

FOG_V2 MiCO 总动员 APP 说明

摘要 (Abstract)

本文档对 MiCO 总动员设备端和 APP 端是使用作出整体说明，给接入 FOG2.0 平台的的用户作一个参考演示。文档中不涉及代码的讲解，让开发者在不了解代码内容的情况下将 MiCOKIT 和 APP 端通过 FOG2.0 平台连通。

适用读者 (Suitable Readers)

本文档适于文档工作管理者及开发者参考使用。

获取更多帮助 (More Help)

登录上海庆科官方网站: <<http://mxchip.com/>>, 获取公司最新产品信息。

登录 MiCO 开发者论坛: <<http://mico.io/>>, 获取更多 MiCO 最新开发资料。

登录 FogCloud 开发者中心: <<http://easylink.io/>>, 获取更多 FogCloud 云开发文档。

微信“扫一扫”关注: “MiCO 总动员”公众号, 获取 MiCO 团队小伙伴最新活动信息。



版权声明 (Copyright Notice)

Copyright (c) 2016 MDWG Trust and the persons identified as the document authors. All rights reserved.

版本记录

日期	版本	作者	更新内容
2016-10-03	V1.0	G.BWang	初版

Mxchip
reprint prohibited

目录

FOG_V2 MiCO 总动员 APP 说明.....	1
版本记录.....	1
1. FOG 平台说明.....	3
1.1 设备端注意事项.....	3
1.1.1 产品注册.....	3
1.1.2 设备管理.....	4
1.2 APP 端注意事项.....	4
1.2.1 应用注册.....	5
2. 设备端说明.....	6
2.1 打开工程.....	6
2.2 更换 PRODUCT ID.....	6
2.3 新固件烧录.....	6
2.4 串口 LOG 信息查看.....	8
3. APP 端说明.....	10
3.1 APICLOUD 环境搭建.....	10
3.2 创建一个空的 APICLOUD 项目.....	10
3.3 COPY 源码到你的工程中.....	13
3.4 代码提交.....	14
3.5 在你的项目中添加组件.....	15
3.6 云端编译.....	18
3.7 扫码下载 APP.....	19
3.8 APP 的使用.....	19
4. 服务与支持.....	20

1. FOG 平台说明

开发者首先登陆 [FOG CLOUD 平台](#)注册账号然后进入控制台页面。后续步骤会对云端的平台的使用做简要说明。云平台的建议与意见请到 MICO 论坛的 [FOG CLOUD 版块](#)发帖提交。



设备端和 APP 端都有数据的收发操作，设备端和 APP 的上传数据使用 HTTPS 协议，设备端和 APP 的接收数据使用的是 MQTT 和 TLS 协议。

1.1 设备端注意事项

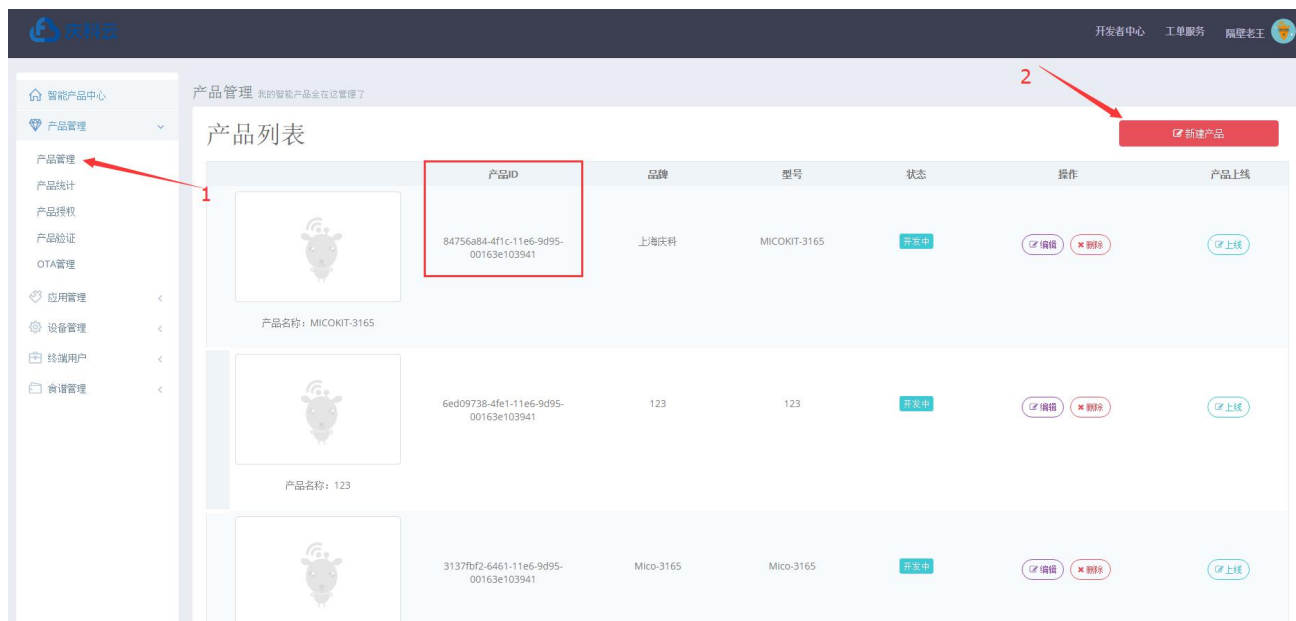
嵌入式端的每一个模块都是一个属于特定产品下的设备。所以设备端的开发固件中必然会有两个非常重要的信息:产品 ID 和设备 ID。产品 ID 是在 FOG CLOUD 平台上由开发者手动创建得到的，设备 ID 是设备端去云端发起 HTTPS 请求注册时得到(注册时设备携带自身的 MAC 地址)。

在云端，产品 ID 被称为 Product ID,设备 ID 被称为 Device ID。一个设备同时拥有 Product ID 和 Device ID 两种属性，来表明自身属于哪类产品下的具体哪个设备。

1.1.1 产品注册

产品注册的步骤就是得到云端后台得到一个唯一的 Product ID。一个开发者账号，可以在云端注册多个 Product ID。

请开发者点击左侧产品管理的进行产品的创建，按照引导的步骤得到 Product ID，记录下来，后面在嵌入式端固件的说明中会用到。



1.1.2 设备管理

在开发者完成产品注册之后，设备端固件写入了新的 Product ID，如果设备端正常运行，则可以在设备管理界面看到有新的设备条目。在条目中开发者可以看到设备的 DEVICE ID、MAC 地址等重要信息。



