# HTTP协议

# HTTP协议概述

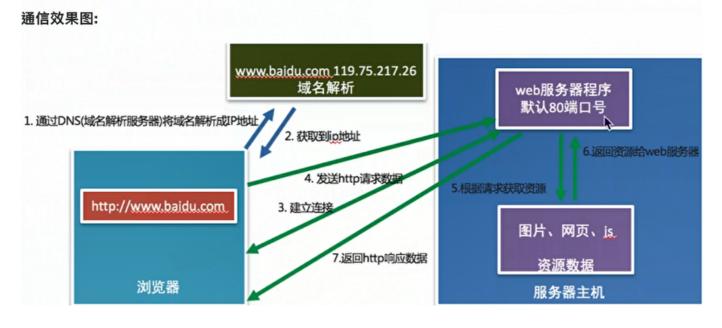
### 一、HTTP协议的介绍

- HTTP协议的全称是(HyperText Transfer Protocol),翻译过来就是超文本传输协议。
- 超文本是超级文本的缩写,是指超越文本限制或者超链接,比如:图片、视频、音乐、超链接等等都属于超文本。
- HTTP协议的制作者是蒂姆.伯纳斯.李,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。

### 二、HTTP协议的作用

它规定了浏览器和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

三、浏览器访问web服务器的通信过程



### 四、小结

- HTTP协议是一个超文本传输协议
- HTTP协议是一个基于TCP传输协议传输数据的
- HTTP协议规定了浏览器和web服务器通信数据的格式,所有超文本数据必须封装成指定的http数据格式

### **URL**

一、URL的概念

URL的英文全称是(Uniform Resource 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,也就是我们常说的网址。

### 二、URL的组成

### https://news.163.com/18/1122/10/E178J2O4000189FH.html

• 协议部分: http://、https://、ftp://

• 域名部分: news.163.com

• 资源路径部分: /18/1122/10/E178J2O4000189FH.html

https默认端口号443,http默认80端口

域名:就是IP地址的别名,他是用点进行分割使用英文字母和数字组成的名字,使用域名的目的就是方便记忆某台主机的IP地址。

### URL的扩展:

https://news.163.com/hello.html?page=1&count=10

查询参数部分: ?page = 1 & count = 10

### 参数部分:

• ? 后面的page表示第一个参数,后面的参数都使用&进行连接

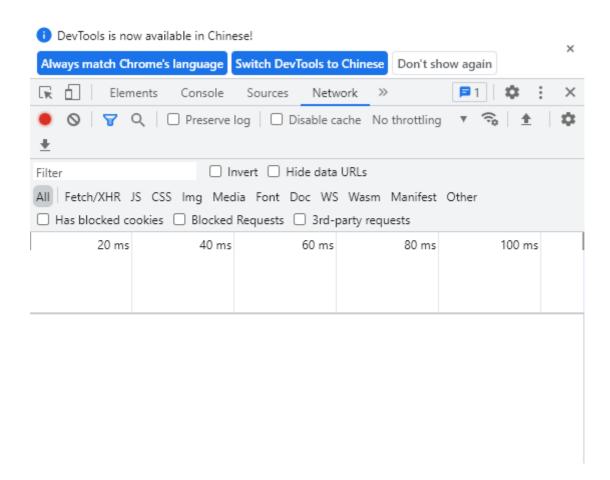
### 三、小结

- URL就是网络资源的地址, 简称网址, 通过URL能够找到网络中对应的资源数据。
- 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

