

# 前后端联调怎么做

@被删

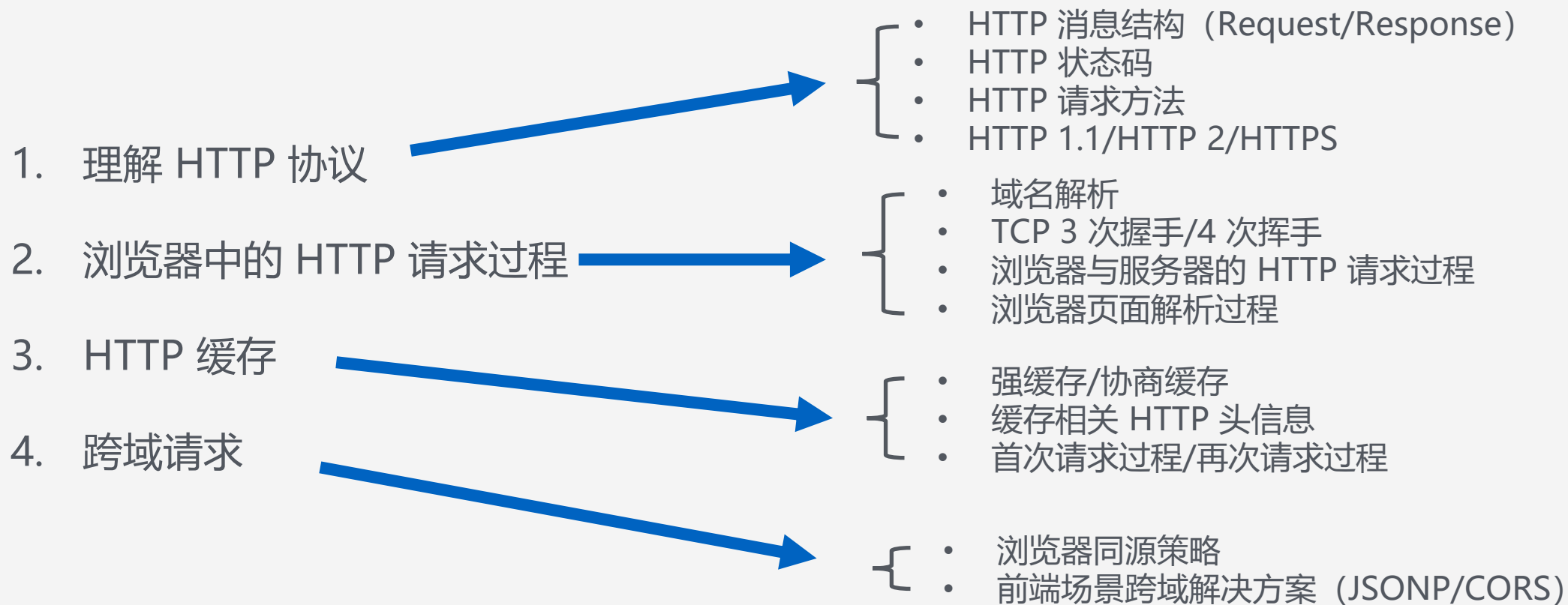
1

# 前后端联调必备知识



# 关于 HTTP 请求

网络请求，是用来从服务端获取需要的信息，然后解析协议和内容，来进行页面渲染或者是信息获取的过程。



虽然这些知识很难吸收，但的确在开发过程中会遇到，还是要掌握的

