一，什么是接口

- USB 是， 硬件接口
- 耳机 3.5mm 是
- 水龙头, 自来水水管 水管， 水
- 灯泡 e27 连接了两个不同的事物，系统，可以进行数据传输。
- api.github.com 服务端，客户端，电视
- UI：user interface 用户接口， 用户界面
- 函数： 公开访问才能是接口
- API： application programming interface , 应用程序可编程接口

接口，两个不同事物之间进行适配的一种工具、规范和协议。

接口的实现。

二，什么是网络请求

- 客户端，前端。主动请求。能够发起对应的请求的客户端。
- 服务端，后端。被动接受。

图：

面试：接口什么协议, 你和谁签了什么协议。 规则

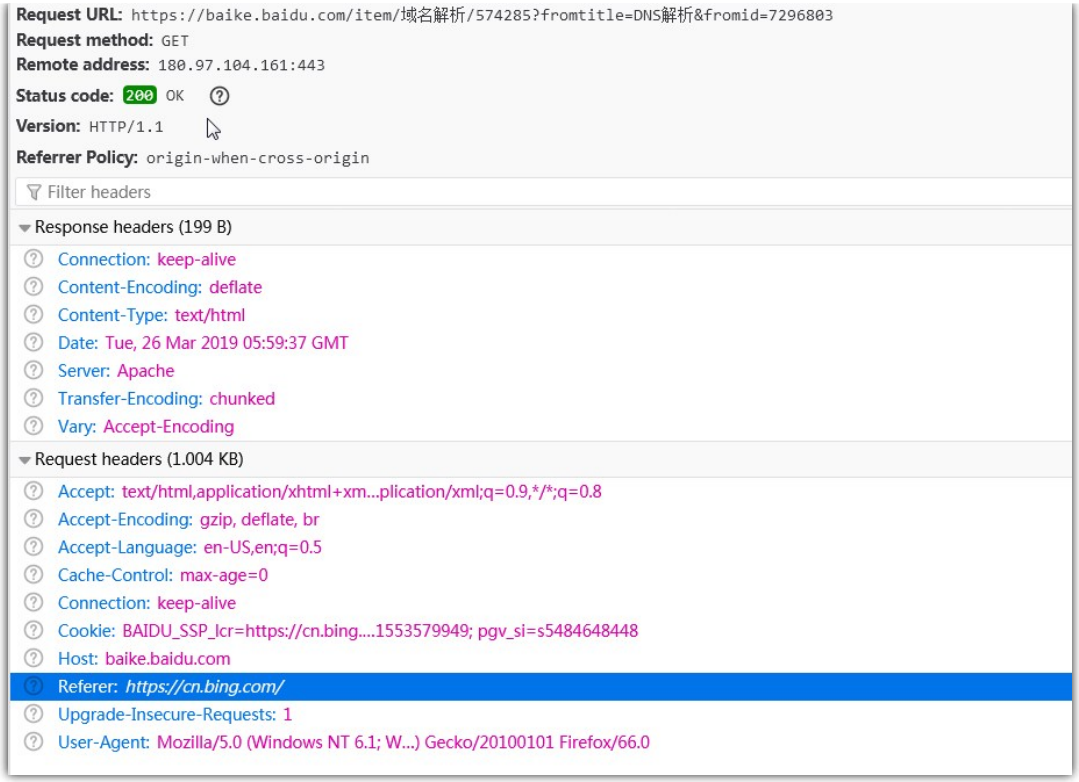
http

websocket

webservice

三，HTTP 请求 协议。

- 请求首行
- 请求头， headers, content-type, 数据格式，爬虫 user-agent: 限制请求的来源。， ip ban
- 请求体



1，请求首行

- url， 域名地址
- 请求方法
  - GET
  - POST
  - PUT
  - DELETE
  - OPTION
  - HEAD
  - 你要寄什么快递？ 圆通，顺丰，京东，联邦，保镖。（玩命快递）
- 远程 IP 地址
- 协议版本

http > tcp / ip 172.123.3.45

(一)， 域名和 IP 地址的区别

- 域名更好记忆；
- 映射 IP 地址。
- DNS 解析： 域名 --》 ip

网络通信就是送包裹，送快递。

- 为什么要有 IP 地址。 邮政系统，送快递。
- 为什么要有端口号。 送到哪个房间。
- 为什么要有域名？ 记不住 IP

访问完域名以后，要进行 DNS 解析，获取地址才能送信啊。

（二），GET 和 POST 的区别

- GET 表示获取资源，POST 表示创建资源；
- GET 没有请求体，POST 有请求体；
- GET 请求参数（query string查询字符串）放在url中以 `?key1=value1&key2=value2` 的形式, POST 不仅可以查询字符串，而且可以放在请求体里面。
- 为什么说 POST 比 GET 更安全?