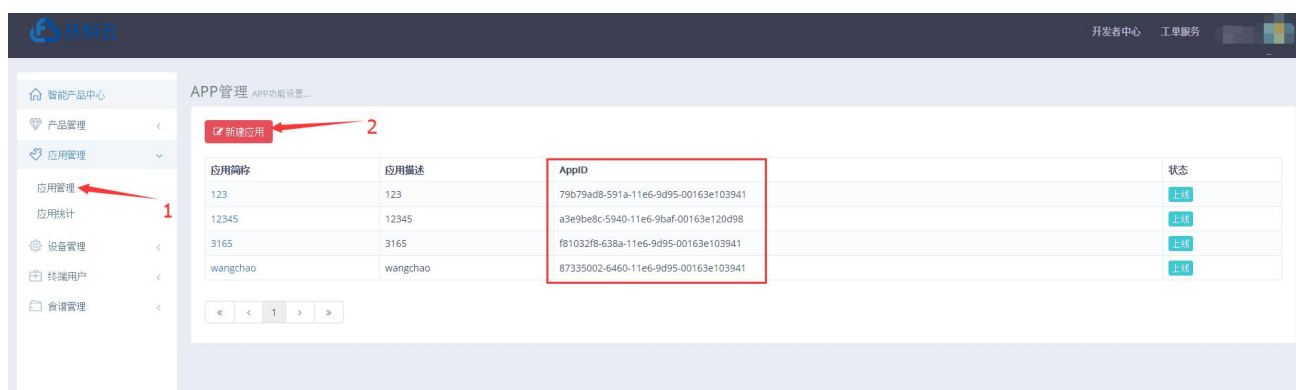
1.2 APP 端注意事项

与设备端类似，每一个手机 APP 都有一个 APP ID 和 CLIENT ID。

APP ID 是在云端的应用创建得到的，每一个手机 APP 的账户在通过手机号注册时得到 CLIENT ID。

CLIENT ID 是一个手机 APP 的账户的唯一标识。

1.2.1 应用注册



在控制台点击应用管理，开发者可以根据引导步骤新建应用。得到的 APPID 后续会在 APP 源码说明中用到。

2. 设备端说明

设备端硬件：MICOKIT-3031、MiCOKIT-3165、MiCOKIT-3166、MiCOKIT-3239、NUCLEO-F411

嵌入式端 FOG 库版本：FOG2.4.0

MQTT 组件库版本：0.1.8

IED：MiCoder

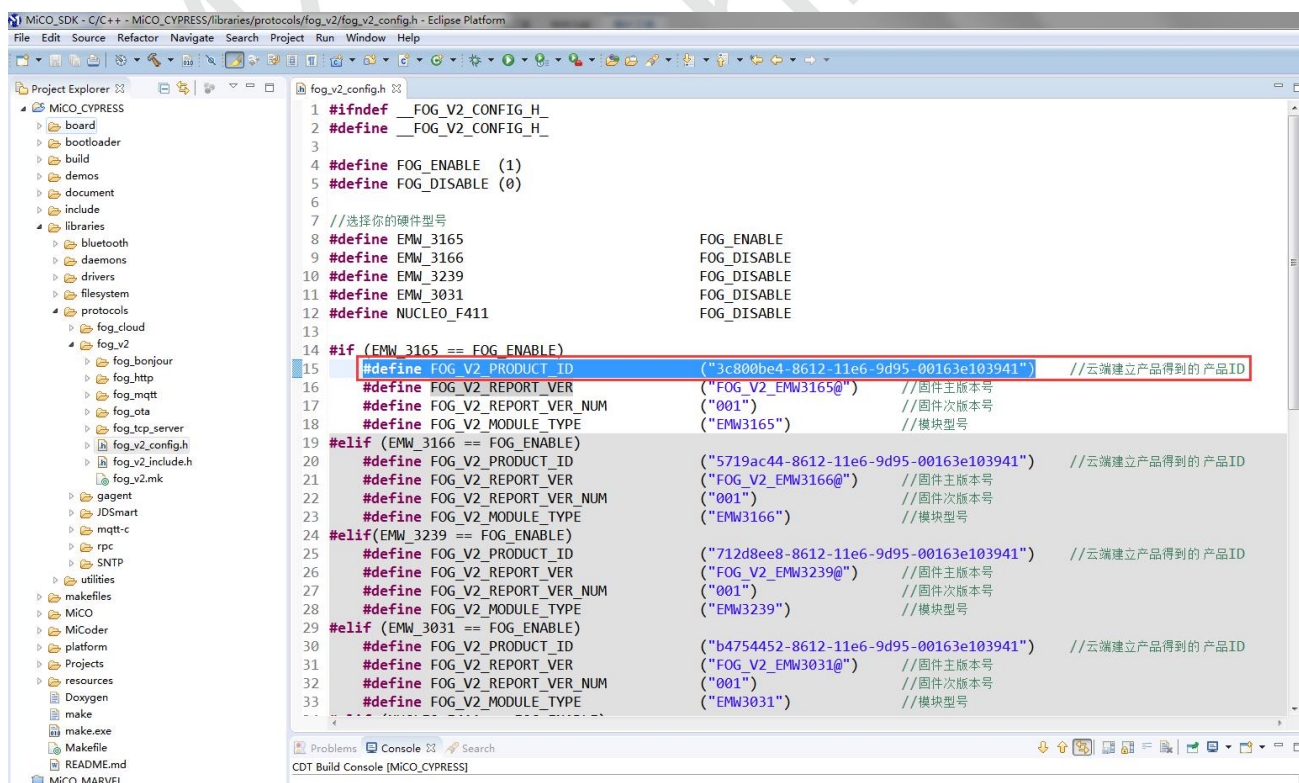
依赖 MiCOSDK：MiCO SDK 3.1

2.1 打开工程

参考视频教程

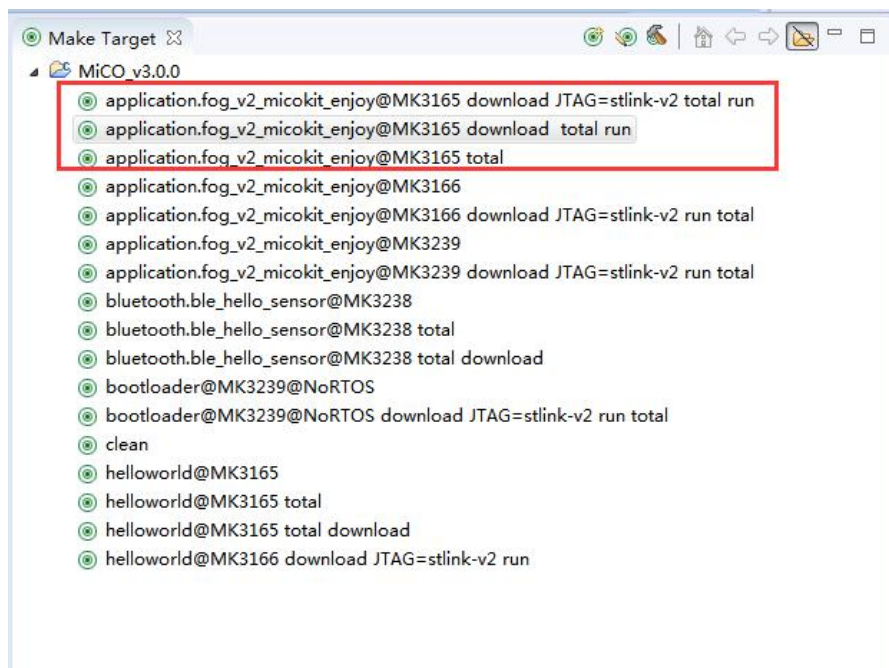
2.2 更换 PRODUCT ID

开发者拿到设备端 SDK 的第一件事情，就是需要把在云端产品管理中注册得到的 Product ID 填写到固件中。Product ID 位置在 fog_v2_config.h 中的 FOG_V2_PRODUCT_ID 定义。



2.3 新固件烧录

如果是 MiCOKIT 则在 MiCoder 中 Make Target 设置如下图所示：



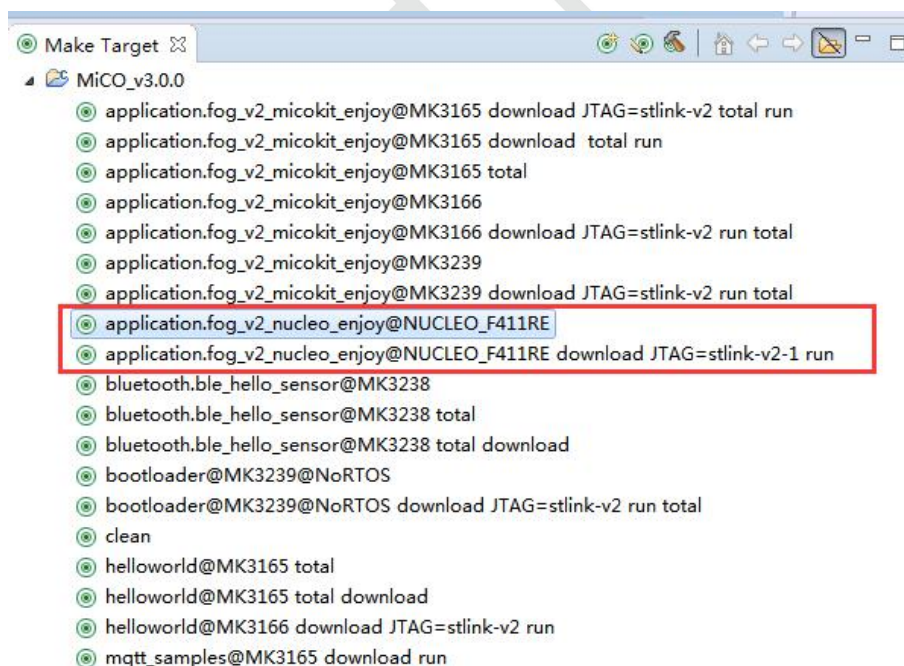
download 表示需要下载固件

JTAG=stlink-v2 表示指定下载方式(如果是开发者的调试器是 JLINK, 参考下面说明)

