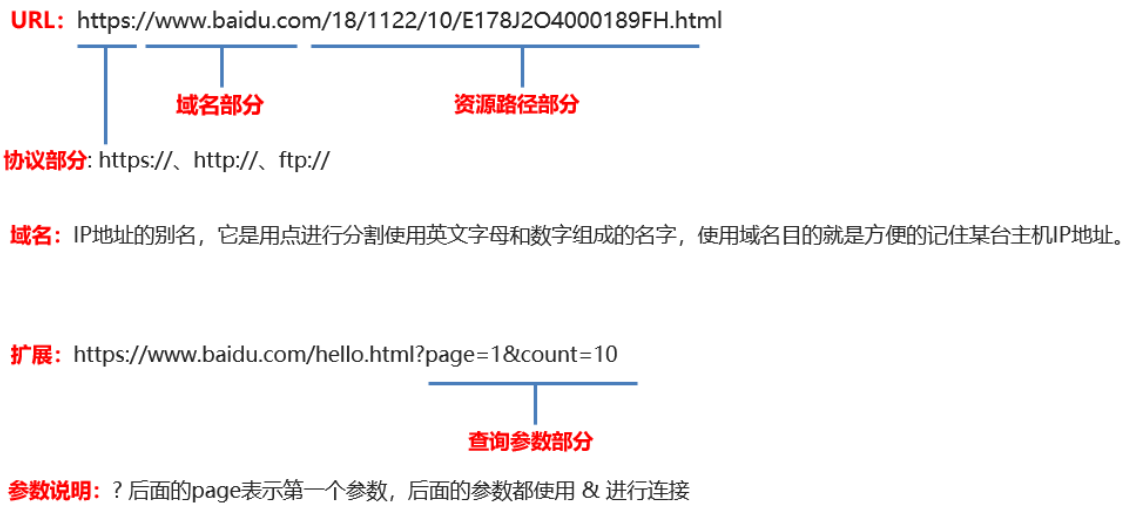
网址

1.网址的概念

网址又称为URL，URL的英文全拼是(Uniform Resoure Locator),表达的意思是统一资源定位符，通俗理解就是网络资源地址。

URL地址：https://www.baidu.com/s?wd=python
https://www.baidu.com/s?wd=hello

2.URL的构成



域名：IP地址的别名，它是用点进行分割使用英文字母和数字组成的名字，使用域名目的就是方便的记住某台主机IP地址。

参数说明：？后面的page表示第一个参数，后面的参数都使用 & 进行连接

知识要点

http：
https：http+ssl
ssl：具有服务器身份验证和数据输出的加密功能

网址就是网络资源的地址，又称为URL，通过URL能够找到对应的资源数据。
URL组成部分
    协议部分
    域名部分
    资源路径部分
    查询参数部分 [可选]

HTTP协议的介绍

思考：

通过HTTP协议来规定浏览器和web服务器之间通讯的数据的格式

1.HTTP协议的概念及作用

HTTP协议的全称是(HyperText Transfer Protocol)，翻译过来就是超文本传输协议。
超文本是指在文本数据的基础上还包括非文本数据，非文本数据有图片、音乐、视频等，而这些非文本数据会使用链接的方式进行加载显示，通俗来说超文本就是带有链接的文本数据也就是我们常说的网页数据。

