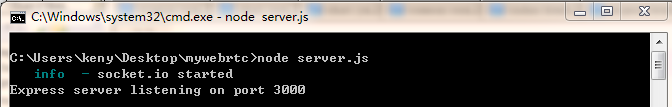
[**浏览器输入URL与Web服务器交互过程**](http://www.cnblogs.com/gnss523-webrtc/articles/4834900.html)

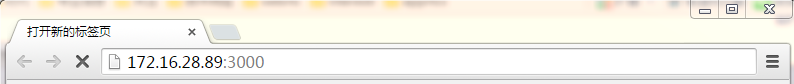
前期用node.js简单搭建了Web服务器，包含简单的注册登入功能。但对于其内部逻辑仍旧一知半解，今天把这个过程好好理一理。

客户端抓包说明：        客户端IP：192.168.1.7，内网服务器IP：172.16.28.89。

首先在本地运行服务器端程序，用到了express框架。



然后在谷歌浏览器端输入URL(Uniform Resource Locator,统一资源定位符)地址,敲下回车键。URL（如http://172.16.28.89:3000/signin）包括协议（http）、网络地址(172.16.28.89:3000)、资源路径(/singin)三个部分。浏览器中输入URL默认缺省http。



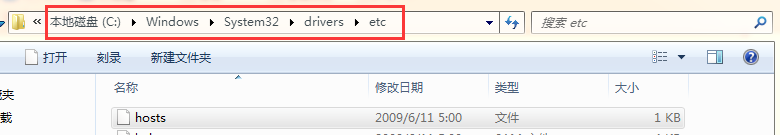
其交互过程如下：

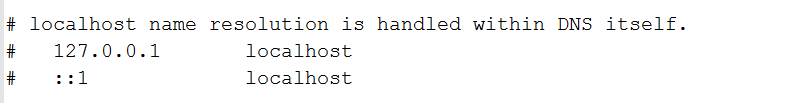
一、查找到URL所在的服务器地址。

1、如果URL的地址是一个域名，不是IP地址，则通过DNS(Domain Name System，域名系统)将域名解析成IP地址。步骤如下：

1）**浏览器缓存。**如果存在，则解析过程结束；

2）**系统缓存。**如果浏览器未缓存，浏览器会检查操作系统缓存中是否有这个这个域名对应的ip地址。windows系统中缓存路径及文件部分内容如下：

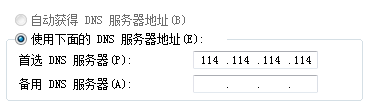




3）**路由器缓存。**若本机中没有解析到，则将请求发送给路由器。wireshark抓包截图如下：

https://images2015.cnblogs.com/blog/815335/201509/815335-20150924111355365-885590686.png

4）**DNS提供商缓存。**若仍未解析到，则检查TCP/IP中设置的DNS服务器，将URL发送给DNS服务器，最终返回域名的Ip地址。windows下DNS服务器设置界面如下:



我是直接输入的服务器ip地址(172.16.28.89:3000)，故不需DNS解析。

2、如果URL中不包括端口号，则使用协议的默认端口号。如http协议的默认端口号为80。我在服务器端设置的监听端口为3000，故浏览器访问时输入对应端口。

二、TCP三次握手，建立连接。

http应用层协议是建立在TCP传输层协议之上的。在浏览器发送http请求之前，会先通过三次握手建立TCP连接，提供可靠传输。

https://images2015.cnblogs.com/blog/815335/201509/815335-20150924115538819-480405745.png

三、浏览器根据http规范，产生请求数据包。wireshark抓包如下:



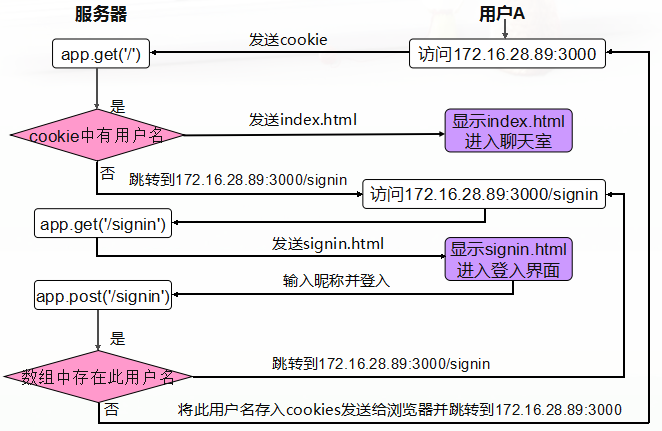
http定义了与服务器交互的不同方法，最基本的方法有4种，分别是GET，POST，PUT，DELETE。

其中最重要的两种方法：**GET是向服务器发索取数据的一种请求，而POST是向服务器提交数据的一种请求。**

抓包显示，第一次访问服务器为**GET请求**，本地并未保存任何cookies信息，故http请求头部中并未包含任何cookie信息。

关于http协议的具体细节后续再深入研究。

四、服务器与客户端交互



1、服务器端响应： app.get('/'function (req, res) {}， 检查请求消息中是否含有cookies信息，没有则让用户重定向到/signin；

2、客户端再次通过URL(**http://172.16.28.89:3000/signin**)像服务器发送GET请求；

3、服务器端响应：app.get('/signin', function (req, res) {}，向客户端发送signin.html文档；

4、文档中包含CSS式样、图片资源，故客户端再次像服务器发起请求；

5、服务器返回请求的资源；

6、客户端显示注册登入页面，输入用户名后点击登入。通过**POST**方法像服务器提交数据请求；

7、服务器端响应:app.post('/signin', function (req, res) {}，检查POST请求中的用户名，若已经存在即该用户名已注册，则让用户重定向到/signin，再次输入用户名注册登入；若用户名不存在，则服务器通过Set-Cookie将用户名存入Cookies并发送给浏览器，同时重定向到172.16.28.89:3000。



8、客户端收到消息后，将Cookie存入本地，此后客户端通过URL(**http://172.16.28.89:3000**)向服务器发送请求时，请求中都会自动包含这个Cookie。

 9、重复第一步，服务器判断请求消息中是否包含Cookies信息，有则返回index.html文档。至此结束。

如有问题，欢迎交流指正。