微信网页开发之 JSSDK 最全开发排坑方案

作者：YYJeffrey
原文链接：https://ld246.com/article/1599892525929
来源网站：链滴

很多朋友在微信开发时会遇到各种奇妙的问题，但限于官方文档和网上的一些解决方案都没有很好的总，所以这篇文章献给那些准备使用或正在使用微信JSSDK开发的朋友。

前期准备

先看看需要准备哪些东西，下面这几项都是开发过程中所需要用到的，但不需要急着去安装，先做个解。

- JSSDK：微信开发必不可少的东西
- 公众号：订阅号、服务号、企业号或个人号（个人号权限可能部分受限）
- NATAPP：一个用于内网穿透的工具
- V-Console：移动端Console调试工具
- Nginx（可选）：反向代理服务器用于部署

几个重要链接

- 官方文档：https://developers.weixin.qq.com/doc/offiaccount/OA_Web_Apps/JS-SDK.html
- 签名校验：https://mp.weixin.qq.com/debug/cgi-bin/sandbox?t=jsapisign
- 鉴权代码：http://demo.open.weixin.qq.com/jssdk/sample.zip

安装JSSDK

为项目安装依赖是最基本的做法。

如果你使用JS那么可以直接调用JS外链：http://res.wx.qq.com/open/js/jweixin-1.6.0.js

如果你使用的npm构建项目依赖：你可以使用npm install weixin-js-sdk或npm install weixin-jsapi

在需要使用的页面引入JSSDK：import wx from 'weixin-js-sdk

正式开发

微信开发之时务必参照微信的官方文档来写，因为微信比较善变，指不定哪天他的接口调用就换一种法了。下面我会以调用相机扫码这个接口为例讲述如何使用JSSDK。

wx.config

JSSDK最迷惑的也是最容易出问题的地方就是使用wx.config注入权限配置的时候，最需要注意的是一个页面或者说URL只需要调用一次即可，所以例如VUE项目一般在mounted时候调用。

```
wx.config({
  debug: true, // 开启调试模式,调用的所有api的返回值会在客户端alert出来，若要查看传入的参数可以在pc端打开，参数信息会通过log打出，仅在pc端时才会打印。
  appId: '', // 必填，公众号的唯一标识
  timestamp: , // 必填，生成签名的时间戳
  nonceStr: '', // 必填，生成签名的随机串
  signature: '',// 必填，签名
  jsApiList: [] // 必填，需要使用的JS接口列表
});
```

分析一下他所需要的参数：debug、jsApiList这两个参数其实是可以在前端写死的，一般开发时debu设为true即可，jsApiList填所需要授权的方法，比如扫码接口，那么久填入scanQRCode；在看其他个参数appId、timestamp、nonceStr、signature这四个参数一般可以由后端返回，appId其实也可写死，但是由于开发和生产可能使用不同的公众号配置，那么这边建议还是由后端返回

接下来可以参照下，扫码接口调用的写法：

```
async mounted() {
  // 调用后端接口获取四个配置参数
  const res = await jsAPI.getSignScan()

  if (res != null) {
    wx.config({
      debug: true,
      appId: res.app_id,
      timestamp: res.timestamp,
      nonceStr: res.nonceStr,
      signature: res.signature,
      jsApiList: ['scanQRCode'],
    })
  }
},
```