HTTP协议的制作者是蒂姆·伯纳斯-李，1991年设计出来的，HTTP协议设计之前目的是传输网页数据的，现在允许传
输任意类型的数据。

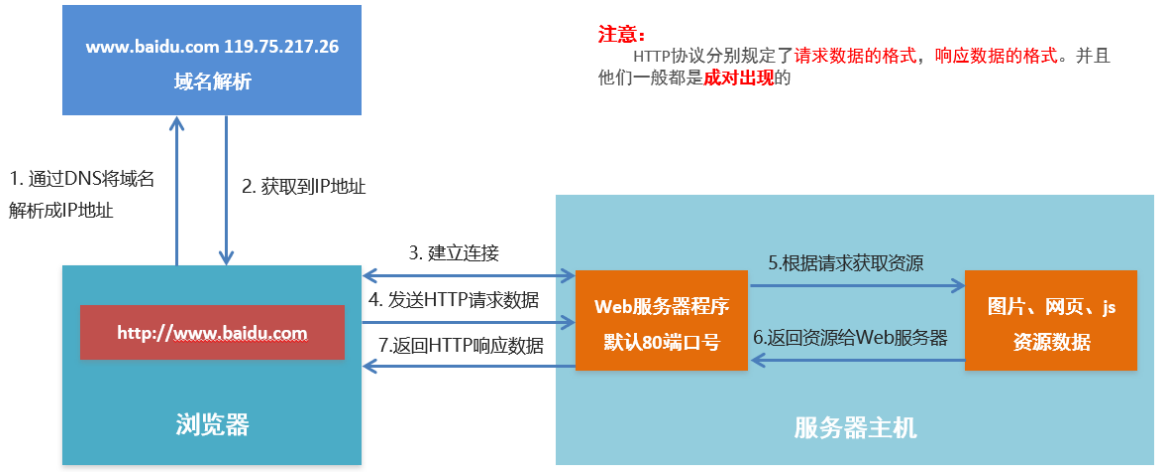
传输HTTP协议格式的数据是基于TCP传输协议的，发送数据之前需要先建立连接。
TCP传输协议是用来保证网络中传输的数据的安全性的，HTTP协议是用来规定这些数据的具体格式的。

注意:

HTTP协议规定的数据格式是浏览器和Web服务器通信数据的格式，也就是说浏览器和Web服务器通信需要使用HTTP协议。

2.浏览器访问Web服务器的过程

> 管理数据的仓库



知识要点

HTTP协议是一个超文本传输协议，它是一个基于TCP传输协议传输数据的，它是浏览器和web服务器传输数据的一个协议(HTTP)。

TCP传输协议是用来保证网络中传输的数据的安全性的，HTTP协议是用来规定这些数据的具体格式的。

HTTP协议分别规定了请求数据的格式，响应数据的格式。并且他们一般都是成对出现的

HTTP请求报文

1.HTTP最常见的请求报文有两种:

GET方式的请求报文：所有的参数都会在url地址上显示
POST方式的请求报文：参数不会在url地址上显示

浏览器--服务器发请求（携带一些信息数据）

https://www.baidu.com/s?wd=python
注册、登录
post方式

说明：

GET：获取Web服务器数据
POST：向Web服务器提交数据（注册，姓名，电话号码，提交服务器，保存数据库）

2.HTTP GET请求报文分析

GET请求方式

3.HTTP GET请求报文分析

```
---- 请求行 ----
GET /a/b/c HTTP/1.1  # GET请求方式 请求资源路径 HTTP协议版本
---- 请求头 -----
Host: www.baidu.com  # 服务器的主机地址和端口号,默认是80
Connection: keep-alive # 和服务端保持长连接
Upgrade-Insecure-Requests: 1 # 让浏览器升级不安全请求，使用https请求
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36  # 用户代理，也就是客户端的名称
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8 # 可接受的数据类型
Accept-Encoding: gzip, deflate # 可接受的压缩格式
Accept-Language: zh-CN,zh;q=0.9 #可接受的语言
Cookie: pgv_pvi=1246921728; # 登录用户的身份标识
---- 空行 ----
```

```
GET / HTTP/1.1\r\n
Host: www.baidu.com\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10_12_4) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/69.0.3497.100
Safari/537.36\r\n
Accept:
text/html,application/xhtml+xml,application
/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\
r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: zh-CN,zh;q=0.9\r\n
Cookie: pgv_pvi=1246921728; \r\n
\r\n  (请求头信息后面还有一个单独的'\r\n'不能省
略)
```

说明:

每项数据之间使用:**\r\n**

4.HTTP POST请求报文分析

POST方式

5.HTTP POST请求报文分析

```
---- 请求行 ----
POST /xmweb?
host=www.douban.com&_t=1542884567319
HTTP/1.1 # POST请求方式 请求资源路径 HTTP协议版
本
---- 请求头 ----
Host: www.douban.com # 服务器的主机地址和端口
号,默认是80
Connection: keep-alive # 和服务端保持长连接
Content-Type: application/x-www-form-
urlencoded  # 告诉服务端请求的数据类型
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10_12_4) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/69.0.3497.100
Safari/537.36 # 客户端的名称
---- 空行 ----
---- 请求体 ----
username=hello&pass=hello # 请求参数
```

```
POST /xmweb?
host=www.douban.com&_t=1542884567319
HTTP/1.1\r\n
Host: www.douban.com\r\n
Connection: keep-alive\r\n
Content-Type: application/x-www-form-
urlencoded\r\n
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10_12_4) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/69.0.3497.100
Safari/537.36\r\n
\r\n(请求头信息后面还有一个单独的'\r\n'不能省略)
username=hello&pass=hello
```

说明:

每项数据之间使用:**\r\n**

知识要点

一个HTTP请求报文可以由请求行、请求头、空行和请求体4个部分组成。

请求行是由三部分组成：请求方式 请求资源路径 HTTP协议版本

GET方式的请求报文没有请求体，只有请求行、请求头、空行组成。

POST方式的请求报文可以有请求行、请求头、空行、请求体四部分组成。
注意：POST方式可以允许没有请求体，但是这种格式很少见。

GET方式

POST方式

HTTP响应报文

1.HTTP响应报文分析

响应报文

2.HTTP响应报文分析

```
--- 响应行/状态行 ---
HTTP/1.1 200 OK # HTTP协议版本 状态码 状态描述
--- 响应头 ---
Server: Tengine # 服务器名称
Content-Type: text/html; charset=UTF-8 # 内容类型
Connection: keep-alive # 和客户端保持长连接
Date: Fri, 23 Nov 2018 02:01:05 GMT # 服务端的响应时间
--- 空行 ---
--- 响应体 ---
<!DOCTYPE html><html lang="en"> …</html> # 响应给客户端的数据
```

```
HTTP/1.1 200 OK\r\n
Server: Tengine\r\n
Content-Type: text/html; charset=UTF-8\r\n
Connection: keep-alive\r\n
Date: Fri, 23 Nov 2018 02:01:05 GMT\r\n
\r\n(响应头信息后面还有一个单独的'\r\n'不能省略)
<!DOCTYPE html><html lang="en"> …</html>
```

说明:

每项数据之间使用:**\r\n**

3.HTTP状态码介绍

- 跟服务器要数据:
- 200:代表请求是正常的
- 404:url正不正确

是用于表示Web服务器响应状态的3位数字代码

> https://www.runoob.com/http/http-status-codes.html
>
> 403:服务器识别出是一个爬虫程序(考虑反爬手段了)

状态码	说明
200	服务器已成功处理了请求
400	错误的请求，请求地址或者参数有误
404	请求资源在服务器不存在
500	服务器内部源代码出现错误

知识要点

一个HTTP响应报文是由响应行、响应头、空行和响应体4个部分组成。
响应行是由三部分组成：HTTP协议版本 状态码 状态描述，最常见的状态码是200

响应报文

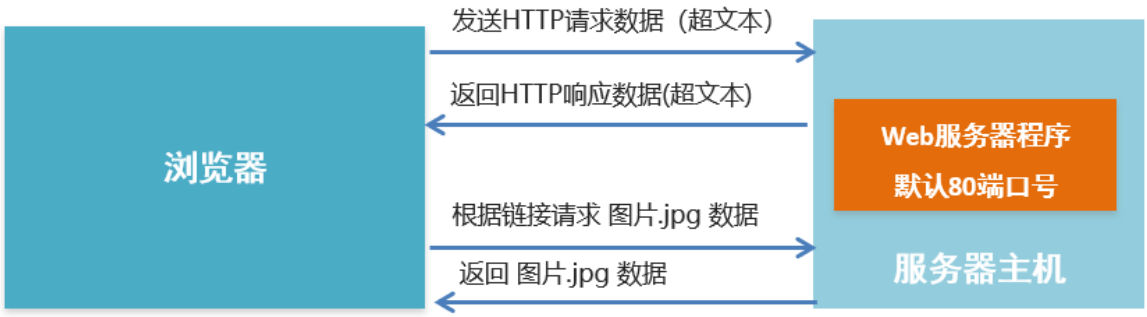
查看HTTP协议的通信过程

1.谷歌浏览器开发者工具的使用

一个页面上的数据是由吧不同url所携带的数据组成起来的

安装Google Chrome浏览器，在Windows和Linux平台按F12调出开发者工具，Mac中选择“视图 ->开发者 ->”开发者工具或者直接使用 alt+command+i 这个快捷键，还有一个多平台通用的操作就是在网页右击选择检查。

