

1.HTTP协议简介

- HTTP (Hyper Text Transfer Protocol) : 全称超文本传输协议, 是用于从万维网 (WWW:World Wide Web) 服务器传输超文本到本地浏览器的传送协议。
- HTTP 是一种应用层协议, 是基于 TCP/IP 通信协议来传递数据的, 其中 HTTP1.0、HTTP1.1、HTTP2.0 均为 TCP 实现, HTTP3.0 基于 UDP 实现。现主流使用 HTTP1.0 和 HTTP3.0
- 协议: 为了使数据在网络上从源头到达目的, 网络通信的参与方必须遵循相同的规则, 这套规则称为协议, 它最终体现为在网络上传输的数据包的格式。

通俗点讲, 协议就是要保证网络通信的双方, 能够互相对接上号。就像是两个人传递纸条通过互相指定的暗号, 如果发送天亮了, 表示可以打游戏了等等

注意: 当我们访问一些网页时, 是显示通过 HTTPS 来进行通信的, 并且当下大多数的网页都是通过 HTTPS 来进行通信的, 因为 HTTPS 在 HTTP 的基础上做了一个加密的工作。HTTPS 将在本文末尾具体介绍

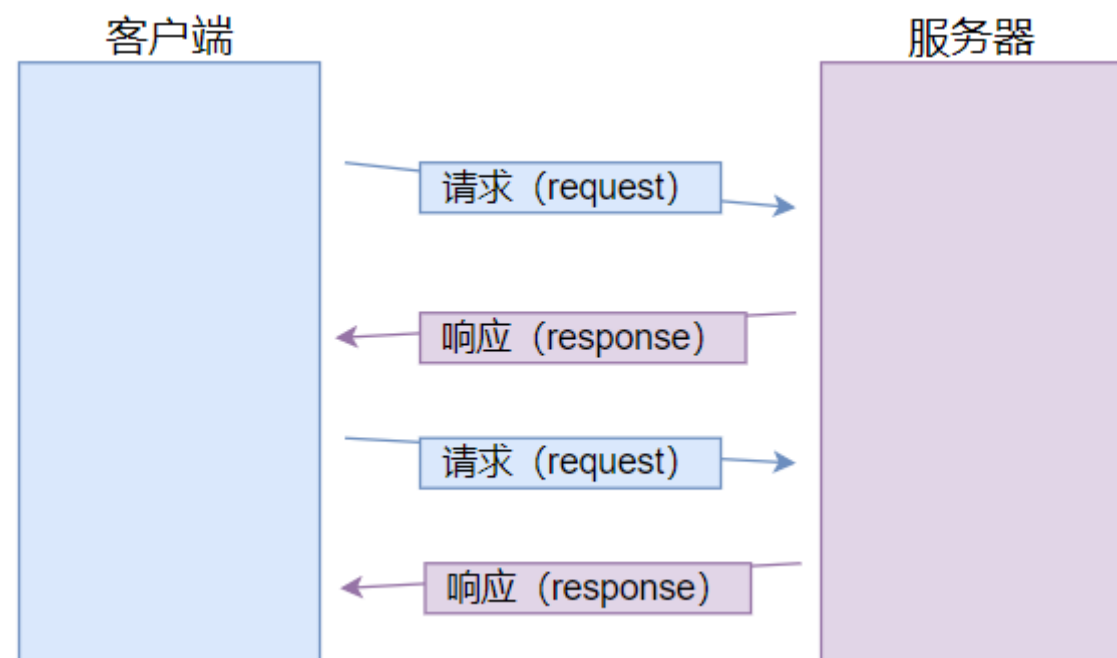
2.HTTP协议工作流程

当我们在浏览器输入一个网址, 此时浏览器就会给对应的服务器发送一个 HTTP 请求, 对应的服务器收到这个请求之后, 经过计算处理, 就会返回一个 HTTP 响应。并且当我们访问一个网站时, 可能涉及不止一次的 HTTP 请求和响应的交互过程。

基础术语:

- 客户端: 主动发起网络请求的一端
- 服务器: 被动接收网络请求的一端
- 请求: 客户端给服务器发送的数据
- 响应: 服务器给客户端返回的数据

HTTP 协议的重要特点: 一发一收, 一问一答



注意:



在网络编程中，除了一发一收之外，还有其它的模式：

- 多发一收：例如上传大文件
- 一发多收：例如看直播时，搜索一个词条可以得到多个视频源
- 多发多收：例如串流（steam link、moonlight 等等）