total 表示同时下载 bootloader 和 RF 射频驱动 (可选项)

run 表示下载完成后让芯片复位一次

如果是开发者拿到的是 NUCLEO-F411, 则设置如下图:



NUCLEO 的底板 STlink 下载器版本为 stlink-v2-1.

说明：MiCOKIT 用户如果使用的是 JLINK,则不需要指定 JTAG=xxxx。但是需要更改 JLINK 驱动，参考说明文档 RM1044CN_MiCoder_ToolChain_and_IDE.pdf 中 MiCoder Windows 版本对 Jlink 仿真调试器的特殊设置章节。

1.3 MiCoder Windows 版本对 Jlink 仿真调试器的特殊设置

在 Windows 系统中使用 Jlink 在 MiCoder 下进行仿真调试时，需要更新 Jlink 的驱动程序，而在 Linux 和 macOS 中则无需修改。

1. 首先，至 MiCoder IDE 的安装路径下，找到驱动 exe 文件，如图 1.3。

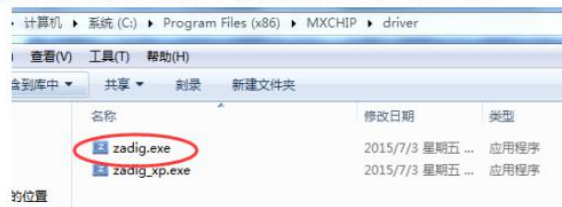


图 1.3 J-Link 驱动路径

2. 双击 exe 文件，进入安装界面，选择 Options->list all devices，示意如图 1.4。

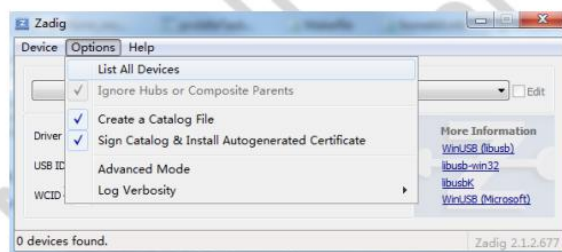


图 1.4 列出所有设备

3. 然后下拉列表选择 J-Link，Driver 选型中选择 libusb-win32 (v1.2.6.0)。然后点击 “Replace Driver”，即可开始更新。



2.4 串口 LOG 信息查看

串口工具我们推荐使用 SecureCRT，，电脑和 MiCOKIT 连接完成之后，在 PC 端设备管理器的端口中查找 COM 口。如果 COM 显示失败可能是因为 FT232 的驱动未安装。