# 前后端联调的一些术语—登录态

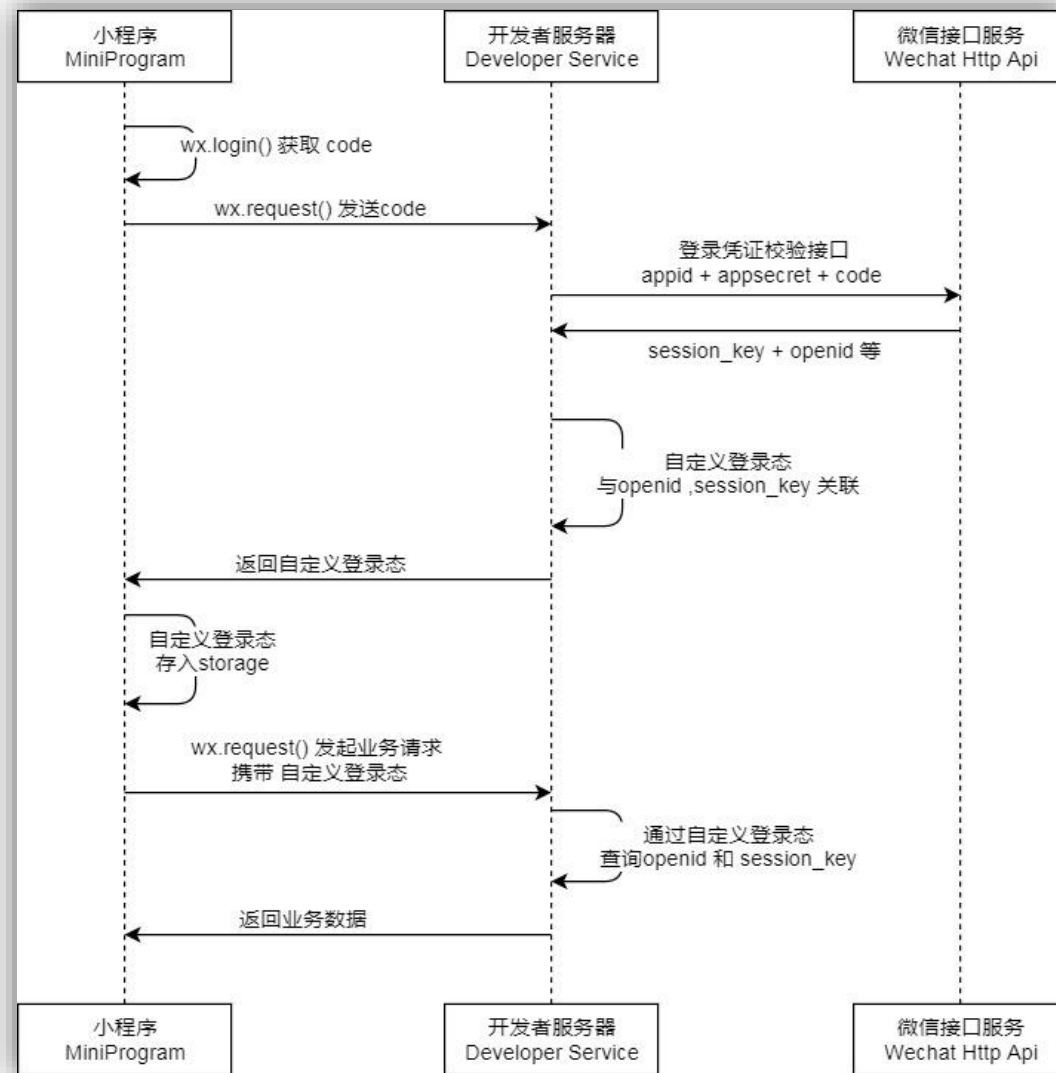
- 需要了解 Cookie/Token/Session 等概念
- 常见的登录态管理过程

## 1. 密码登录

- 前端账号密码提交到后台（加密）
- 后台注入 Cookie/生成 Token
- 前端请求浏览器自动带上 Cookie
- 前端存储 Token
- 每次请求中带上 Token