3.URL介绍

- URI 全称 Uniform Resource Identifier，即统一资源标志符
- URL 全称 Universal Resource Locator，即统一资源定位符

什么是URL？URL 就是我们平常所说的网址，例如 <https://www.baidu.com> 是一个 URL，而 URI 与 URL 概念类似，只不过 URL 是 URI 的一个子集！也就是说每个 URL 都是 URI，但并非每个URI都为 URL。

URI 除了URL之外还包括一个子集，叫作 URN，全称为 Universal Resource Name，即统一资源名称。URN 只为资源命名而不指定如何定位资源，例如 `urn:isbn:0451450642` 指定了一本书的 ISBN，可以标识这本书，但没有指定到哪里获取这本书。

在互联网中，URN 使用的非常少，几乎所有的 URI 都是 URL，所以对于一般的网页链接我们既可以称为 URI，也可以称为 URL，在日常生活中是称 URL 的多一点。

3.1.URL格式

URL 的书写并不是随意书写的，它是由因特网标准RFC1738进行约定的，可参考<https://datatracker.ietf.org/doc/html/rfc1738>，具体格式如下：

```
scheme://[username:password@]hostname[:port][/path][;parameters][?query][#fragment]
```

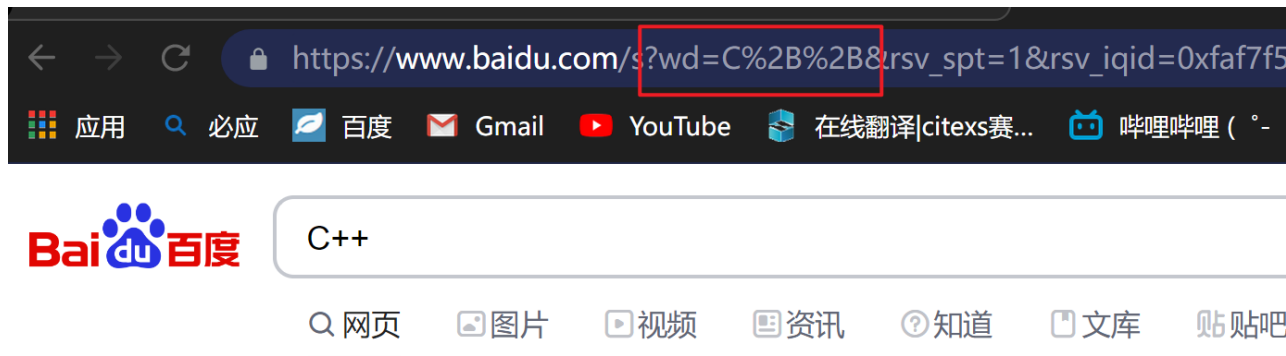
CSDN @爱躺平的威威

在URL规定格式中，括号包括内容是非必要部分，例如 <https://www.baidu.com> 就只包含了 scheme 和 hostname 两部分。但是哪些非必要的部分，并不代表不重要，接下来我将分别介绍每一部分代表的含义。

- scheme：协议。常用的协议有 http、https、ftp等，另外 scheme 也常常被称为 protocol。
- username, password：用户名和密码，有时候登陆时会显示，现在用到比较少
- hostname：主机地址，该地址可以是域名也可以是IP地址，比如 <https://www.baidu.com> 这个 URL 中的 hostname 就是 `www.baidu.com`（百度的二级域名）。
- port：端口，这个就是服务器设定的服务端口，例如 <https://127.0.0.1:8080> 这个 URL 中的端口就是 8080。但是有些 URL 中没有端口信息，这是使用了默认端口，例如 http 协议的默认端口是 80，https 协议的默认端口是443。
- path：路径，指的就是网络资源在服务器中的指定地址
- parameters：参数，用来指定访问某个资源时的附加信息
- query：查询，用来查询某个类资源，如果又多个资源，则用 & 隔开
- fragment：片段，是对资源描述的部分补充。

3.2.URL encode

在URL格式中，像 / ? : @ 等这样的字符，已经被URL当做特殊字符使用。如果 URL 中查询部分涉及到这些特殊字符，就可能导致该 URL 解析错误。这个原理就像编程语言中的 变量名 不能是“关键字”一样。例如在java中，我们不能给一个变量起名为 ‘int’。针对这种情况的发生，我们就会对 URL 中的特殊字符进行转义，这个转义过程就叫 URL encode。例如，我在百度中搜索“C++”，在URL中 ‘+’ 就被转义为 “%2B”。