FT232 驱动下载地址：<http://www.ftdichip.com/Drivers/VCP.html>

底板 USB 串口在 application 模式下的波特率 115200，模式为 8-N-1。

注意 RTS/CTS 不要打钩选中，否则 CLI 交互的输入是无效的。



NUCLEO-F411 模块重启后 log 信息显示如下:

```
[0][APP: hello_fog.c: 36] APP Version:FOG_V2_NUCLEO_F411@001
[495][Platform: mico_platform_common.c: 98] Platform initialised, build by GNUC
[987][RTOS: mico_rtos_common.c: 82] Started FreeRTOS v7.1.0
[1591][SYSTEM: system_misc.c: 222] Free memory 64072 bytes
[1597][SYSTEM: system_misc.c: 228] Kernel version: 31621002.050
[1603][SYSTEM: system_misc.c: 231] MiCO version: 3.0.0
[1608][SYSTEM: system_misc.c: 233] Wi-Fi driver version w10: Nov 7 2014 16:03:45 version 5.90.230.12 FWI, mac C8:93:46:80:28:A9
[1619][SYSTEM: mico_system_init.c: 61] Empty configuration. Starting configuration mode...
[1629][SYSTEM: system_easylink.c: 262] Start easylink combo mode
```

同时, 这个串口支持 CLI 交互, 用户可以输入 help 查看有哪些命令可以使用

```
help

====Build-in Commands====
help:
version:
echo:
exit: CLI exit
scan: scan ap
wifistate: Show wifi state
wifidebug: wifidebug on/off
ifconfig: Show IP address
arp: arp show/clean
ping: ping <ip>
dns: show/clean/<domain>
sockshow: Show all sockets
tasklist: list all thread name status
memshow: print memory information
memdump: <addr> <length>
memset: <addr> <value 1> [<value 2> ... <value n>]
memp: print memmp list
wifidriver: show wifi driver status
reboot: reboot MiCO system
tftp: tftp
time: system time
ota: system ota
flash: Flash memory map
micodebug: micodebug on/off

====User Commands====
```

对于 SecureCRT 的使用, 开发者可以网上查阅资料。

3. APP 端说明

MiCOKIT 总动员 APP 是基于 APICLOUD 平台开发的。

APICLOUD 官网地址: <http://www.apicloud.com/>

无论是跨平台的 APICLOUD 还是原生的 Android 和 IOS, 都需要庆科的 APP 组件模块。

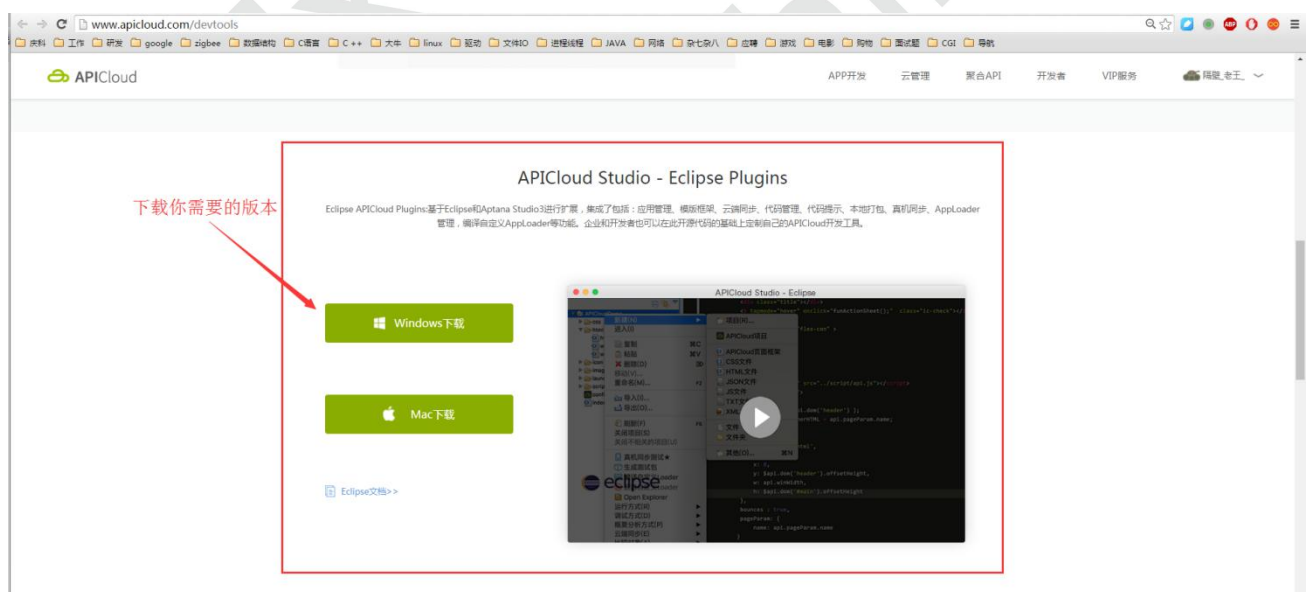
在 APICLOUD 平台上, IOS 和 Android 的组件包的 git 地址为:

<https://github.com/mxchipSDK/Fog2.0/tree/master/APICloud#login>

鉴于 IOS 和 Android 一直在保持更新, 开放出去的 SDK 中会有一个文件夹专门存放适配当前应用层代码稳定的 Android 和 IOS 组件包。同时建议开发者优先使用 SDK 里面的组件包。