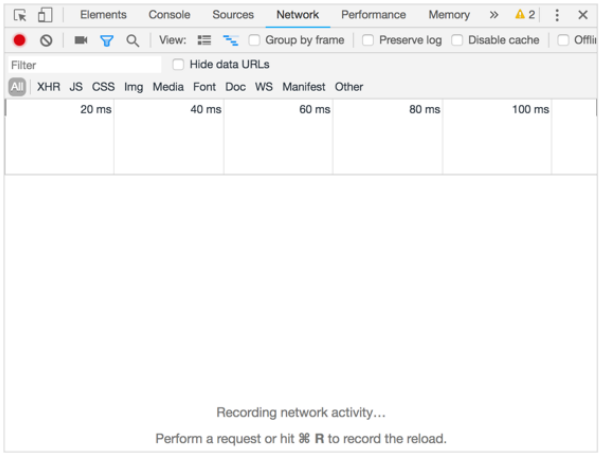
提示：开发者工具还是查看网页布局和JS代码调试的利器。



注意：每一次浏览器和服务器的数据通讯，都是成对出现的即请求和响应，
同时每一次请求和响应都必须符合HTTP协议的格式

2.谷歌浏览器开发者工具的使用

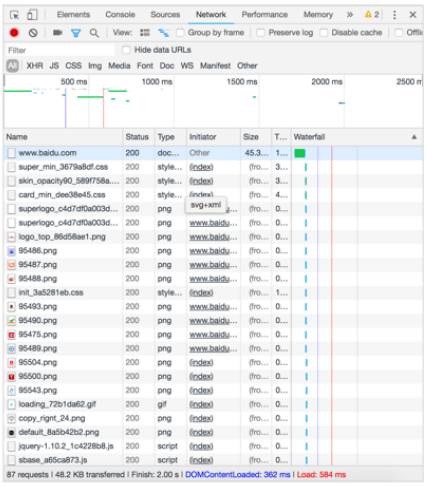
> 重点是在网络：



标签选项说明：

- 元素（Elements）：用于查看或修改HTML标签
- 控制台（Console）：执行JS代码
- 源代码（Sources）：查看静态资源文件，断点调试JS代码
- 网络（Network）：查看http协议的通信过程