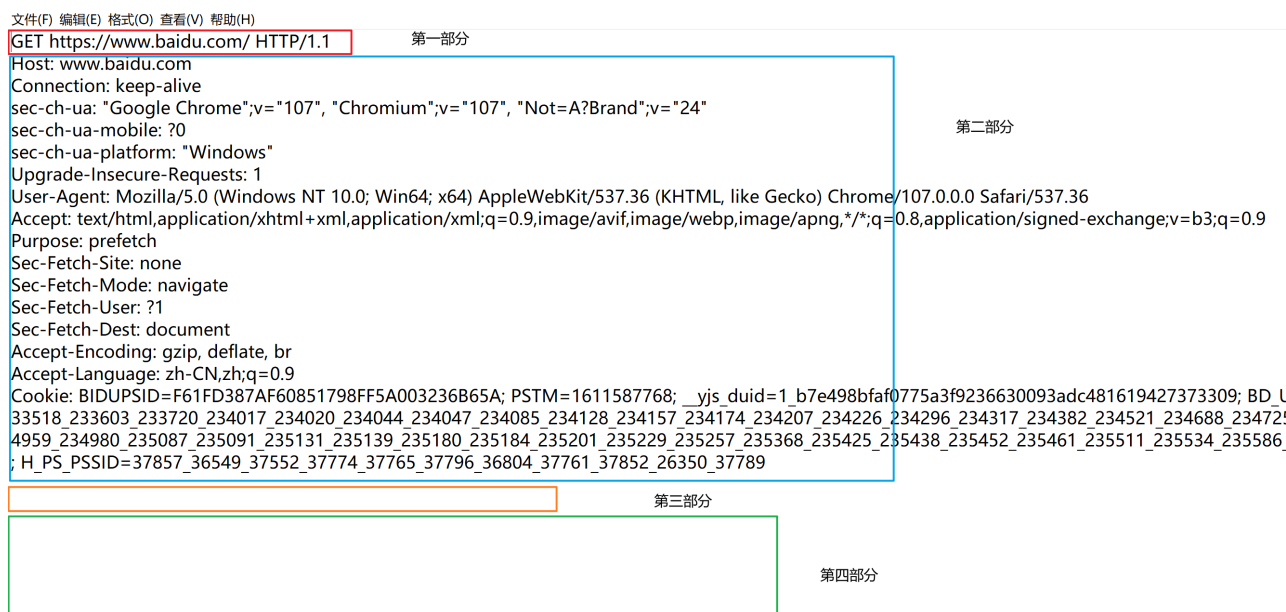


除了特殊字符之外，中文也会进行转义。可参考转义编码网站：http://mytju.com/classcode/tools/encode_utf8.asp

4.HTTP协议格式

HTTP协议格式可以分为两种，HTTP请求报文格式 和 HTTP响应报文格式。

4.1.请求报文格式



4.1.1.第一部分：首行

首行包括：HTTP方法+URL+版本号

GET https://www.baidu.com/ HTTP/1.1

方法

URL

版本号

CSDN @爱躺平的威威

1. HTTP方法，也就是你想通过该请求做些什么事情，常见的方法有 GET、POST，这些具体会在 请求头 中介绍。
2. URL 请求的网址
3. 版本号，一般情况下会省略

4.1.2.第二部分：请求头header

请求头，主要由一个个键值对组成。每一行，是一个键值对，键和值之间使用“:”来分割。

```
Host: www.baidu.com
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Google Chrome";v="108"
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/108.0.0.0 Safari/537.36
Accept: 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
Purpose: prefetch
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=F61FD387AF60851798FF5A003236B65A; PSTM=1611587768; __yjs_duid=1_b7e498bfaf0775a3f9236630093adc481619427373309; BD_UPN=12314753; 33518_233603_233720_234017_234020_234044_234047_234085_234128_234157_234174_234207_234226_234296_234317_234382_234521_234688_234725_234749_234784959_234980_235087_235091_235131_235139_235180_235184_235201_235229_235257_235368_235425_235438_235452_235461_235511_235534_235586_235741_235774
```

4.1.3.第三部分：空行

空行 代表 header 的结束的标记。一个HTTP请求的 header 可以有若干个，官方就用空行，作为 header 的结束标记，类似于链表的 null。

4.1.4.第四部分：正文 body

正文部分并不是所有的请求都有的，一些需要提交数据的请求中就会包含。例如，我正在写这篇文章，我抓取该网页请求，就能看见请求中正文。正文格式有很多格式，常见的就是json。

[illegible]

The diagram illustrates the structure of an HTTP response, divided into four main sections:

- 第一部分 (Part 1):** The status line, highlighted with a red border, is "HTTP/1.1 200 OK".
- 第二部分 (Part 2):** The headers, highlighted with a blue border, include:
 - Bdpagetype: 2
 - Bdqid: 0xd4b8b7df0005ca36
 - Cache-Control: private
 - Connection: keep-alive
 - Content-Type: text/html;charset=utf-8
 - Date: Thu, 01 Dec 2022 06:51:47 GMT
 - Expires: Thu, 01 Dec 2022 06:51:46 GMT
 - Isprivate: 1
 - Server: BWS/1.1
 - Set-Cookie: BDSVRTM=499; path=/
Set-Cookie: BD_HOME=1; path=/
Set-Cookie: H_PS_PSSID=37857_36549_37552_37774_37765_37796_36804_37761_37852_26350_37789; path=;/; d
 - Strict-Transport-Security: max-age=172800
 - Traceid: 1669877507284930125815328203500161452598
 - X-Frame-Options: sameorigin
 - X-Ua-Compatible: IE=Edge,chrome=1
 - Content-Length: 463866
- 第三部分 (Part 3):** The content type, highlighted with an orange border, is "text/html".
- 第四部分 (Part 4):** The HTML body content, highlighted with a green border, starts with the DOCTYPE declaration and includes meta tags for content type and X-UA-Compatible, followed by a title and style tags.

首行包括：版本号+状态码+状态码描述

HTTP/1.1 200 OK

版本号 状态码 描述

1. 版本号，与请求首行里的版本号相同
2. 状态码，描述当前请求的“状态”，例如200请求成功、404未找到
3. 状态码描述，对状态码的解释说明

4.2.2.第二部分：响应头 header

与请求中header一样，都是由多个键值对组成。每个键值对占一行，键和值之间使用：空格来分割。

```
Bdpagetype: 2
Bdqid: 0x87adf6f3000058c8
Cache-Control: private
Connection: keep-alive
Content-Type: text/html;charset=utf-8
Date: Sun, 18 Dec 2022 11:39:31 GMT
Expires: Sun, 18 Dec 2022 11:39:31 GMT
Isprivate: 1
Server: BWS/1.1
Set-Cookie: BDSVRTM=493; path=/
Set-Cookie: BD_HOME=1; path=/
Set-Cookie: H_PS_PSSID=37857_36549_37552_37683_37908_37796_36804_37926_37852_37902_26350_37789_37881; path=/; domain=.baic
Strict-Transport-Security: max-age=172800
Traceid: 167136357101984355949776741889628723400
X-Frame-Options: sameorigin
X-Ua-Compatible: IE=Edge,chrome=1
Content-Length: 469601
```

4.2.3.第三部分：空行

header 的结束标记

```
Cache-Control: private
Connection: keep-alive
Content-Type: text/html;charset=utf-8
Date: Sun, 18 Dec 2022 11:39:31 GMT
Expires: Sun, 18 Dec 2022 11:39:31 GMT
Isprivate: 1
Server: BWS/1.1
Set-Cookie: BDSVRTM=493; path=/
Set-Cookie: BD_HOME=1; path=/
Set-Cookie: H_PS_PSSID=37857_36549_37552_37683_37908_37796_36804_37926_37852_37902_
Strict-Transport-Security: max-age=172800
Traceid: 167136357101984355949776741889628723400
X-Frame-Options: sameorigin
X-Ua-Compatible: IE=Edge,chrome=1
Content-Length: 469601
```

空行

```
<!DOCTYPE html> <!--STATUS OK-->
```

```
<html> <head> <meta http-equiv="Content-Type" content="text/html;charset=utf-8"> <met
01_1660835115000.png"> <title>百度一下，你就知道</title>
```

```
<style index="new" type="text/css"> #form_bdcua{top:20px} bdcua{display:none;position:abs
```