## 2. 扫码登录

- 前端轮询后台，后台查询第三方平台
- 用户扫码登录，后台获取用户信息
- 后台生成 Token/Cookie，给到前端

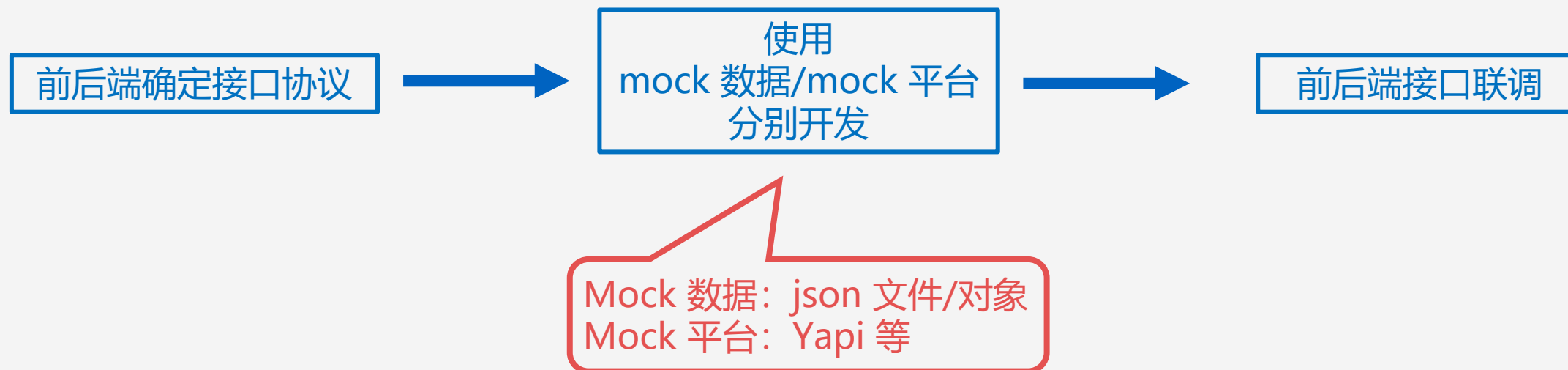


小程序登录过程

# 前后端联调的一些术语—接口定义

- 需要了解常见 HTTP 错误码 (1xx/2xx/3xx/4xx/5xx)
- 前后端自定义接口格式/错误码
- RESTful API/pb 协议

前后端联调过程：



# 前后端联调的一些术语—代理配置

考虑到前端本地开发、跨域的情况，我们常常需要配置代理，使得前端的请求能顺利地发送到后台的服务器上

- 了解前端本地起服务 (localhost/127.0.0.1)
- 了解 HOST 配置
- IP 代理/域名代理

**[ 前端常用代理配置工具: whistle ]**

1. 查看/修改HTTP、HTTPS、HTTP2、WebSocket、TCP 请求响应数据
2. 设置 hosts (支持 IPv6) 、http-proxy、https-proxy、socks
3. 作为 HTTP 代理或反向代理
4. 集成常用的 web 调试工具，如 weinre 和 log 等
5. 支持用 Node 编写插件扩展