Body参数方式	Content-type
Text	text/plain
Form	application/x-www-form-urlencoded
JSON	application/json
File	不确定

2，请求头

你包裹上面的说明，你寄的是什么啊，多久到啊，价值多少啊。你这个请求的说明信息。

不需要每个都了解，但是如果某个收包裹的人（服务器）要求你必须带上指定的头信息，你就要带上，不然访问不成功。。。 比如很多退件的，会给你说明，不要到付，到付拒收！！

就算他没有说明，你到付，也不会有人收。。

- user-agent
  - 用户代理
  - 手机，浏览器， postman, jemter
- content-type：请求数据格式
- cookie
  - 会员卡

3，请求体

你的包裹。 包裹可能是空的，你把一些信息直接放到包裹说明上，懒得拆了。 --》 我很好，不用担心。放到里面多麻烦，有风险。太容易被人看到了。

快递：勿忘我

四，响应

我请求到你家里蹭饭，

> 吾兄： 见字如面。自上次别。

> 李云龙版： 老赵啊，上次在你家里喝的那点五粮液，太少啦！你啥时候这么抠啦，那顿饭吃得真tn 的不爽啊。这样，这周六，我老李再到你家去一趟，你记得提前准备 10 斤小龙虾啊。
>
> 苏大强版：明哲，我那房子啥时候给我买啊。我跟你讲，明成和丽丽天天不给我好脸色，我一天都不想呆了，你要再不回来，我要跳江啦。

这信怎么样啊？是收到了啊，还是中途快递给掉了？ 赵政委到底什么态度啊？让不让去啊？有没有小龙虾和五粮液啊。得有个回信不是？

- 响应首行
- 响应头
- 响应体

1，响应首行（状态行）

- 协议版本号
- 响应的状态码。
    - 内部人员规定一种非常专业的信息格式，暗号。
    - 0001 表示 我已收到回件，
    - 0002 已装车，且一切正常。。。
    - 更加简洁。

状态码大类	表示的含义	客户端client要做的事	服务器端server要做的事

状态码大类	表示的含义	客户端client要做的事	服务器端server要做的事
1xx	Informational 信息	啥都不用做，知道就好	告诉client，信息收到了，我后续会处理
2xx	Successful 成功	啥都不用做，知道就好	告诉client，请求已正确处理
3xx	Redirection 重定向	重新请求返回的新地址 -> 才能获取真正需要的数据	告诉client，你需要的内容，由于一些原因，比如地址已发生变化了，然后返回该内容的新地址
4xx	Client Error客户端的错误	确保用正确的参数和信息正确，重新请求	告诉client，请求已正确处理
5xx	Server Error 服务器端的错误	（一般来说）都无需啥操作 -> 往往需要服务器端改了bug后，重新发送请求	需要服务器Server端自己找到具体出了啥错 -> 往往是服务器端的代码的bug导致了出错

（一），最常用的状态码及含义

- Successful - 2xx：成功类，行为被成功地接受、理解和采纳
-
  - 200 OK
  - 服务器成功返回用户请求的数据
    - 往往为了简化处理
    - POST创建成功后应该返回201的，创建
- 404 NOT FOUND
  - 找不到资源
- 500 INTERNAL SERVER ERROR
  - 服务器内部错误
  - 最常见的原因是：服务器内部挂了
  - 比如你传递参数中有些参数是空，而导致后台代码无法解析，出现异常而崩溃

（二），次常用的状态码及含义次常用的响应码及含义

- Successful - 2xx：成功类，行为被成功地接受、理解和采纳
  - 201 CREATED
    - 通过POST或PUT创建资源成功
  - 204 NO CONTENT
    - 资源修改成功，但是没有返回内容
    - 常用于DELETE操作的返回

- Redirection - 3xx：重定向类，为了完成请求，必须进一步执行的动作
    - 301 永久重定向
    - 302 临时重定向
    - 304 Not Modified