3.谷歌浏览器开发者工具的使用



使用说明：

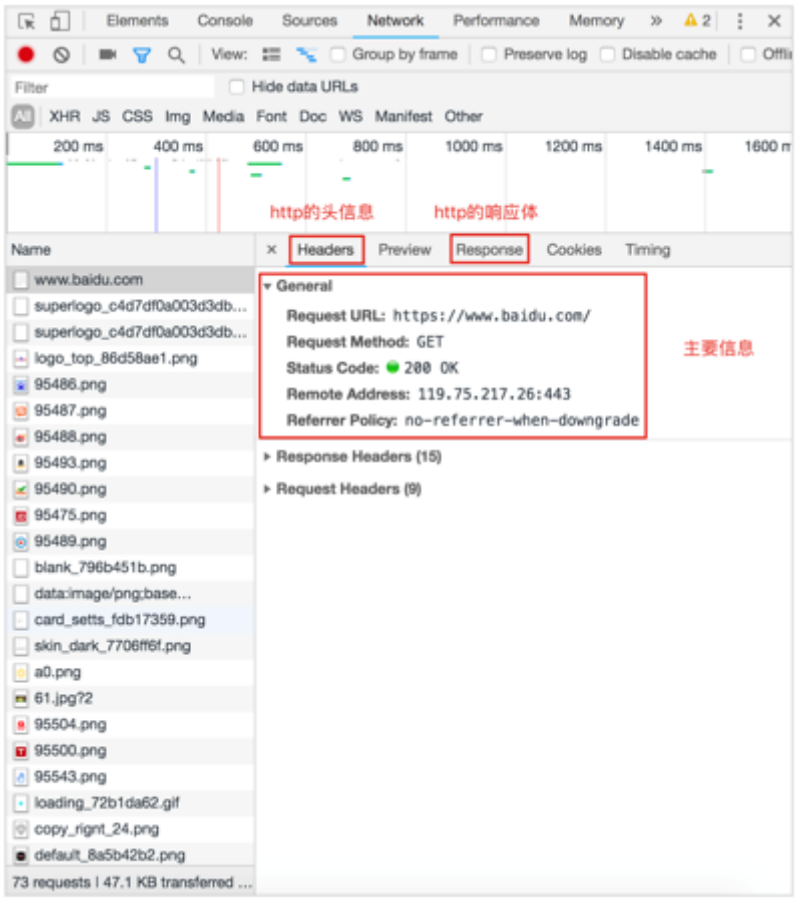
① 点击Network标签选项

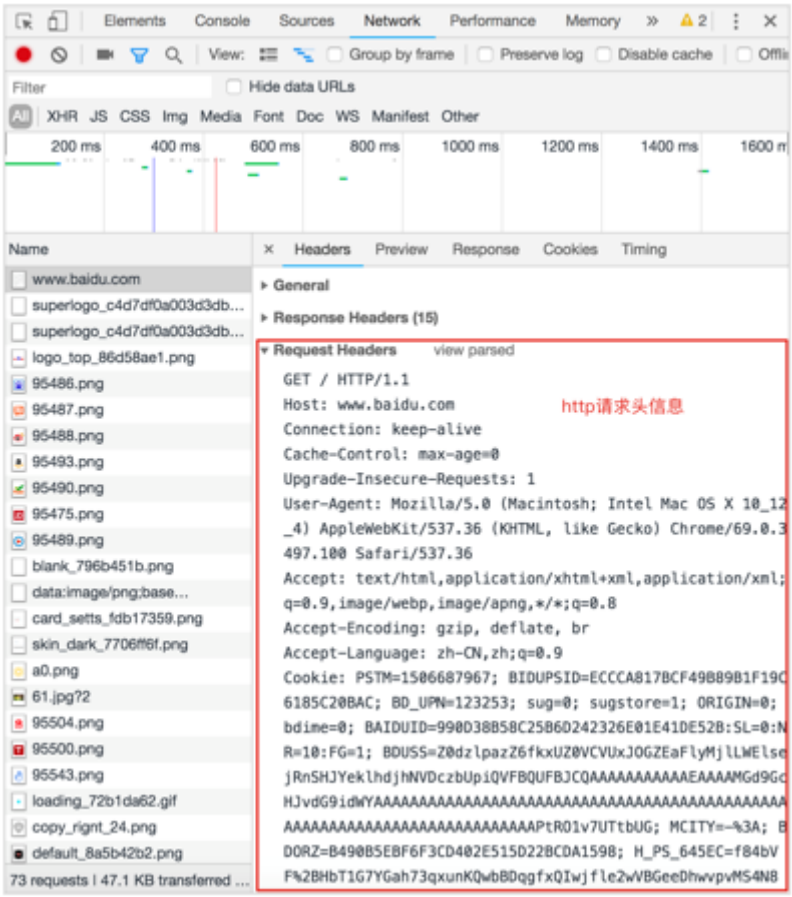
② 在浏览器地址栏输入百度的网址，就能看到请求百度首页的HTTP的通信过程

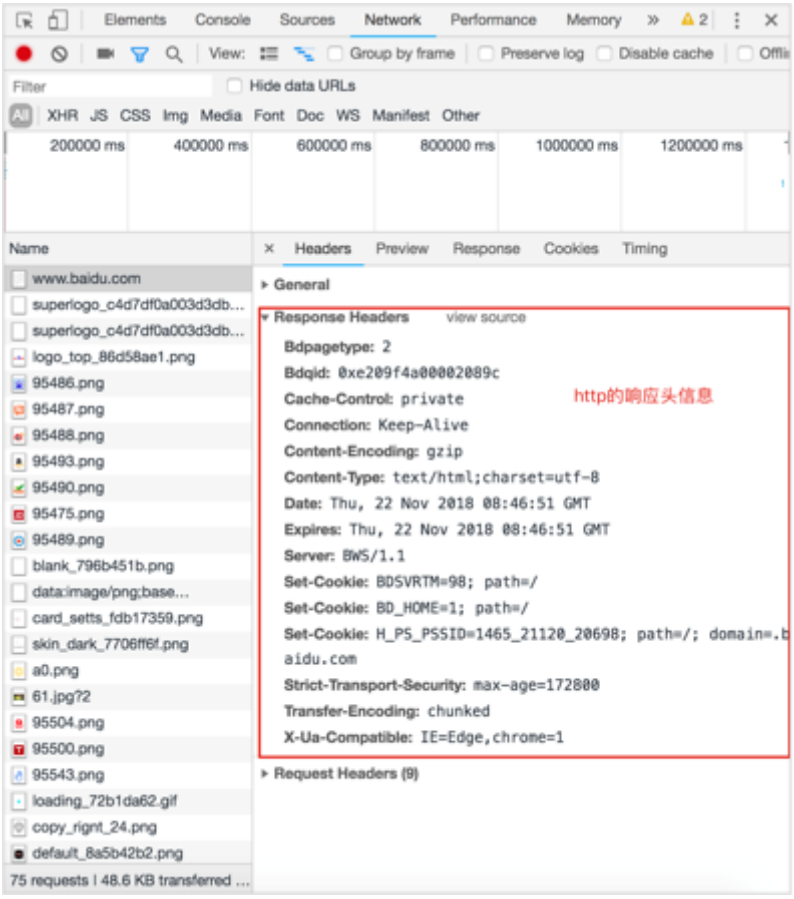
③ 这里的每项记录都是请求+响应的一次过程