3.1 APICLOUD 环境搭建

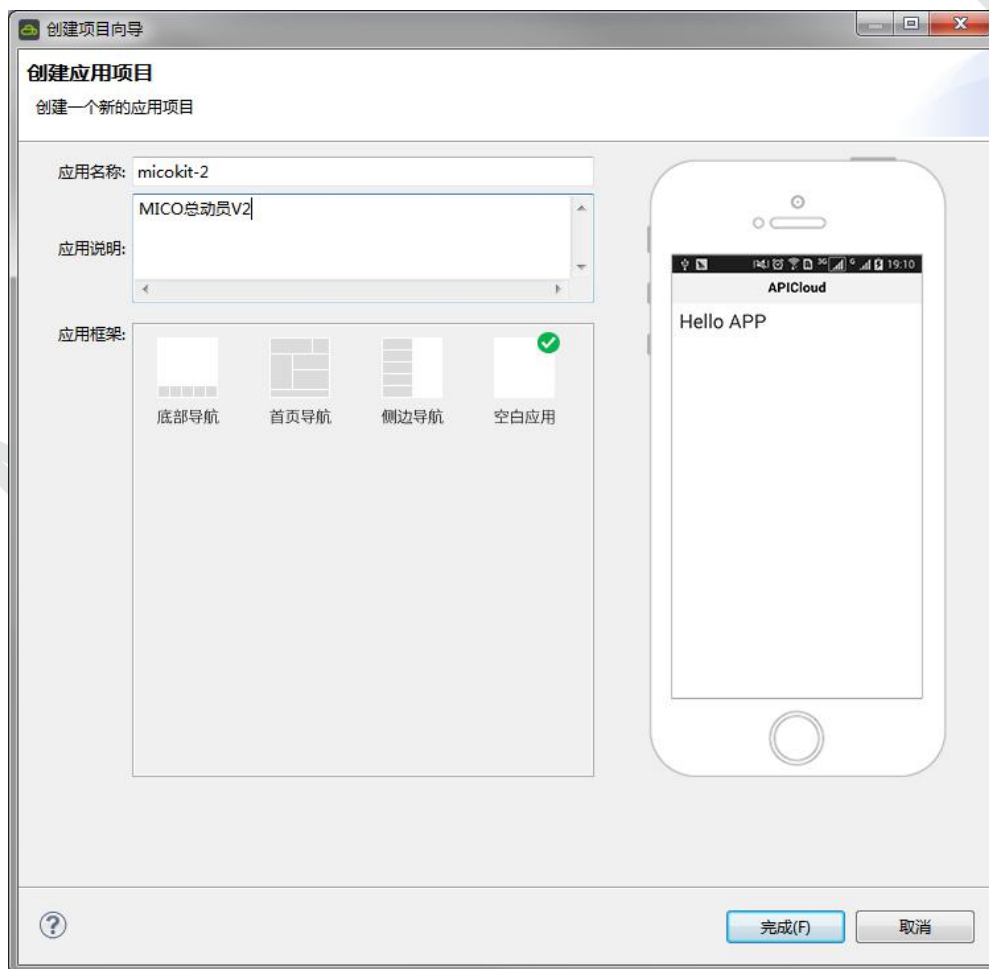
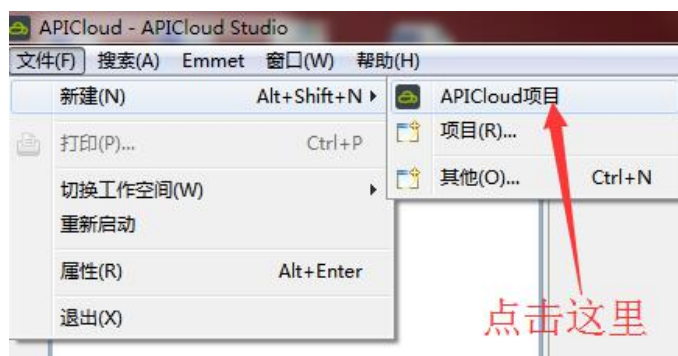
(1) 下载 APICLOUD 的最新 IDE, 下载地址: <http://www.apicloud.com/devtools>