- Client Error - 4xx：客户端错误类，请求包含语法错误或者请求无法实现
-
    - 401 UNAUTHORIZED
    -
        - 没有权限访问该资源
        - 典型情况：用户没有登录，没有获得对应的access token而直接访问某资源
    - 403 FORBIDDEN
        - 禁止访问
        - 典型情况：虽然用户已登录，但是去更新/删除需要更高权限才能操作的资源
    - 405 METHOD NOT ALLOWED
        - 方法不允许
        - 举例：比如某个资源不允许POST请求，但是你确发起了POST请求

2， 响应头

- content-type， 返回数据的格式
- set-cookie,

3, 响应体

- 返回的数据
- json, text， html

五，cookie， session 和 token

- cookie
    - http 是无状态的，没有记性。
    - cookie 是让服务器记住你。发会员卡，给你推荐
    - 存储在浏览器
- session。
    - 在服务器记住用户信息状态的。验证
    - 服务器验证
- token， 令牌。 跨平台。 只要他有这个令牌，不管他是什么身份，手机，浏览器，电视，
    - 保存在客户端本地， local_storage
    - 移动端流通。手机，平板，web, 第三方客户端
    - 口令
    - 口令是会变的。

六，输入 url 后的过程

a.域名解析，DNS 解析 -》 ip 地址

b.发起TCP连接的三次握手，建立连接。

c.建立TCP连接后发起http请求

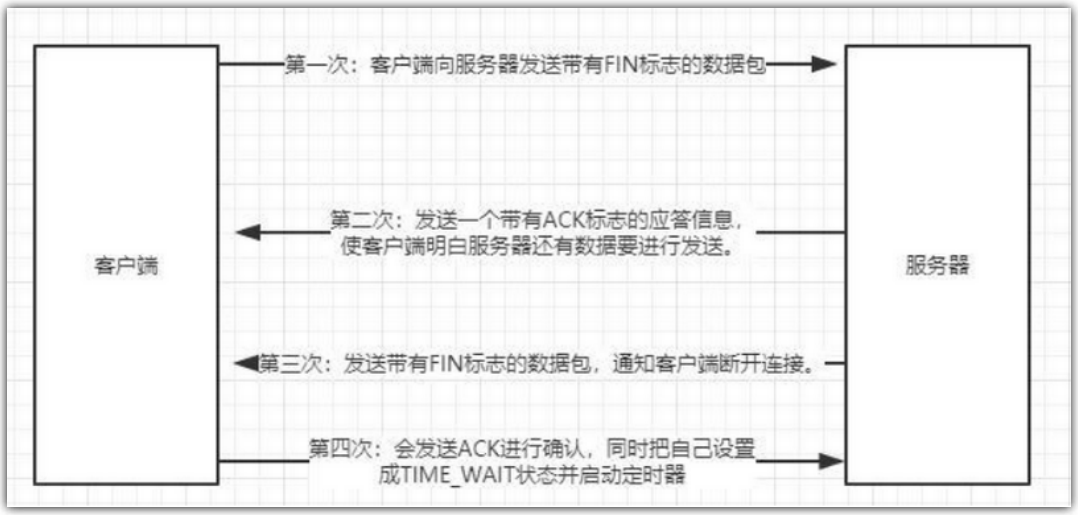
d.服务端响应http请求，返回响应报文

e.浏览器页面渲染，展示。

f.断开TCP连接，四次挥手

七，三次握手四次挥手

- 第一次握手：建立连接时，客户端向服务端发送请求报文（SYN），“我想建立连接”
- 第二次握手：服务器收到请求报文后，如同意连接，则向客户端发送确认报文（SYN/ACK）“同意建立”
- 第三次握手：客户端收到服务器的确认后，再次向服务器发送确认报文，完成连接（ACK）
- 三次握手主要是为了防止已经失效的请求报文字段发送给服务器，浪费资源。



- 第一次挥手：客户端想分手，发送消息（FIN）给服务器
- 第二次挥手：服务器通知客户端已经接受的挥手请求，返回确认消息（ACK），但还没做好分手准备；
- 第三次挥手：服务器已经做好分手准备，通知客户端（FIN）
- 第四次挥手：客户端发送消息给服务器（ACK），确认分手，服务器关闭连接。