4.2.4.第四部分：正文body

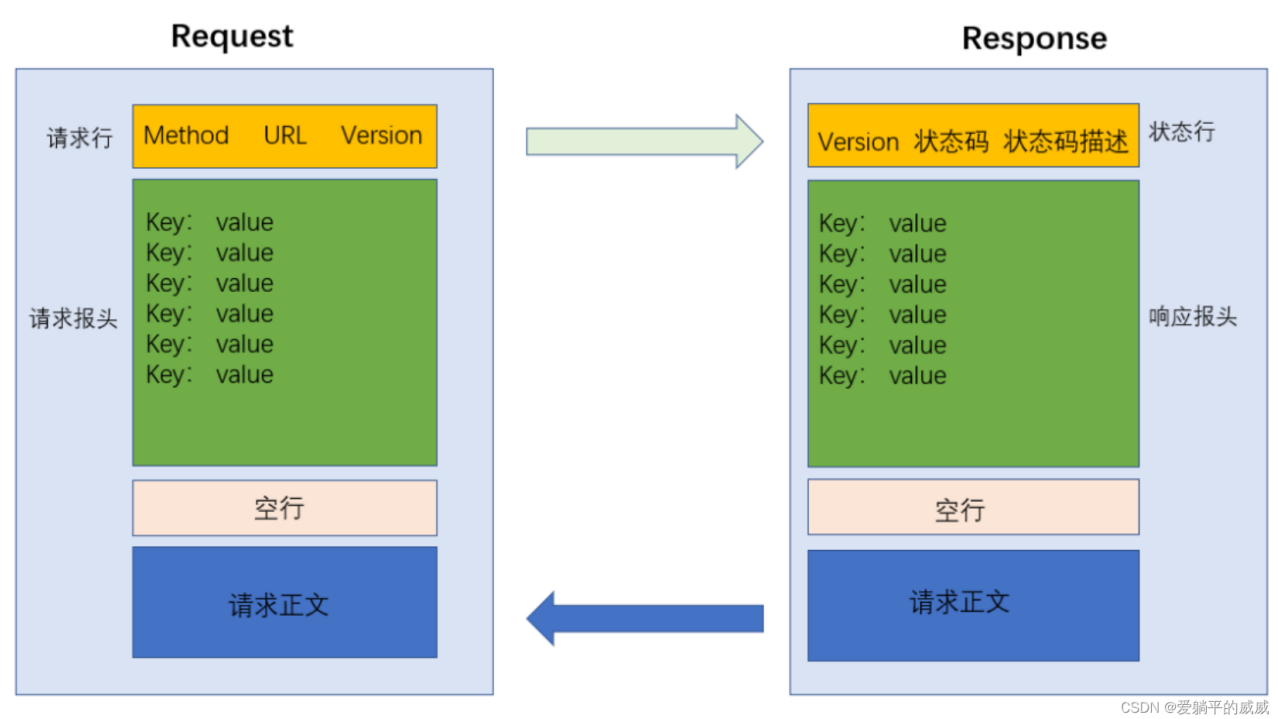
正文格式有很多中，常见的就是 json、html等，此处访问百度的响应报文中正文就是html，如果是登陆页面的话json格式的多一点。

```
<!DOCTYPE html> <!--STATUS OK-->

<html> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> <meta http-equiv="X-UA-Compatible" content="IE=
01_1660835115000.png"> <title>百度一下，你就知道</title>
<style index="newi" type="text/css"> #form .bdsug{top:39px}.bdsug{display:none;position:absolute;width:535px;background:#fff;border:1px solid #ccc;
g-direct p{color:#00c;font-weight:700;line-height:34px;padding:0 8px;margin-top:0;cursor:pointer;white-space:nowrap;overflow:hidden}#m .bd
height:22px;background-color:#f5f5f5}.bdsug .bdsug-pc-direct-tip-is{position:absolute;right:15px;top:3px;width:55px;height:15px;display:block
1d1d1}.bdpfmenu{font-size:12px;background-color:#fff}.bdpfmenu a,.usermenu a{display:block;text-align:left;margin:0!important;padding:0 9px
d solid)#wrapper .bdnuarrow em{border-bottom-color:#d8d8d8;top:-1px}#wrapper .bdnuarrow i{border-bottom-color:#fff;top:0}#gxsHead .f
t-family:Arial,"PingFang SC","Microsoft YaHei",sans-serif;z-index:1}#head_wrapper.sam_head_wrapper2 #form .bdsug-new{width:545px;z-index:
{background-color:#F5F5F6!important}.wrapper_new #form .sam_search .bdsug-new .bdsug-s{background-color:#F1F3FD!important}#head_w
dsug-new{width:100%;box-shadow:0 4px 0px rgba(0,0,0,.1);border:1px solid rgba(0,0,0,.05)!important;border-radius:12px;top:56px}#head_w
us:8px;height:32px;width:76px;text-align:center;line-height:32px;font-size:13px;color:#FFF;position:absolute;z-index:2;top:50px}.sam_search .sa

<style type="text/css" index="head"> blockquote,body,button,dd,dl,dt,fieldset,form,h1,h2,h3,h4,h5,h6,hr,input,legend,li,ol,p,pre,td,textarea,th,
html{color:#000;overflow-y:scroll;overflow:-moz-scrollbars}
body,button,input,select,textarea{font-size:12px;font-family:Arial,sans-serif}
h1,h2,h3,h4,h5,h6{font-size:100%}
em{font-style:normal}
small{font-size:12px}
ol,ul{list-style:none}
a{text-decoration:none}
a:hover{text-decoration:underline}
legend{color:#000}
```