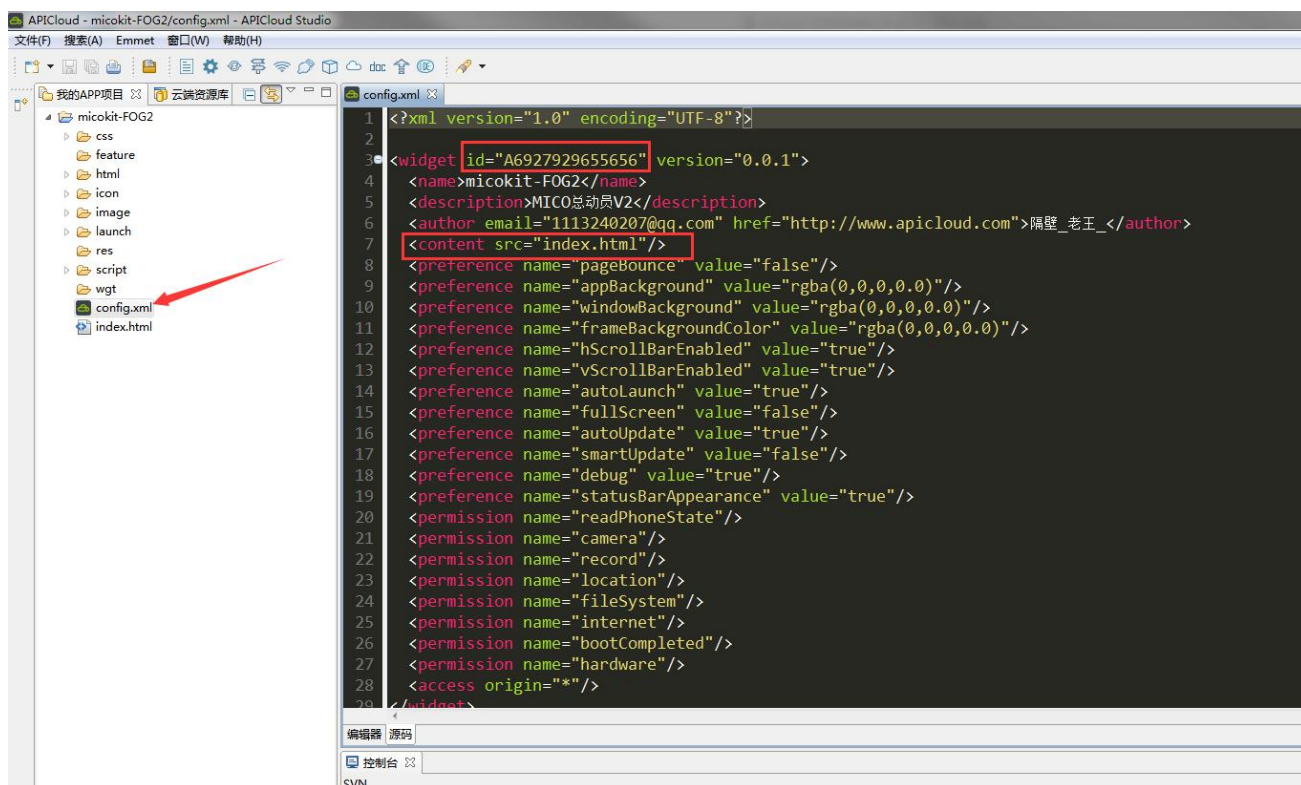


(2) 在 APICloud 平台注册一个账号

3.2 创建一个空的 APICloud 项目

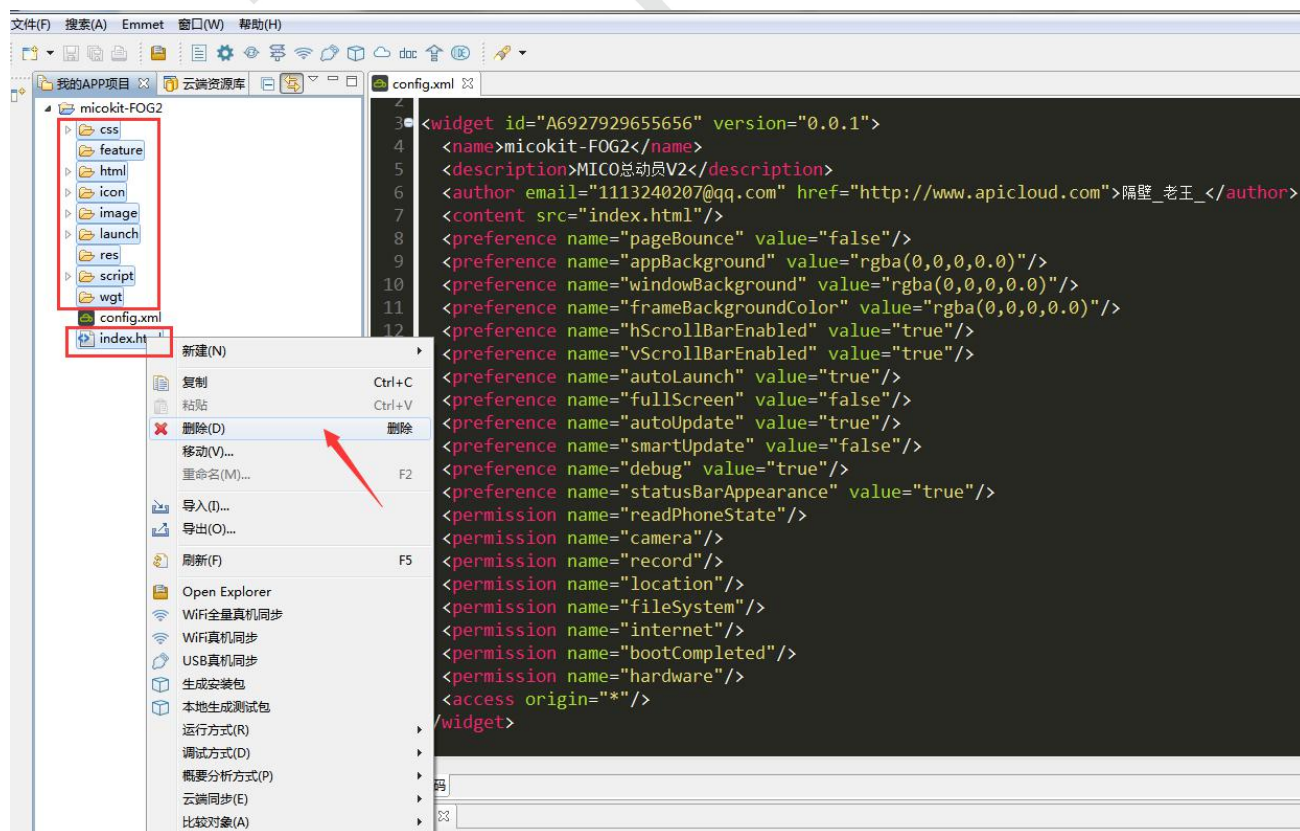
(1) 打开 APICloud 的 Eclipse IDE 之后, 文件->新建->APICloud 项目中新建一个项目。