2

# 前后端联调技巧



# 前端常用 HTTP 请求相关 API

Ajax 请求: Asynchronous JavaScript and XML,  
意思是用 JavaScript 执行异步网络请求。

异步请求的核心: XMLHttpRequest API

- 统一接口参数处理
- Promise 化
- 统一的异常处理 (请求失败/错误码处理)
- 自动重试请求

前端常用请求库: Axios

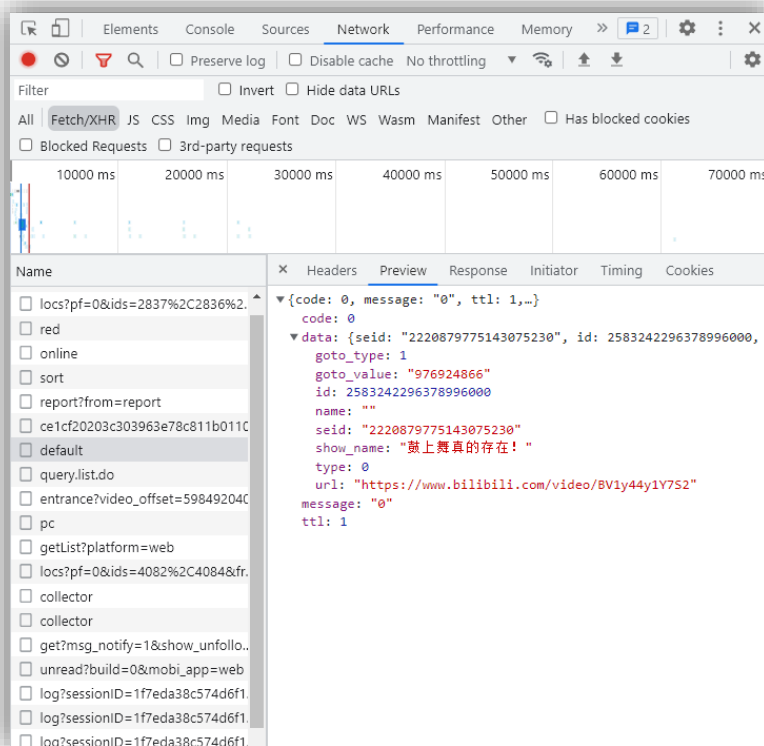
```
var request = new XMLHttpRequest(); // 新建XMLHttpRequest对象

request.onreadystatechange = function() {
  // 状态发生变化时, 函数被回调
  if (request.readyState == 4) {
    // 成功完成
    // 判断响应结果:
    if (request.status == 200) {
      // 成功, 通过responseText拿到响应的文本
      console.log(request.responseText);
    } else {
      // 失败, 根据响应码判断失败原因:
      console.log(request.status);
    }
  }
};

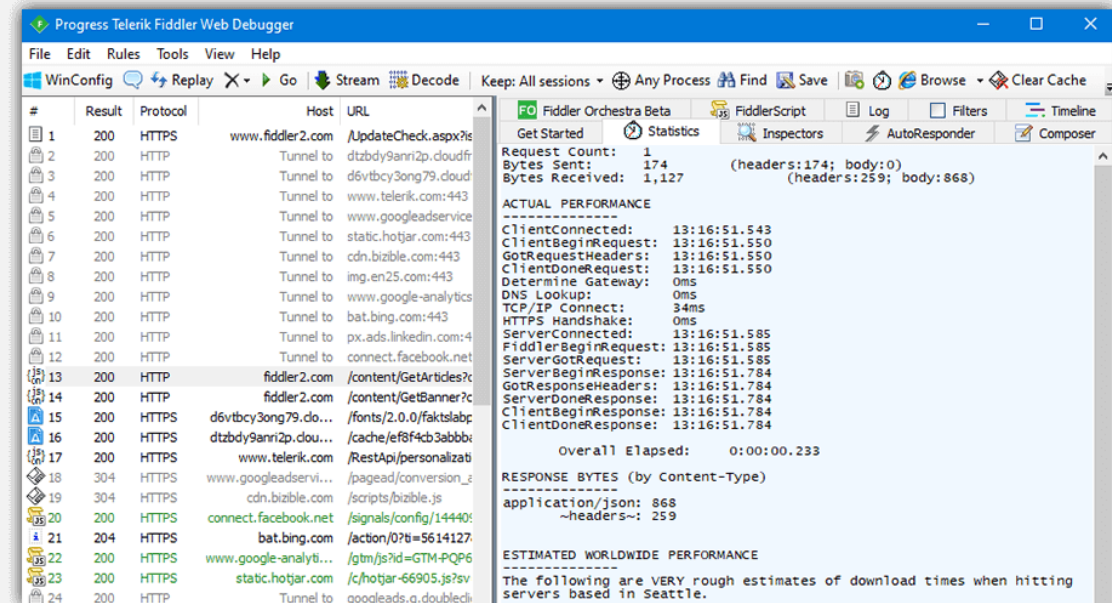
// 发送请求
// open的参数:
// 一: 请求方法, 包括get/post等
// 二: 请求地址
// 三: 表示是否异步请求, 若为false则是同步请求, 会阻塞进程
request.open("GET", "/api/categories", true);
request.send();
```