# 查看HTTP协议的通信过程

一、谷歌浏览器开发者工具

首先需要安装Google Chrome浏览器,然后Windows和Linux平台按F12调出开发者工具。

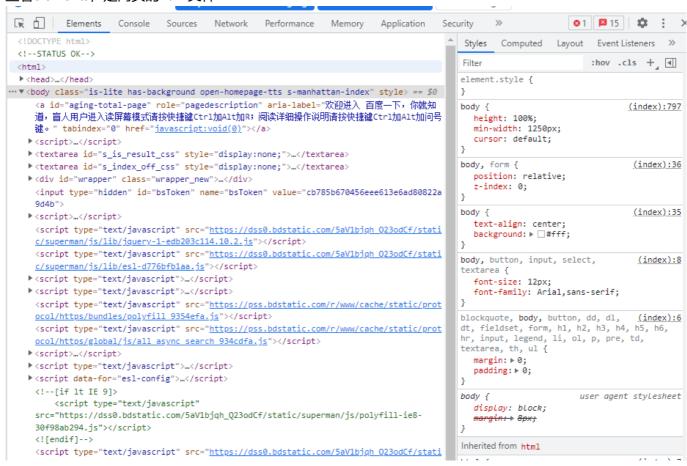


### 开发者工具的标签选项说明:

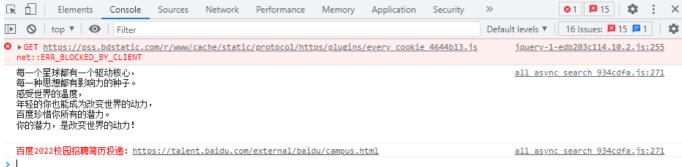
- 元素 (ELements):用于查看或者修改HTML标签
- 控制台(Console): 执行js代码
- 源代码 (Sources): 查看静态资源文件, 断点调试js代码
- 网络(Network):查看http协议的通信过程

### 输入www.baidu.com,可以看到浏览器发出54个请求,下面的每一行都是一 i DevTools is now available in Chinese! Always match Chrome's language Switch DevTools to Chinese Don't show again o 1 🔼 15 🔯 Elements Console Sources Network Performance Memory Application Security Ċ ☐ Invert ☐ Hide data URLs ☐ Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other ☐ Has blocked cookies ☐ Blocked Requests ☐ 3rd-party requests 100 ms 200 ms 300 ms 400 ms 500 ms 600 ms 700 ms 800 ms 900 ms 1000 ms 1100 ms 1200 ms 130 Initiator Waterfall Status Туре Time www.baidu.com 200 document Other 87.7 kB 313 ms newfanyi-da0cea8f7e.png 200 (disk cache) 3 ms png (index) newxueshuicon-a5314d5c83.png 200 (disk cache) 2 ms png (index) jquery-1-edb203c114.10.2.js 200 21 ms script (index) (disk cache) esl-d776bfb1aa.js 200 16 ms script (index) (disk cache) opolyfill\_9354efa.js 200 script (index) (disk cache) 16 ms - III all\_async\_search\_934cdfa.js 200 (index) 489 B 132 ms script ф sbase-47057af807.js 200 (disk cache) 16 ms script (index) min\_super-7ddd157405.js 200 (disk cache) 16 ms script (index) s\_super\_async-5488e156f6.js 200 19 ms script (index) (disk cache) newwenku-d8c9b7b0fb.png 200 png (index) (disk cache) 16 ms newbaike-889054f349.png 200 png (index) (disk cache) 15 ms newzhidao-da1cf444b0.png 200 png (index) (disk cache) 15 ms newjiankang-f03b804b4b.png 200 png (index) (disk cache) 14 ms yingxiaoicon-612169cc36.png 200 png (index) (disk cache) 14 ms newzhibo-a6a0831ecd.png 200 png (index) (disk cache) 14 ms newyinyue-03ecd1e9b9.png 200 (index) (disk cache) 14 ms png a7.png 200 (index) (disk cache) 14 ms png fca5686667796478467696 200 (index) (disk cache) 13 ms jpeg logo\_white-d0c9fe2af5.png 200 (disk cache) 13 ms png (index) PC\_880906d2a4ad95f5fafb2e540c5cdad7.png 200 (disk cache) 14 ms png (index) $\Rightarrow$ result.png (index) (disk cache) 12 ms pnq ф = result@2.png (index) (disk cache) 12 ms pnq peak-result.png (index) (disk cache) 11 ms pna 477 B cd37ed75a9387c5b.js 78 ms script (index) 64102786834.jpg?x-bce-process=image/cro... 495 B 148 ms ipeg (index) 🗈 data:image/png;base... 200 (memory ca... 0 ms ı (index) png data:image/png;base... 200 (memory ca... 0 ms (index) png iconfont-3585cbf68d.woff2 200 1 (disk cache) 2 ms font (index) every\_cookie\_4644b13.js <u>jquery-1-edb203c114...</u> 0 B (blocked:ot., 3 ms script bzPopper\_7bc4f0e.js 200 485 B 39 ms esl-d776bfb1aa.js:33 script nu\_instant\_search\_ac49315.js 200 jquery-1-edb203c114. script (disk cache) 2 ms swfobject\_0178953.js 200 script esl-d776bfb1aa.js:33 (disk cache) 2 ms 54 requests | 93.9 kB transferred | 2.5 MB resources | Finish: 1.11 s | DOMContentLoaded: 749 ms | Load: 938 ms