4.3.协议格式总结



5.HTTP请求方法

HTTP 中的方法，就是 HTTP 请求报文中的首行的第一个部分。原本 HTTP 的设计者，是希望通过不同的方法，来表达不同的语义。但是至今，其实也没有被实现，以下具体的方法具体起到了什么作用，完全看程序员在代码中是如何实现的。虽然 HTTP 中的方法很多，但是最常用的就两个 GET 和 POST。以下主要介绍这两个方法：

方法	说明	适用版本号
----	----	-------

GET	获取资源	HTTP 1.0、HTTP 1.1
POST	传输实体主体	HTTP 1.0、HTTP 1.1
PUT	传输文件	HTTP 1.0、HTTP 1.1
HEAD	获得报文首部	HTTP 1.0、HTTP 1.1
DELETE	删除文件	HTTP 1.0、HTTP 1.1
OPTIONS	访问支持的方法	HTTP 1.1
TRACE	追踪路径	HTTP 1.1
CONNECT	要求用隧道协议连接代理	HTTP 1.1
LINK	建立和资源之间的联系	HTTP 1.1
UNLINE	断开连接关系	HTTP 1.1

5.1.GET 方法

5.1.1.基本介绍

GET 是最常用的 HTTP 方法，常用于获取服务器上的某个资源。以下几种方式都会触发 GET 方法的请求

- 在浏览器中直接输入 URL 回车或点击浏览器收藏夹中的链接，此时浏览器就会发送出一个 GET 请求。
- HTML 中的 link、img、script 等标签的属性中放的 URL，浏览器也会构造出 HTTP GET 请求
- 使用 Javascript 重点 ajax，也能构造出 HTTP GET 请求
- 各种编程语言（只要能够访问网络），就都能够构造出 HTTP GER 请求

5.1.2.GET 请求的特点

- 首行里面的第一个部分就是 GET
- URL 里面的 query string 可以为空，也可以不为空
- GET 请求的 header 有若干个键值对结构
- GET 请求的 body 一般是空的

5.1.3.GET请求示例

搜狗首页请求：


```
GET https://www.sogou.com/ HTTP/1.1
Host: www.sogou.com
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36
Accept: 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
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Cookie: SUV=003B56A6DA4C2A82610BB3A8CFD5D583; SUID=DAE58074A021B00A0000000061CDAA89; ABTEST=0|1646288379|v17; browerV=3; osV=1; SNUID=13197C8FFCFE25B13BF697ABFCA
CSDN @吞吞吐吐大魔王
```

5.2.POST 方法

5.2.1.基本介绍

POST 方法也是一种常见的方法，多用于提交用户输入的数据给服务器（如登录页面）。以下几种方法都会触发 POST 方法的请求

- 通过 HTML 中的 form 标签可以构造 POST 请求
- 使用 JavaScript 的 ajax 可以构造 POST 请求

5.2.2.POST 请求的特点

- 首行第一个部分就是 POST
- URL 里面的 query string 一般是空的
- POST 请求的 header 里面有若干个键值对
- POST 请求的 body 一般不为空（body 的具体数据格式，由 header 中的 Content-Type 来描述；body 的具体数据长度，由 header 中的 Content-Length 来描述