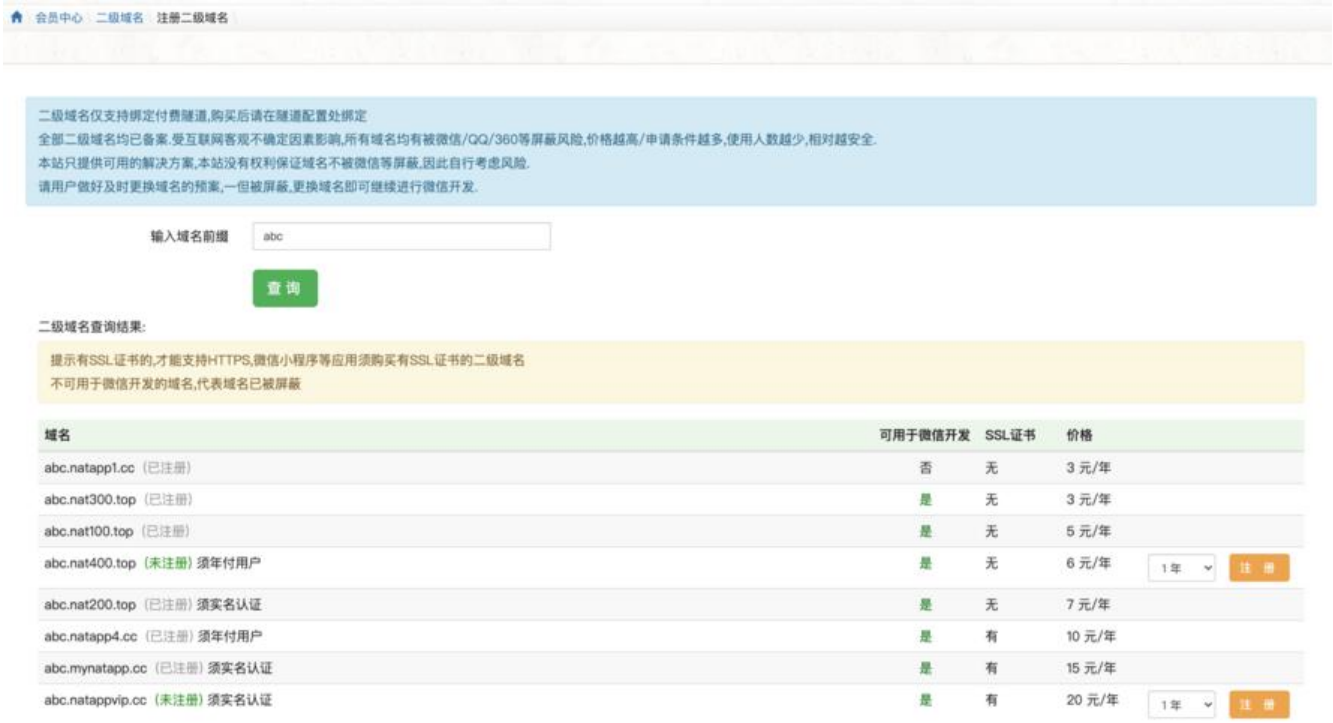
在重要链接中有一个官方的鉴权代码，发给后端看看，他就知道怎么把四个参数传给你了，嘿嘿！ tuck_out_tongue_closed_eyes

NATAPP内网穿透

前端写完wx.config先别急着往下走，这个时候如果是本地开发，没有测试服域名的情况下，需要使一个工具NATAPP，这一个工具可以将内网地址转换成一个公网域名，如果有测试服域名就可以跳过一步，当然除了NATAPP你也可以寻找其他的代替产品例如NAT123等。

下载好对应系统的NATAPP后，就需要开通一条隧道，你可以使用支付宝实名认证获得一条免费的隧做测试，但由于免费的氪金买一条9块钱的隧道，接着做一些简单的隧道配置，名字随意，主要是本地址、本地端口，必须和你前端页面所开的地址和端口一致。

上面有一个二级域名需要配置，如果你不绑定二级域名那么在后续配置公众号时也是没法用的，因为费提供的地址会带一个端口号，但是微信公众号中配置JS安全域名时是不允许带端口号的。选择一个以注册的域名，3块的选一个就可以填入刚才配置隧道的二级域名页面了。