### 查看elements, 是网页的html文件

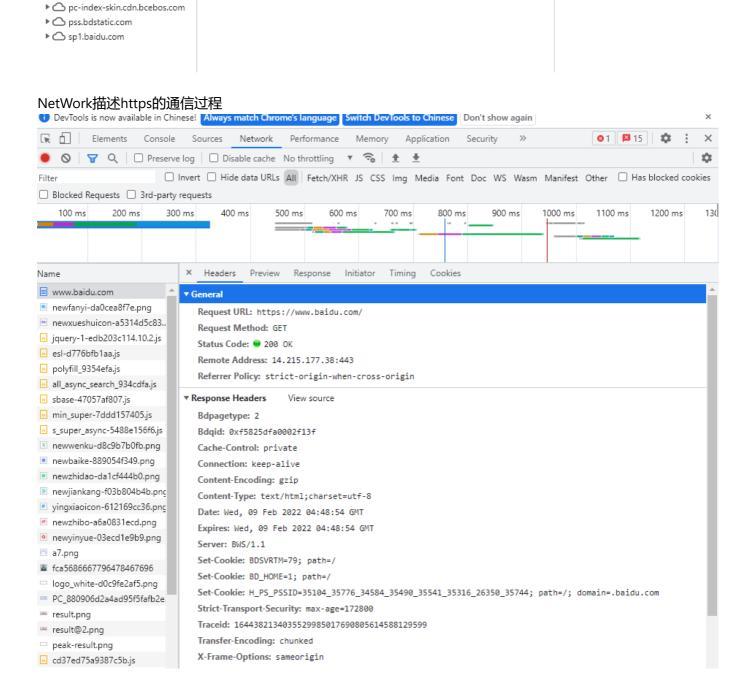


### console是控制台可以执行is代码。



I





2022/2/9 HTTP协议的介绍.md

### headers是https协议的主要信息:

```
▼ General
   Request URL: https://www.baidu.com/
   Request Method: GET
   Status Code: 9 200 OK
   Remote Address: 14.215.177.38:443
   Referrer Policy: strict-origin-when-cross-origin
```

可以清楚的看到https协议的默认端口号就是443,请求方式是get。

### Request URL 请求头信息,

```
Request Headers
                     View parsed
```

```
GET / HTTP/1.1
Host: www.baidu.com
Connection: keep-alive
sec-ch-ua: "Not; A Brand"; v="99", "Google Chrome"; v="97", "Chromium"; v="97"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.
4692.99 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.
application/signed-exchange; v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Cookie: BIDUPSID=596549FEBD7BB0978B443F126131E34C; PSTM=1641627619; BAIDUID=596549FEBD7BB09722355A51D87
6B1:FG=1; BD_UPN=12314753; __yjs_duid=1_92f5af7da5389645f0f08d551708100e1641810903990; BDORZ=B490B5EBF6
```

