两个网站有相同的背景(同一家大公司旗下的，或一家大公司持有这两个网站的股票，例如Google)，你会发现在电商网站的广告区域会向你推荐便宜的酒店或机票等，这种情况就是通过Third Party Cookie实现的。

实现的原理：网站A和网站B必须都嵌入了一段服务或代码，它能够在用户访问网站时，同时去想Google的某个网站访问去请求一个1px X 1px的简单图像实现一个HTTP请求，这样Google的网站就会返回一个Cookie，此Cookie会存储在用户的计算机上(这个就是Third Party Cookie)，并且网站A会将用户的行为上传到Google的服务器上，当用户登陆网站B时，网站B也会向Google网站建立一个1px X 1px的简单HTTP请求，并且由于用户本地存储了之前Google网站返回的Cookie，网站B就可以根据Cookie从Google的服务器上获取之前在网站A上浏览的信息，从而进行广告推送。

Refer：标识请求是从哪个页面发过来的，服务器可以拿这些信息并做相应的处理。如来源统计，防盗链处理，反爬虫等。

User-Agent：帮助服务器识别客户使用的操作系统以及版本信，浏览器及版本等。服务器可以根据不同的User-Agent来进行不同的反应，例如桌面上的浏览器发起的请求和手机浏览器发起的请求，可以进行不同的处理并返回不同的Response。

HTTP响应报文：



HTTP状态码：是一个3位数字



常用的HTTP状态码：

200：请求成功，一般用于GET和POST请求。

201：已创建。成功请求并创建了新的资源，Twitter项目中会用到。

301：请求的URL已被永久移动到新位置，即永久重定向，经常出现在URL末尾少了一个反斜杠/的情况

302：临时重定向，原URL还可以使用。

304：未修改。所请求的资源未修改，服务器返回此状态码时，不会返回任何资源。客户端通常会缓存访问过的资源，通过提供一个头信息指出客户端希望只返回在指定日期之后修改的资源

400：客户端请求语法错误，服务器无法理解。例如Twitter登陆时少了user\_id等

401：请求用户的身份认证。

403：禁止访问，用户没有权限。

404：资源不存在，服务器无法根据客户端的请求找到资源。

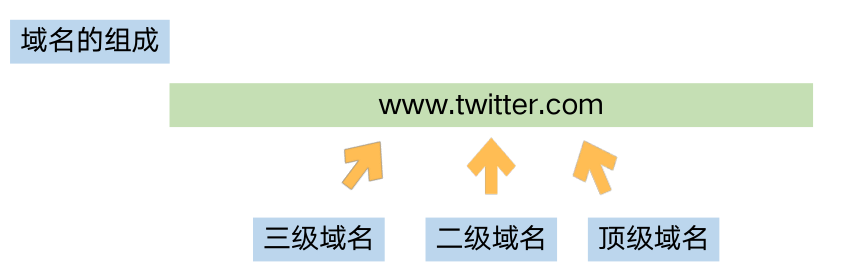
405：请求方法错误，如使用GET方法，但服务器需要POST方法，或使用了POST方法，但需要的是GET方法。

500：服务器内部错误，无法完成请求

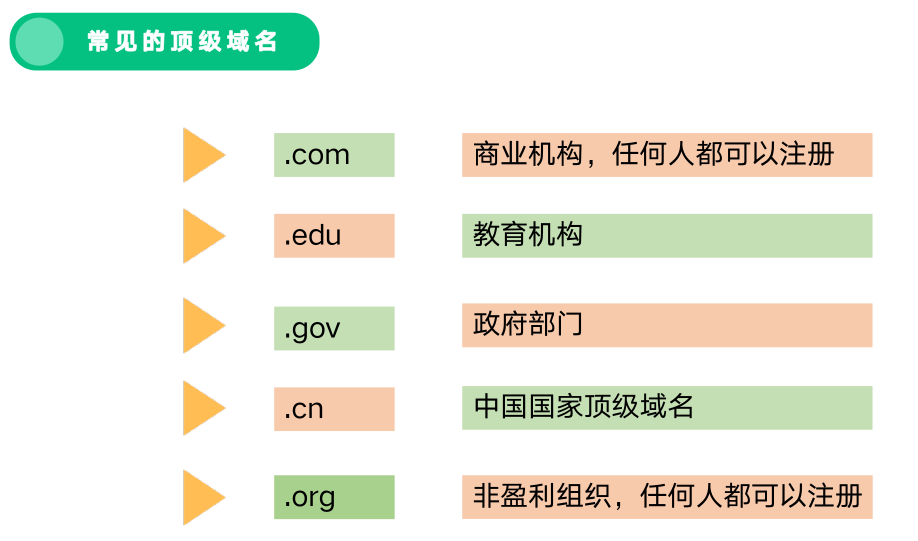
502：作为网关或者代理工作的服务器尝试执行请求时，从远程服务器收到一个无效的请求。