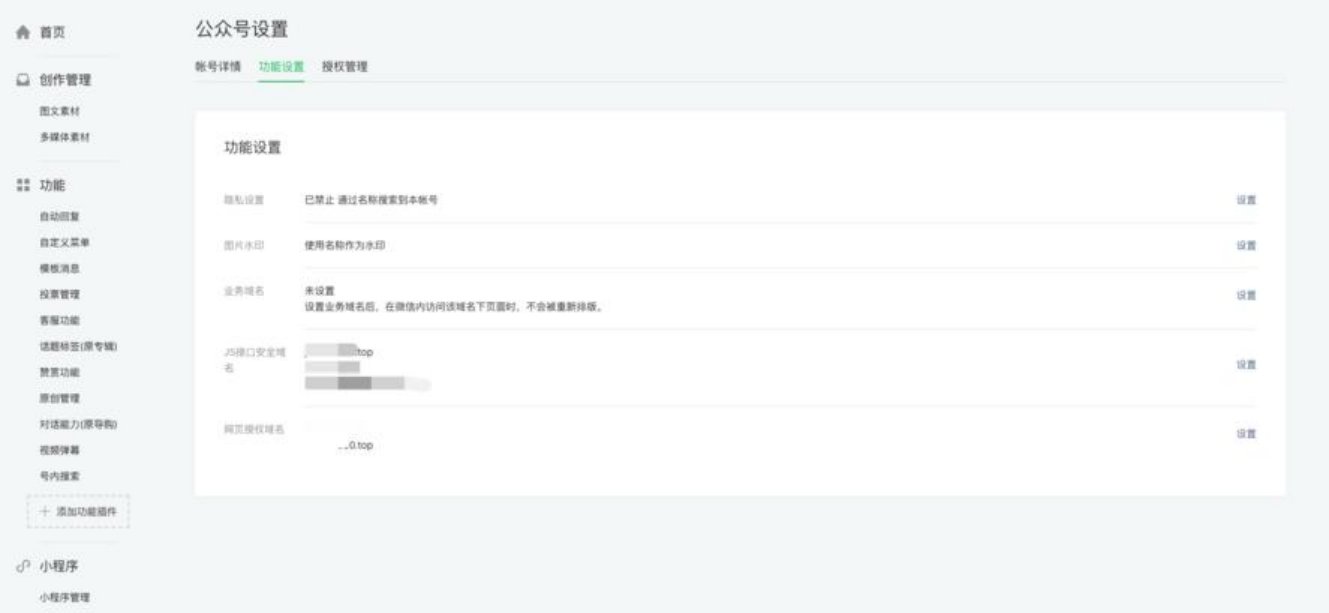
至此你消费了12块钱，找你老板走报销吧，程序员搞钱不容易，半顿外卖没了
ob

接下来需要下载NATAPP提供的配置或者直接复制我下方的即可，authtoken的地方换成你的隧道的authtoken即可，其他都不需要修改，然后将其保存为config.ini配置文件，并发在NATAPP应用的同级录里，启动NATAPP即可使用了。

```
# config.ini
[default]
authtoken=xxx                    #对应一条隧道的authtoken
clienttoken=                     #对应客户端的clienttoken,将会忽略authtoken,若无请空,
log=none                         #log 日志文件,可指定本地文件, none=不做记录,stdout直接屏幕输出 ,默认为none
loglevel=ERROR                   #日志等级 DEBUG, INFO, WARNING, ERROR 默认为EBUG
http_proxy=                      #代理设置 如 http://10.123.10.10:3128 非代理上网用户务必留空
```

配置JS安全域名

既然都走到这里了，那微信配置这块也给大家详细介绍一下，在设置->公众号设置->功能设置->JS口安全域名的地方，将你刚刚设置NATAPP的域名填入到这里。



配置域名时会让你将一个txt文件放到根目录，那么我们把这个txt文件先下载，你如果安装了Flask，你可以构建一个非常小的Web应用如下所示，把路径route路径配置为他所需要的路径，return为txt的内容即可。

```
from flask import Flask
app = Flask(__name__)

@app.route("/MP_verify_eEralKkJErqKwsGo.txt")
def hello():
    return "eEralKkJErqKwsGo"
```

使用Python运行上述代码后，需要将NATAPP后台的配置里将本地端口暂时先修改为5000，因为Flas的端口默认为5000，之后你就可以将你的这个域名成功的配置上去了。

wx.ready() 和 wx.error()

如果你的接口是用户主动触发的，就不需要wx.ready()可以直接跳过，例如扫码接口；但非主动触发行为就必须把你调用的方法写在wx.ready()里，例如获取定位等。

```
wx.ready(function(){
  // config信息验证后会执行ready方法，所有接口调用都必须在config接口获得结果之后，config一个客户端的异步操作，所以如果需要在页面加载时就调用相关接口，则须把相关接口放在ready函中调用来确保正确执行。对于用户触发时才调用的接口，则可以直接调用，不需要放在ready函数中。
});
```

wx.error()也是一个很好判断鉴权是否成功的一个很好办法，wx.config将debug设置为True在开发时非常有必要。

```
wx.error(function(res){
    console.log(res)
});
```

扫码接口

此处调用扫码接口即在你需要触发的地方参考官方文档编写即可，不做过多赘述。

原文链接：微信网页开发之 JSSDK 最全开发排坑方案

```
wx.scanQRCode({
  needResult: 1,
  scanType: ['qrCode', 'barCode'],
  success(res) {
    const result = res.resultStr
    console.log(result)
  },
})
```

真机调试

配置和编码技术后，调试阶段就需要用到刚刚配置的域名打开页面，因为做了穿透所以访问域名和访本地地址能够达到同样的效果，需要注意的时微信开发当然是需要在微信浏览器里打开的，如果你想手机看到控制台输出的信息，你可以安装一下V-Console，npm install vconsole

```
import Vconsole from 'vconsole'
const vConsole = new Vconsole()
```

Invalid Host header

如果打开域名，页面内容没有出现，而是出现了“Invalid Host header”这样的提示，那么这是因为开发模式下，会校验HOST，此时只需要把校验域名关闭即可，VUE项目可以通过修改vue.config.js实现。

```
devServer: {
    disableHostCheck: true,
},
```

域名没问题了，访问很慢？这是因为开发模式下没有进行打包，需要加载的资源很多，此时还有一种法，使用Nginx并把项目打包上去，这里有一个麻烦的地方就是需要调试时都需要npm run build，至你可以run watch监听文件变化并自动打包，这样啥也不用做了，当然build项目你不需要拷贝目录你可以将Nginx的配置指向你打包的绝对路径即可。

```
# nginx.conf
worker_processes 1;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    charset utf-8;

    server {
        listen 8080;
        root /home/workspace/project/dist;  # 项目的打包目录
        index index.html;
    }
}
```