(2) 在新建的工程中，`config.xml` 比较重要，里面的 `id` 和 `src` 开发者需要留意，后面的步骤会用到。

(3) 删除当前工程下面除了 `config.xml` 文件外的其他所有内容

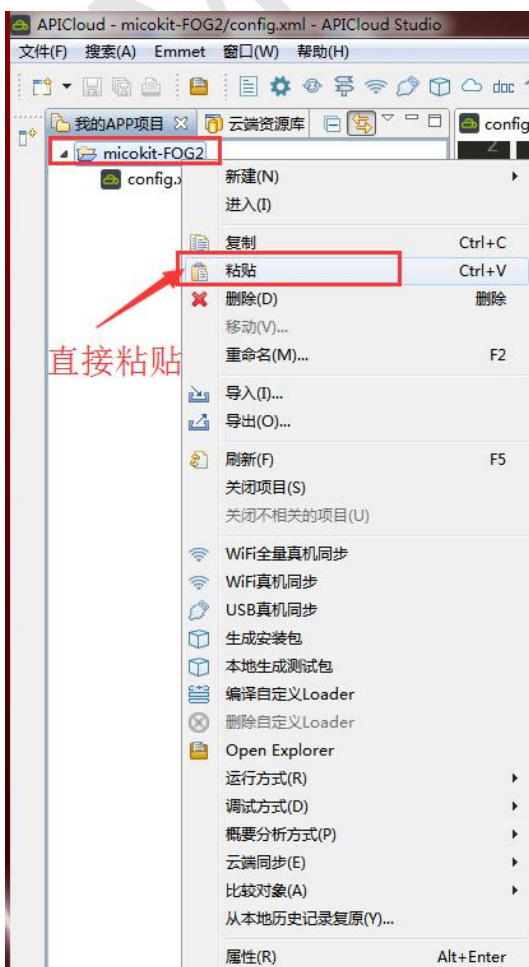


3.3 COPY 源码到你的工程中

(1)复制 APP 端的源码



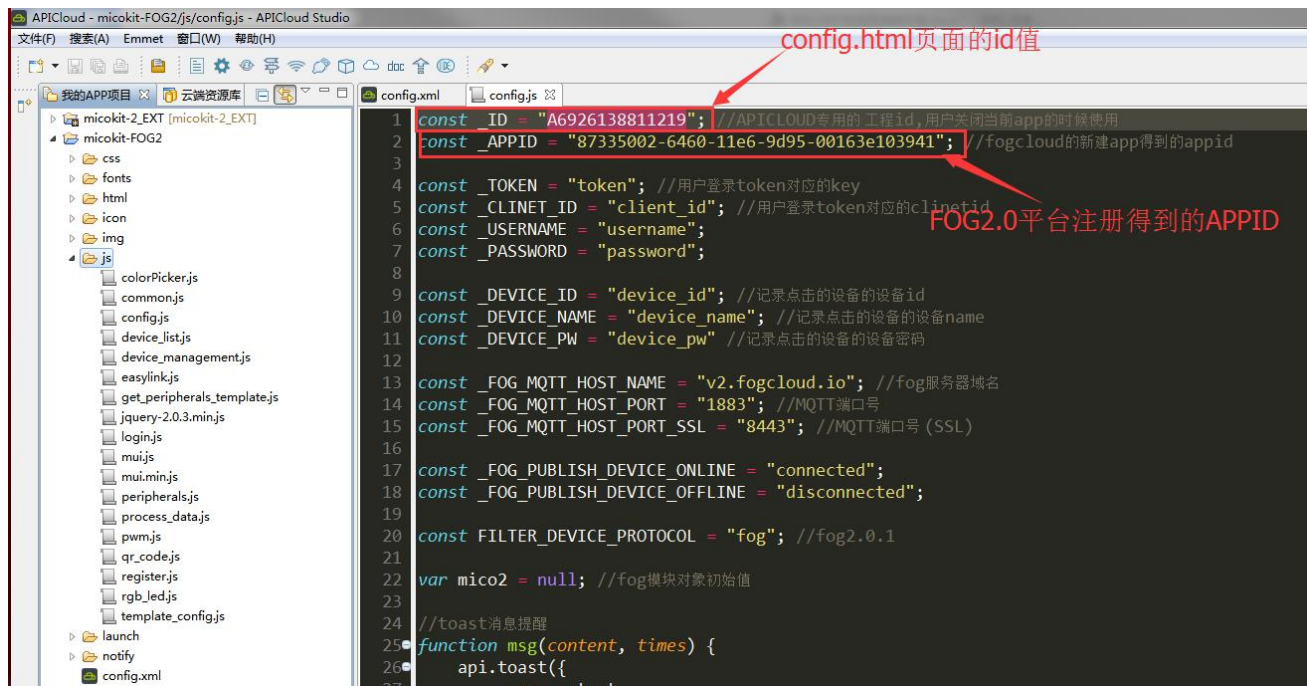
(2)直接粘贴到项目中



(3)要改动的地方有三处

(A)将根目录 config.html 中的<content src="index.html"/> 改为<content src="html/login.html"/>,因为 APP 的首页换成了 html 文件夹下面的 login.html 文件。

(B)将根目录 config.html 中的<widget id="A6927929655656" version="0.0.1"> id 值拿到, 然后填入到 js 文件夹里面的 config.js 文件中的_ID 值中