5.2.3.POST请求示例

QQ邮箱登录请求

```
POST https://t.captcha.qq.com/cap_union_new_verify HTTP/1.1
Host: t.captcha.qq.com
Connection: keep-alive
Content-Length: 3838
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98"
Accept: application/json, text/javascript, */*; q=0.01
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36
sec-ch-ua-platform: "Windows"
Origin: https://t.captcha.qq.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://t.captcha.qq.com/cap_union_new_show?aid=522005705&protocol=https&accver=1&showtype=embed&ua=TW96aWxsYS81LjAgKFdpbmRvd3MgTiQgMTAuMDsgV2luNjQ7IHg2NCkU2oKfmpnp4VvOFbHCF881E8lrztOhiAiyGkZ4GOcX48Wq+CkrVNNCC4XsmHqaTYJG0PBwXSci565AYRoXfGdtGlqI85I48723uyQj3eq98VTunsSvLZcph5DMrUpGms06iTp2H1z4whFXV7iCVQ+G3pFYwmo+
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Cookie: RK=hXqV4Zj75+; ptcz=7e55ccdd09bcf8dacfb5389288635416e70b4d8f0866a3a1df56fd6dbcac1046; tvfe_boss_uid=ade1b35e80bb9f2a; pgv_pvid=8674286020; TDC_itoken=210015779%3
aid=522005705&protocol=https&accver=1&showtype=embed&ua=TW96aWxsYS81LjAgKFdpbmRvd3MgTiQgMTAuMDsgV2luNjQ7IHg2NCkU2oKfmpnp4VvOFbHCF881E8lrztOhiAiyGkZ4GOcX48Wq+CkrVNNCC4XsmHqaTYJG0PBwXSci565AYRoXfGdtGlqI85I48723uyQj3eq98VTunsSvLZcph5DMrUpGms06iTp2H1z4whFXV7iCVQ+G3pFYwmo+
```

5.3.其他方法

- PUT：与 POST 相似，但是具有幂等特性，一般用于更新
- DELETE：删除服务器指定资源
- OPTIONS：返回服务器所支持的请求方法
- HEAD：与 GET 类似，只不过响应体不返回，只返回响应头

- TRACE：能显示服务器端收到的请求，测试的时候会用到
- CONNECT：预留，暂无使用

6.HTTP响应

响应，就是由服务器返回给客户端，可以分为三个部分： 响应状态码、响应头、响应体。

6.1.响应状态码

状态码，是一个数字，这个数字描述了当前这次请求的“状态”（成功，失败，失败的原因）。下面我们列举出来了一些常用的响应状态码，更多细节请参考官方文档：<https://www.runoob.com/http/http-status-codes.html>

6.1.1.信息响应

状态码	说明	详情
100	继续	请求者应当继续提出请求。服务器已收到请求的一部分，正在等待其余部分
101	切换协议	请求者已要求服务器切换协议，服务器已确认并准备切换

6.1.2.成功响应

状态码	说明	详情
200	成功	服务器已成功处理请求
201	已创建	请求成功并且服务器创建了新的资源
202	已接收	服务器已接收请求，但尚未处理
203	非授权信息	服务器已成功处理了请求，但返回的信息来自另一个源
204	无内容	服务器成功处理了请求，但没有返回任何内容
205	重置内容	服务器成功处理了请求，内容被重置
206	部分内容	服务器成功处理了部分请求

6.1.3.重定向

状态码	说明	详情
300	多种选择	针对请求，服务器可执行多种操作

301	永久移动	请求的网页已永久移动到新位置，即永久重定向
302	临时移动	请求的网页暂时跳转到其他页面，即暂时重定向
303	查看其他位置	如果原来的请求是POST，重定向目标文档应该通过GET提取
304	未修改	此处请求返回的网页未经修改，继续使用上次的资源
305	使用代理	请求者应该使用代理访问该页面
307	临时重定向	临时从其他位置响应请求的资源

6.1.4.客户端错误

状态码	说明	详情
400	错误请求	服务器无法解析该请求
401	未授权	请求无法解析该请求
403	禁止访问	服务器拒绝此请求
404	未找到	服务器找不到请求的页面
405	方法禁用	服务器禁用了请求中指定的方法
406	不接收	无法使用请求的内容响应请求的页面
407	需要代理授权	请求者需要使用代理授权
408	请求超时	服务器请求超时
409	冲突	服务器在完成请求时发送的冲突
410	已删除	请求的资源已永久删除
411	需要有效长度	服务器不接收不含有效内容长度标头字段请求
412	未满足前提条件	服务器未满足请求者在请求中设置的某一个前提条件
413	请求实体过大	请求实体过大，超出服务器的处理能力
414	请求 URI 过长	请求的网址太长，服务器无法处理
415	不支持类型	请求格式不被请求页面支持
416	请求范围不符	页面无法提供请求范围
417	未满足期望值	服务器未满足期望请求标头字段的要求

6.1.5.服务器错误

状态码	说明	详情
500	服务器内部错误	服务器遇到错误，无法完成请求
501	未实现	服务器不具备完成请求的能力
502	错误网关	服务器作为网关或代理，接收到上游服务器的无效响应
503	服务不可用	服务器目前无法使用
504	网关超时	服务器作为网关或代理，没有及时从上游服务器收到请求
505	HTTP版本不支持	服务器不支持请求中使用 HTTP 协议版本