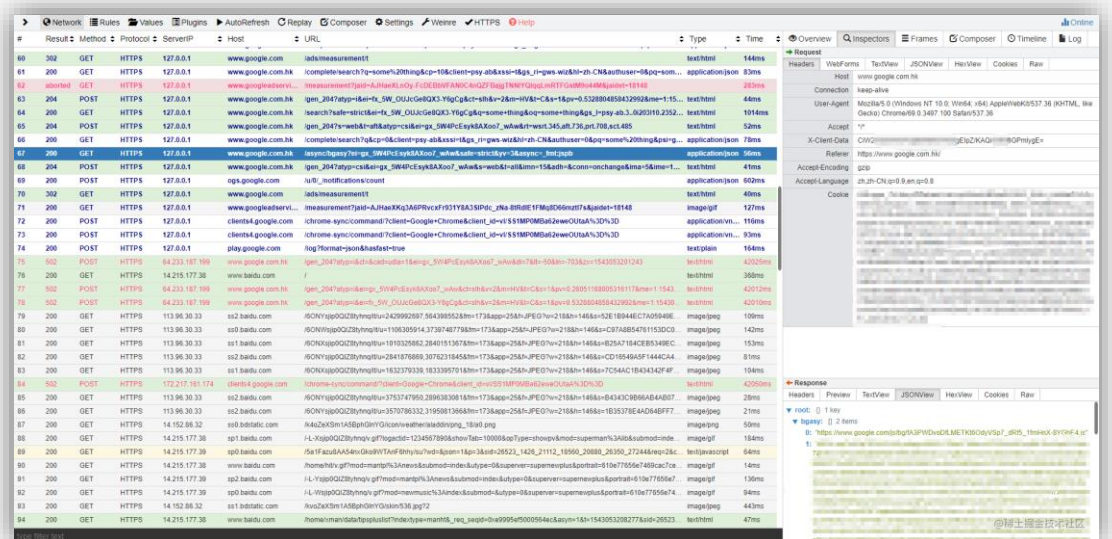
# 常用开发调试技巧



## 浏览器查看请求：Network 面板



## Fiddler: 抓包/重放/修改请求/响应内容



## whistle: 代理/抓包/重放/修改请求/响应内容

# 如何自行学习和调试

找一些开放平台提供的 API（百度地图/高德地图/小程序云开发），进行请求的调试和处理

Web服务API

概述

获取密钥

地点检索

地点输入提示

正/逆地理编码

地理编码

全球逆地理编码

轻量级轨迹服务

道路信息预警服务

路线规划

智能调度

私有化图层

天气查询

批量算路

服务介绍

服务文档

使用指南

常见问题

更新日志

资源下载

接口功能介绍

地理编码

https://api.map.baidu.com/geocoding/v3/?address=北京市海淀区上地十街10号&output=json&ak=您的ak&callback=s  
howLocation //GET请求

注意：当前为V3.0版本接口文档，V2.0及以前版本自2019.6.18起新用户无法使用。老用户仍可继续使用V2.0及以前版本请求实现逆地理编码服务，为保障用户体验，建议您尽快迁移到V3.0版本。

请求参数

参数名	参数含义	类型	举例	默认值	是否必须
address	待解析的地址。最多支持84个字节。 可以输入两种样式的值，分别是： 1、标准的结构化地址信息，如北京市海淀区上地十街十号【推荐，地址结构越完整，解析精度越高】 2、支持“*路与*路交叉口”描述方式，如北一环路和阜阳路的交叉路口  第二种方式并不总是有返回结果，只有当地址库中存在该地址描述时才有返回。	string	北京市海淀区上地十街10号	无	是



谢谢



Github: godbasin  
@被删