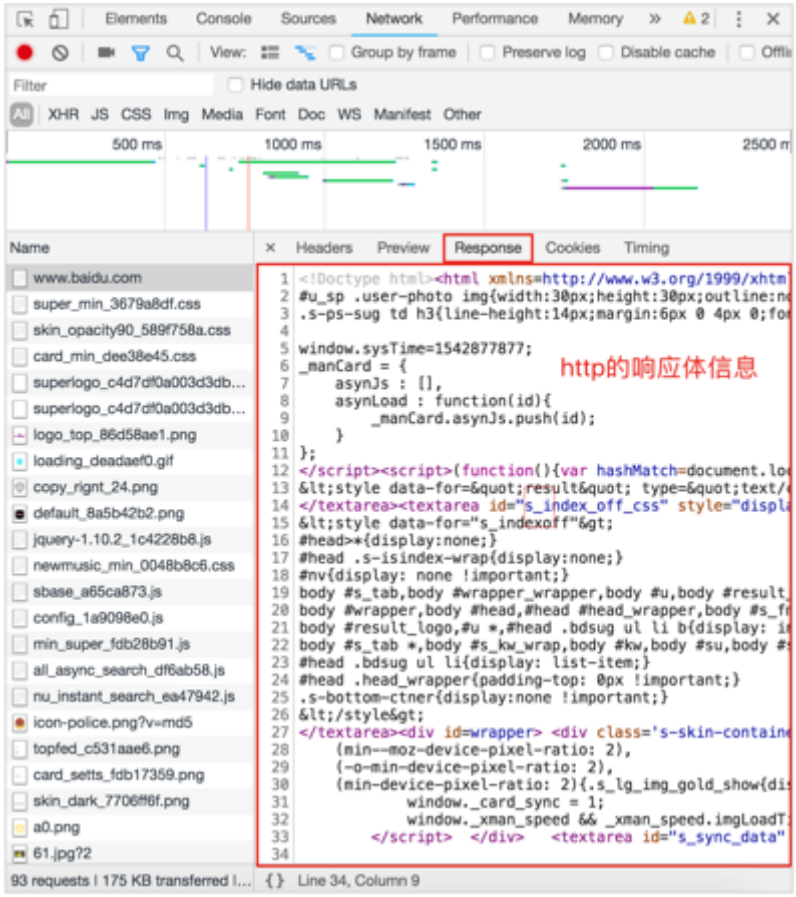
4.查看HTTP协议的通信过程

> 问题：一个网页显示的数据只对应一个url地址吗？
>
> 不是，一个网页的数据由多个url组成，
>
> 静态加载：
>
> 鼠标右击——查看网页源码——呈现出源码内容（当前访问网站url，向这个url发请求，得到的内容）
>
> 搜索快捷键：ctrl+F
>
> 动态加载：优先去xhr里面找