CD402E515D22BCDA1598; BAIDUID\_BFESS=E303719578E0B4F68F31A5F5AF91AE0D:FG=1; BDUSS=FFsdUNBT1dpVT1wcn1Fd11 isitId=c78d9226-041c-4423-8721-952fa007e3a5; ab\_sr=1.0.1\_MjBhMTg0YjllNDY0MzkzNjUwN2MzZGJlZGM4MzUwNzU0Nz

view source是查看浏览器查看发送给web服务器的原始数据, view parsed是查看浏览器发送给web服务器解析 过的数据。

### Reponse代表响应体,是服务器发送给浏览器的数据,浏览器使用这些数据渲染成百度的网页

```
× Headers Preview
                     Response Initiator Timing Cookies
   1 <!DOCTYPE html><!--STATUS OK-->
          <html><head><meta http-equiv="Content-Type" content="text/html;charset=utf-8"><meta http-equiv="X-</p>
   4 <style index="newi" type="text/css">#form .bdsug{top:39px}.bdsug{display:none;position:absolute;width:
   6 <style type="text/css" index="head">blockquote,body,button,dd,dl,dt,fieldset,form,h1,h2,h3,h4,h5,h6,hr
   7 html{color:#000;overflow-y:scroll;overflow:-moz-scrollbars}
   8 body, button, input, select, textarea (font-size: 12px; font-family: Arial, sans-serif)
   9 h1,h2,h3,h4,h5,h6{font-size:100%}
  10 em{font-style:normal}
  11 small{font-size:12px}
  12 ol,ul{list-style:none}
  13 a{text-decoration:none}
  14 a:hover{text-decoration:underline}
  15 legend{color:#000}
  16 fieldset, img{border:0}
  17 button,input,select,textarea{font-size:100%}
  18 table{border-collapse:collapse;border-spacing:0}
  19 img{-ms-interpolation-mode:bicubic}
  20 textarea{resize:vertical}
  21 .left{float:left}
  22 .right{float:right}
  23 .overflow{overflow:hidden}
  24 .hide{display:none}
  25 .block{display:block}
  26 .inline{display:inline}
  27 .error{color:red;font-size:12px}
  28 button,label{cursor:pointer}
  29 .clearfix:after{content:'\20';display:block;height:0;clear:both}
  30 .clearfix{zoom:1}
  31 .clear{clear:both;height:0;line-height:0;font-size:0;visibility:hidden;overflow:hidden}
  32 .wordwrap{word-break:break-all;word-wrap:break-word}
  33 .s-yahei{font-family:arial, 'Microsoft Yahei', '微软雅黑'}
  34 pre.wordwrap{white-space:pre-wrap}
  35 body{text-align:center;background:#fff}
  36 body,form{position:relative;z-index:0}
  37 td{text-align:left}
  38 img{border:0}
  39 #s_wrap{position:relative;z-index:0;min-width:1000px}
  40 #wrapper{height:100%}
  41 #head .s-ps-islite{_padding-bottom:370px}
  42 #head_wrapper.s-ps-islite{padding-bottom:370px}
  43 #head_wrapper.s-ps-islite #s_lm_wrap{bottom:298px;background:0 0!important;filter:none!important}
```

### 二、小结

- 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标签选项就可以查看每一次的请求和响应的过程,调出开发者工具的通用方法是网页右击选择检查。
- 开发者工具的Headers选项总共有三个部分组成:
  - 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

# HTTP请求报文

一、HTTP请求报文介绍

HTTP最常见的请求报文有两种:

1. Get方式的请求报文

### 2. Post方式的请求报文

### 说明:

• Get: 获取web服务器数据

• Post: 向web服务器提交数据 (常用于用户登录)