504：充当网关或代理的服务器，未及时从远端服务器获取请求

域名系统DNS：



常见的顶级域名：

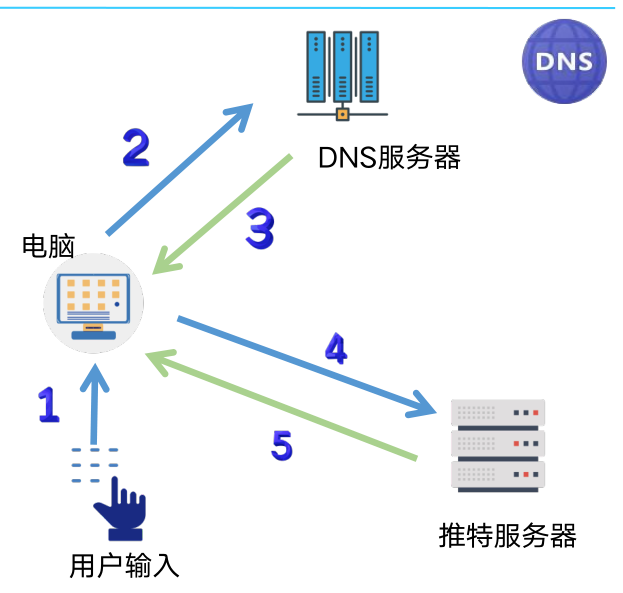


DNS就类似一个电话簿，如果需要给一个人打电话，不仅需要指定名字，还需要指定他对应的电话号码，这里人名就相当于域名，电话号码就是IP地址，DNS就是电话簿。

通过ping + 域名，可以查看它对应的IP地址。

浏览器访问网站的过程：域名对应的IP地址不容易记忆，所以才有了域名。

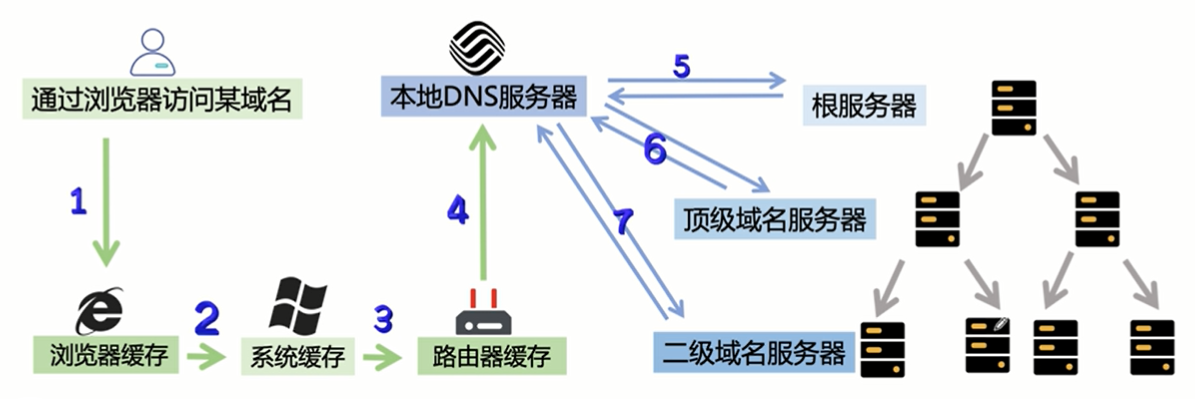
1. 输入[www.twitter.com](http://www.twitter.com)并回车
2. 浏览器发起域名解析(DNS)
3. 从DNS服务器获取[www.twitter.com对应的ip](http://www.twitter.com对应的ip)地址(108.160.170.33)
4. 浏览器向108.160.170.33请求网页
5. 推特服务器响应请求



DNS解析的全过程：DNS解析时并不是上来就访问DNS最终根服务器的(全球只有13个)，而是先通过一层一层的缓存来获取域名对应的IP地址的。

通过浏览器访问某域名：

1. 浏览器缓存中查找，如果未找到，进行第2步
2. 系统缓存中查找，如果未找到，进行第3步
3. 路由器缓存(有可能会存在多级路由器缓存)，如果未找到，进行第4步
4. 本地DNS服务器，如果未找到，进行第5步
5. 去根服务器查找，根服务器下面有很多计算机集群，最终查找到需要的IP地址



本地的电信提供商，会在多个地方都布置本地DNS服务器，PC上一般都是默认的DNS服务器，也可以手动指定第三方的DNS，如国内的114.114.114.114，Google的8.8.8.8。