Elements
Console
Sources
Network
Performance
Memory
Filter
Hide data URLs
XHR JS CSS Img Media Font Doc WS Manifest Other
200 ms
400 ms
600 ms
800 ms
1000 ms
1200 ms
1400 ms
http的头信息
http的响应体
Name
Headers
Preview
Response
Cookies
Timing
www.baidu.com
General
Request URL: https://www.baidu.com/
Request Method: GET
Status Code: 200 OK
Remote Address: 119.75.217.26:443
Referrer Policy: no-referrer-when-downgrade
主要信息
Response Headers (15)
Request Headers (9)
73 requests | 47.1 KB transferred


Elements
Console
Sources
Network
Performance
Memory
Filter
Hide data URLs
XHR JS CSS Img Media Font Doc WS Manifest Other
200 ms
400 ms
600 ms
800 ms
1000 ms
1200 ms
1400 ms
Name
Headers
Preview
Response
Cookies
Timing
www.baidu.com
General
Response Headers (15)
Request Headers
view parsed
GET / HTTP/1.1
Host: www.baidu.com
http请求头信息
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12
_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3
497.100 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;
q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Cookie: PSTM=1506687967; BIDUPSID=ECCCA817BCF49B89B1F19C
6185C20BAC; BD_UPN=123253; sug=0; sugstore=1; ORIGIN=0;
bdime=0; BAIDUID=990D38B58C25B6D242326E01E41DE52B:SL=0:N
R=10:FG=1; BDUSS=Z0dzlpazZ6fkxUZ0VCVUxJOGZEaFlyMjlLWElse
jRnSHJYeklhdjhNVDczbUpiQVFBQUFBJCQAAAAAAAAAAAEAAAAMGd9Gc
HJvdG9idWYAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAPtRO1v7UTtbUG; MCITY=-%3A; B
DORZ=B490B5EBF6F3CD402E515D22BCDA1598; H_PS_645EC=f84bV
F%2BHbT1G7YGah73qxunKQwbBDqgfxQIwjfle2wVBGeeDhwvpvMS4N8
73 requests | 47.1 KB transferred


Elements
Console
Sources
Network
Performance
Memory
Filter
Hide data URLs
XHR JS CSS Img Media Font Doc WS Manifest Other
200000 ms
400000 ms
600000 ms
800000 ms
1000000 ms
1200000 ms
Name
Headers
Preview
Response
Cookies
Timing
www.baidu.com
General
Response Headers
view source
Bdpagetype: 2
Bdqid: 0xe209f4a00002089c
Cache-Control: private
http的响应头信息
Connection: Keep-Alive
Content-Encoding: gzip
Content-Type: text/html;charset=utf-8
Date: Thu, 22 Nov 2018 08:46:51 GMT
Expires: Thu, 22 Nov 2018 08:46:51 GMT
Server: BWS/1.1
Set-Cookie: BDSVRTM=98; path=/
Set-Cookie: BD_HOME=1; path=/
Strict-Transport-Security: max-age=172800
Transfer-Encoding: chunked
X-Ua-Compatible: IE=Edge,chrome=1
Request Headers (9)
75 requests | 48.6 KB transferred


Elements
Console
Sources
Network
Performance
Memory
Filter
Hide data URLs
XHR JS CSS Img Media Font Doc WS Manifest Other
500 ms
1000 ms
1500 ms
2000 ms
Name
Headers
Preview
Response
Cookies
Timing
www.baidu.com
http的响应体信息
93 requests | 175 KB transferred
Line 34, Column 9

知识要点

谷歌浏览器的开发者工具是查看http协议的通信过程利器，通过Network标签选项可以查看每一次的请求和响应的通信过程，调出开发者工具的通用方法是在网页右击选择检查。