### 二、HTTP GET 请求报文分析

请求报文头部信息 (每一项信息之间都有/r/n 进行分割)

```
--http get请求报文
---请求行---
GET / HTTP/1.1 => 请求方法 (方式) 请求的资源路径 (/ 代表主页) http协议的版本
---请求头---
Host: www.baidu.com => 服务器的主机ip地址和端口号,提示如果看不到端口号,默认是80
Connection: keep-alive => 和服务端程序保持长连接, 当客户端和服务端有一段时间没有进行通
信,那么服务器程序会主动向客户端断开连接,这个时间大概是三到五分钟
Cache-Control: max-age=0
sec-ch-ua: " Not; A Brand"; v="99", "Google Chrome"; v="97", "Chromium"; v="97"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1 => 让客户请求不安全请求,以后要使用https
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/97.0.4692.99 Safari/537.36 => 客户端代理, 客户端程序的名称
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9 => 告诉服务端程序可以接受的数
据类型
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br => 告诉服务端程序 支持的压缩算法
Accept-Language: zh-CN,zh;q=0.9 => 告诉服务端程序支持的语言
Cookie: BIDUPSID=596549FEBD7BB0978B443F126131E34C; PSTM=1641627619;
BAIDUID=596549FEBD7BB09722355A51D87326B1:FG=1; BD UPN=12314753;
yjs duid=1 92f5af7da5389645f0f08d551708100e1641810903990;
BDORZ=B490B5EBF6F3CD402E515D22BCDA1598;
BDUSS=FFsdUNBT1dpVT1wcn1Fd112cm5wQ3FQM2gtN1AwLTZXREVuNTVRMGNHWVcwaWxpSVFBQUFBJCQAA
AAAAAAAAAAAAAAAAAAABZFAmIWRQJiM;
BDUSS BFESS=FFsdUNBT1dpVTlwcn1Fdl12cm5wO3FOM2gtN1AwLTZXREVuNTVRMGNHWVcwaWxpSVFBOUF
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABZFAmIWRQJiM; baikeVisitId=2dd394e3-b4e0-4ffd-baef-
2e398b5b89af; BD HOME=1;
H_PS_PSSID=35104_35776_34584_35490_35541_35316_26350_35744; sug=3; sugstore=0;
ORIGIN=0; bdime=0; BA_HECTOR=2500a521802gag0gaa1h06osu0r => 客户端用户身份的标识
```

---最后还有一个空行---

### 三、小结

- 一个HTTP请求报文可以由请求行、请求头、空行和请求体四个部分组成。
- 请求行是由三个部分组成:
  - 。 请求方式
  - 。 请求资源路径 (一般是主页index.html)
  - 。 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

# 请求行 请求行 请求头 空行 GET方式 方式 请求体 POST方式

# HTTP响应报文

## 一、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---
Bdpagetype: 2
Bdqid: 0x9716a8f60002c67c
Cache-Control: private
Connection: keep-alive => 和客户端保持长连接
Content-Encoding: gzip
Content-Type: text/html;charset=utf-8 => 服务器发送给浏览器的内容类型以及编码格式
Date: Wed, 09 Feb 2022 07:53:25 GMT => 服务器的时间
Expires: Wed, 09 Feb 2022 07:53:25 GMT
Server: BWS/1.1 => 服务器的名称
```

Set-Cookie: BDSVRTM=124; path=/
Set-Cookie: BD\_HOME=1; path=/

Set-Cookie: H\_PS\_PSSID=35104\_35776\_34584\_35490\_35541\_35316\_26350\_35744; path=/;

domain=.baidu.com

Strict-Transport-Security: max-age=172800

Traceid: 1644393205025555405810887074923730355836

X-Frame-Options: sameorigin

X-Ua-Compatible: IE=Edge,chrome=1

Transfer-Encoding: chunked => 服务器发送给客户端程序(浏览器)的数据不确定数据长度,数据发送结束的接受标识: 0\r\n, Content-Length: 200字节 ,服务器发送给客户端程序的数据确

定长度,这两个内容长度选项二者选其一

---空行---

\r\n

---响应体: 就是真正给浏览器解析的数据---

网页数据

### HTTP 状态码是用于表示web服务器响应状态的3位数字代码。

状态码	说明	
200	请求成功	
307	重定向	
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	

## 二、小结

- 一个HTTP响应报文是由响应行,响应头、空行和响应体四个部分组成
- 响应行是由三部分组成: HTTP协议版本 状态码 状态描述, 最常见的状态码就是200 (响应成功)