6.2.响应头

响应头，包含了服务器对请求的应答信息，如 Content-Type、Server、Set-Cookie等。下面简要介绍一些常用的响应头信息

- Date：用于表示响应产生的时间。
- Last-Modified：用于指定资源的最后修改时间。
- Content-Encoding：用于指定响应内容的编码。
- Server：包含服务器的信息，例如名称、版本号...
- Content-Type：文档类型，指定返回的数据是什么类型，也就是响应报文 body 中的数据格式。常见有 text/html 格式为 HTML，text/css 格式为 CSS，application/javascript 格式为 Javascript，application/json 格式为 JSON。
- Set-Cookie：设置 Cookie。响应头中 Set-Cookie 用于告诉浏览器需要将次内容放置 Cookie 中，下次请求将 Cookie 带上。

6.3.响应体

响应体，也能叫正文，响应的数据都存在正文中，例如请求一个网页，正文就是网页的 HTML代码。

HTTP/1.1 200 OK
Bdpagetype: 2
Bdqid: 0x87adf6f3000058c8
Cache-Control: private
Connection: keep-alive
Content-Type: text/html; charset=utf-8
Date: Sun, 18 Dec 2022 11:39:31 GMT
Expires: Sun, 18 Dec 2022 11:39:31 GMT
Isprivate: 1
Server: BWS/1.1
Set-Cookie: BDSVRTM=493; path=/
Set-Cookie: BD_HOME=1; path=/
Set-Cookie: H_PS_PSSID=37857_36549_37552_37683_37908_37796_36804_37926_37852_37902_26350_37789_37881; path=/; domain=.baidu.com
Strict-Transport-Security: max-age=172800
Traceid: 167136357101984355949776741889628723400
X-Frame-Options: sameorigin
X-UA-Compatible: IE=Edge,chrome=1
Content-Length: 469601

<!DOCTYPE html><!--STATUS OK-->

<html><head><meta http-equiv="Content-Type" content="text/html; charset=utf-8"><meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"><meta co
01_1660835115000.png"><title>百度一下， 你就知道</title>
<style index="newi" type="text/css">#form .bdsug{top:39px}.bdsug{display:none;position:absolute;width:535px;background:#fff;border:1px solid #ccc!important;_overflow
g-direct p{color:#00c;font-weight:700;line-height:34px;padding:0 8px;margin-top:0;cursor:pointer;white-space:nowrap;overflow:hidden}#m .bdsug .bdsug-direct p img{wid
height:22px;background-color:#f5f5f5}.bdsug .bdsug-pc-direct-tip-is{position:absolute;right:15px;top:3px;width:55px;height:15px;display:block;background:url(https://pss.l
1d1d1).bdpfmenu{font-size:12px;background-color:#fff}.bdpfmenu a,.usermenu a{display:block;text-align:left;margin:0!important;padding:0 9px;line-height:26px;text-deco
d solid}#wrapper .bdnuarrow em{border-bottom-color:#d8d8d8;top:-1px}#wrapper .bdnuarrow i{border-bottom-color:#fff;top:0}#gxszHead .prefpanelclose{cursor:pointer
t-family:Arial,"PingFang SC","Microsoft YaHei",sans-serif;z-index:1}#head_wrapper.sam_head_wrapper2 #form .bdsug-new{width:545px;z-index:1;border:1px solid #4E6EF2
{background-color:#F5F5F6!important}.wrapper_new #form .sam_search .bdsug-new .bdsug-s{background-color:#F1F3FD!important}#head_wrapper #form .sam_search .b
dsug-new{width:100%;box-shadow:0 4px 4px 0 rgba(0,0,0,.1);border:1px solid rgba(0,0,0,.05)!important;border-radius:12px;top:56px}#head_wrapper #form .sam-bdsug.bd
us:8px;height:32px;width:76px;text-align:center;line-height:32px;font-size:13px;color:#FFF;position:absolute;z-index:2;top:50px}.sam_search .sam_search_rec_hover:before,s

<style type="text/css" index="head">blockquote,body,button,dd,dl,dt,fieldset,form,h1,h2,h3,h4,h5,h6,hr,input,legend,li,ol,p,pre,td,textarea,th,ul{margin:0;padding:0}
html{color:#000;overflow-y:scroll;overflow:-moz-scrollbars}
body,button,input,select,textarea{font-size:12px;font-family:Arial,sans-serif}
h1,h2,h3,h4,h5,h6{font-size:100%}
em{font-style:normal}