Headers选项总共有三部分组成：
General：主要信息
Response Headers：响应头
Request Headers：请求头

Response选项是查看响应体信息的

面试要点

简述一个请求的生命周期

TCP/IP分层模型

OSI参考模型	TCP/IP协议簇	对应的网络协议	传送的数据类型	使用的中间设备	
应用层	应用层	HTTP、TFTP、FTP、NFS、WAIS、SMTP	报文	网关	端对端
表示层		Telnet、Rlogin、SNMP、Gopher			
会话层		SMTP、DNS			
传输层	传输层	TCP、UDP	TCP：报文段 UDP：用户数据报		
网络层	网络层	IP、ICMP、ARP、RARP、AKP、UUCP	数据包	路由器	
数据链路层	数据链路层	FDDI、Ethernet、Arpanet、PDN、SLIP、PPP	帧	网桥	点对点
物理层		IEEE 802.1A、IEEE 802.2到IEEE 802.11	比特	转发器	

在浏览器中输入www.baidu.com后执行的全部过程

1、客户端浏览器通过DNS解析到www.baidu.com的IP地址220.181.27.48，通过这个IP地址找到客户端到服务器的路径。客户端浏览器发起一个HTTP会话到220.161.27.48，然后通过TCP进行封装数据包，输入到网络层。

2、在客户端的传输层，把HTTP会话请求分成报文段，添加源和目的端口，如服务器使用80端口监听客户端的请求，客户端由系统随机选择一个端口如5000，与服务器进行交换，服务器把相应的请求返回给客户端的5000端口。然后使用IP层的IP地址查找目的端。

3、客户端的网络层不用关心应用层或者传输层的东西，主要做的是通过查找路由表确定如何到达服务器，期间可能经过多个路由器，这些都是由路由器来完成的工作，我不作过多的描述，无非就是通过查找路由表决定通过那个路径到达服务器。

4、客户端的链路层，包通过链路层发送到路由器，通过邻居协议查找给定IP地址的MAC地址，然后发送ARP请求查找目的地址，如果得到回应后就可以使用ARP的请求应答交换的IP数据包现在就可以传输了，然后发送IP数据包到达服务器的地址。

事件顺序

(1) 浏览器获取输入的域名www.baidu.com

(2) 浏览器向DNS请求解析www.baidu.com的IP地址

(3) 域名系统DNS解析出百度服务器的IP地址

(4) 浏览器与该服务器建立TCP连接(默认端口号80)

(5) 浏览器发出HTTP请求，请求百度首页

(6) 服务器通过HTTP响应把首页文件发送给浏览器

(7) TCP连接释放

(8) 浏览器将首页文件进行解析，并将Web页显示给用户。

涉及到的协议

(1) 应用层：HTTP(WWW访问协议)，DNS(域名解析服务)

DNS解析域名为目的IP，通过IP找到服务器路径，客户端向服务器发起HTTP会话，然后通过运输层TCP协议封装数据包，在TCP协议基础上进行传输

(2) 传输层：TCP(为HTTP提供可靠的数据传输)，UDP(DNS使用UDP传输)
HTTP会话会被分成报文段，添加源、目的端口；TCP协议进行主要工作

(3)网络层：IP(IP数据数据包传输和路由选择)，为数据包选择路由，IP协议进行主要工作

(4)数据链路层：ICMP(提供网络传输过程中的差错检测)，ARP(将本机的默认网关IP地址映射成物理MAC地址)
相邻结点的可靠传输，ARP协议将IP地址转成MAC地址。

robots协议（了解）

> 君子协议

robots协议也称爬虫协议、爬虫规则等，是指网站可建立一个robots.txt文件来告诉搜索引擎哪些页面可以抓取，哪些页面不能抓取，而搜索引擎则通过读取robots.txt文件来识别这个页面是否允许被抓取。但是，这个robots协议不是防火墙，也没有强制执行力，搜索引擎完全可以忽视robots.txt文件去抓取网页的快照。 [5] 如果想单独定义搜索引擎的漫游器访问子目录时的行为，那么可以将自定的设置合并到根目录下的robots.txt，或者使用robots元数据（Metadata，又称元数据）。
robots协议并不是一个规范，而只是约定俗成的，所以并不能保证网站的隐私。