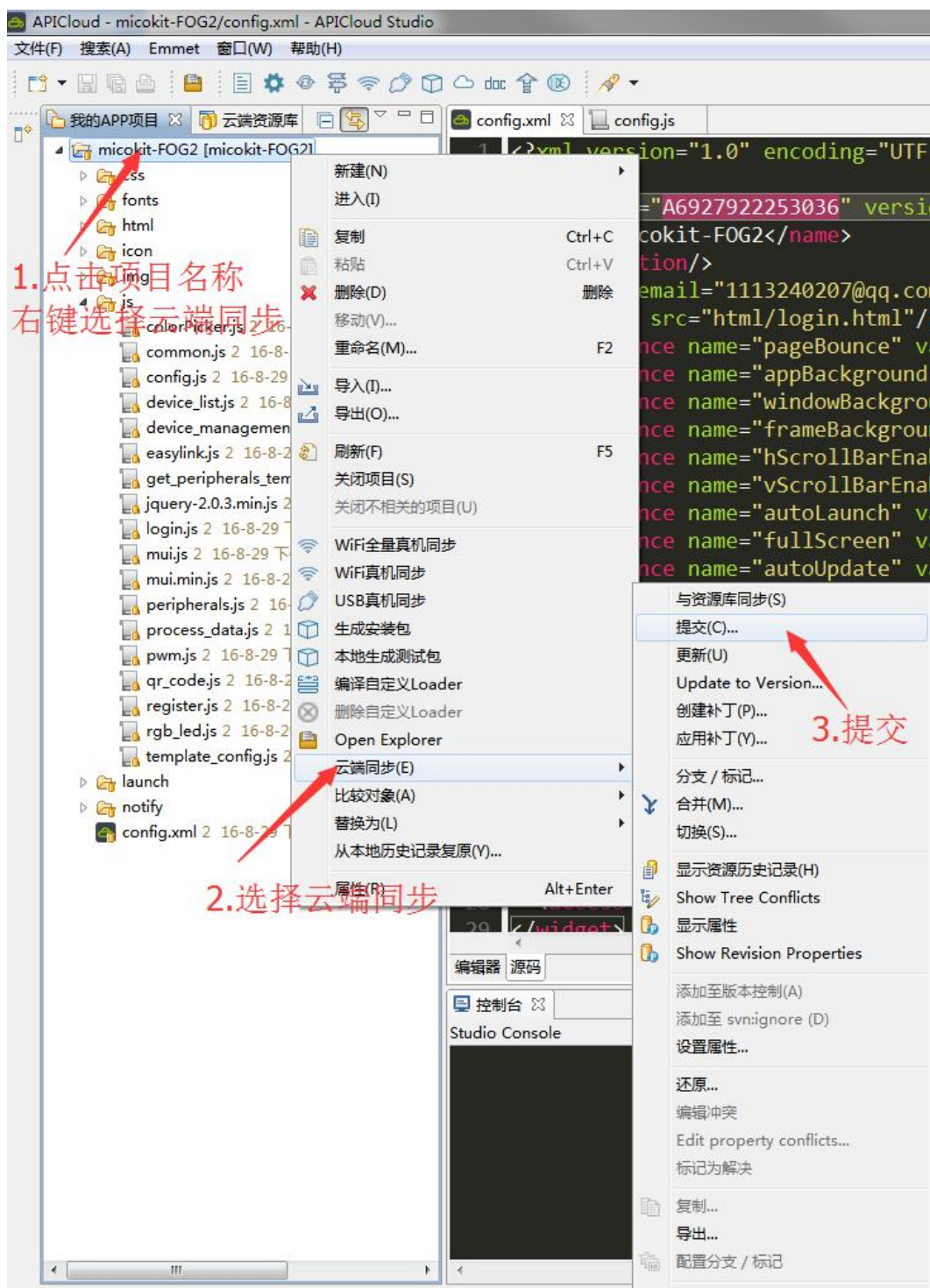


(C)将 js 文件夹里面的 config.js 文件中的 _APPID 的值换位当前 2.2.1 章节中在 FOG2.0 平台上应用注册得到的 APPID。

3.4 代码提交

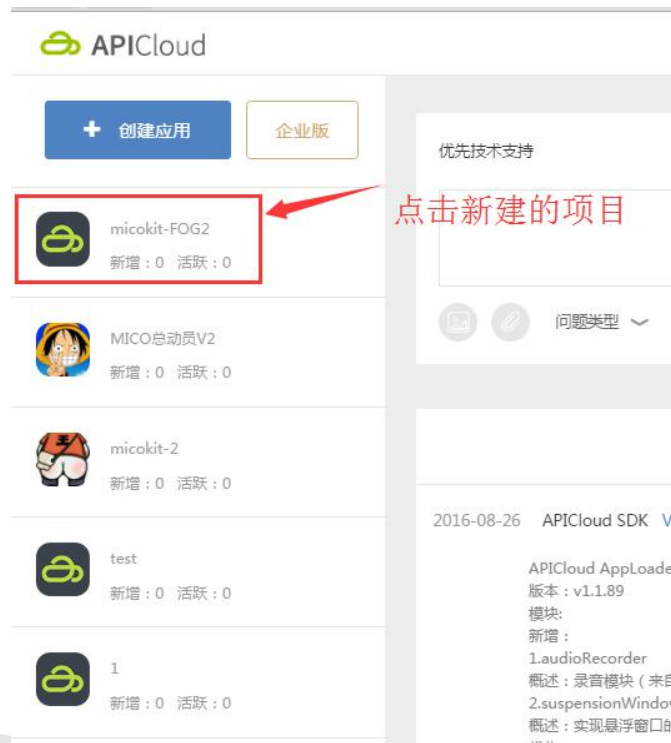
按照 4.3 的步骤做完之后, 你需要把你的代码提交到 APICloud 平台上。

注意: 鼠标选中项目名称再点击右键选择云端同步, 不要选择其他文件夹。

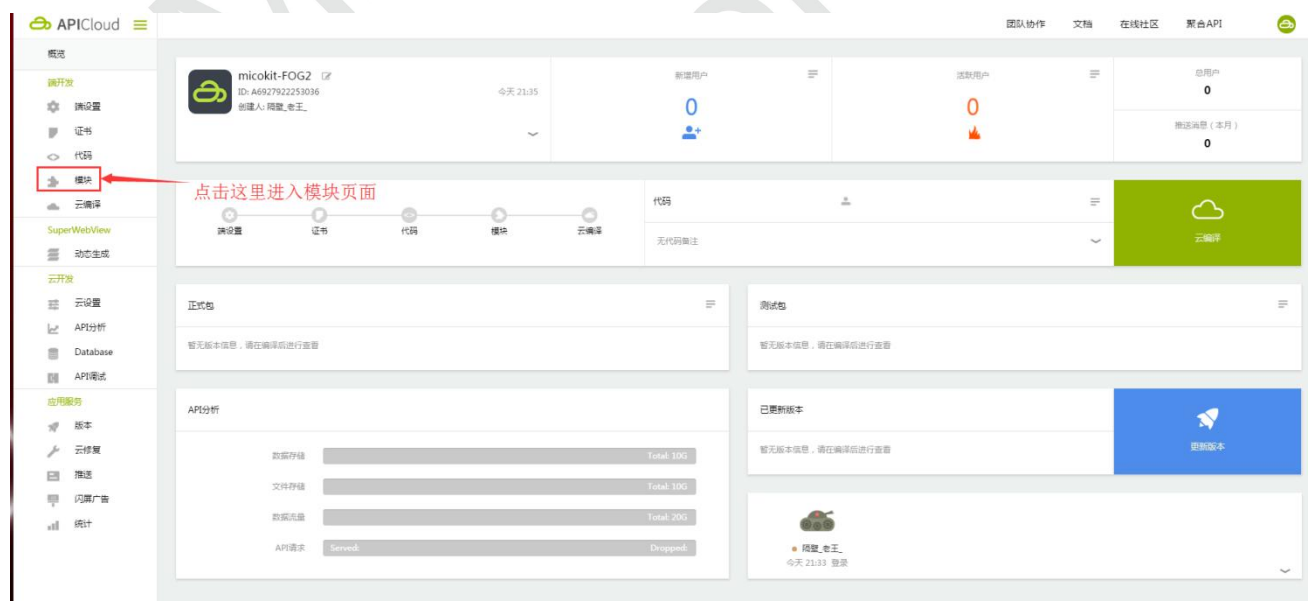


3.5 在你的项目中添加组件

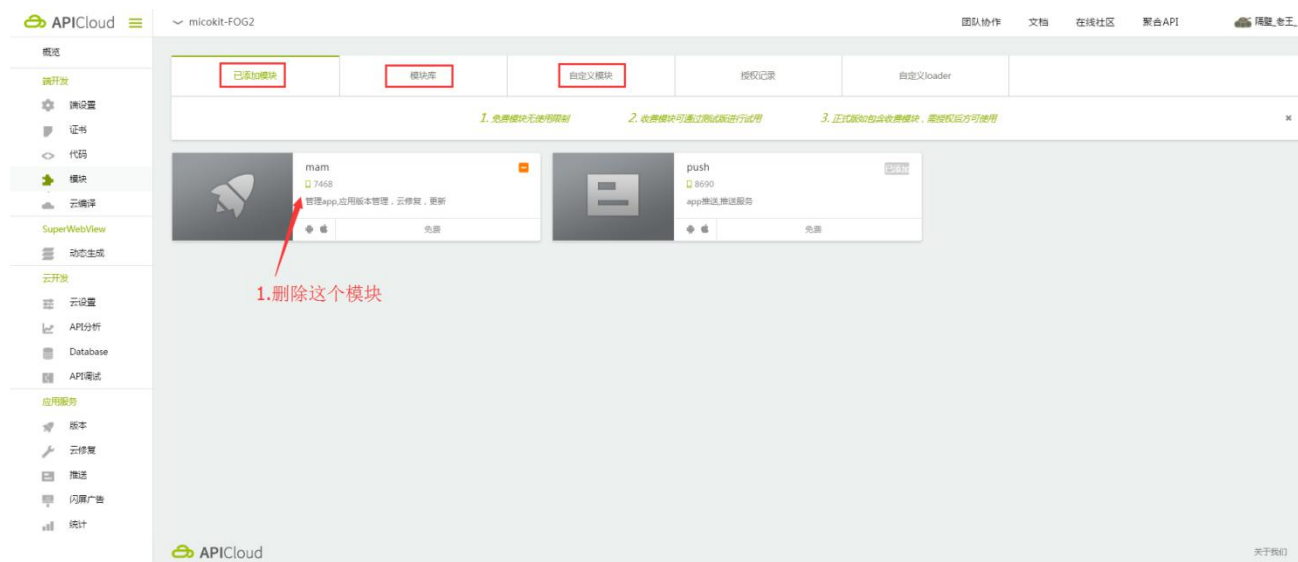
(1) 进入 APICloud 的控制台，发现新加入了一个项目



(2) 点击模块，然后进入模块控制页面



(3) 删除已添加模块中的 mam 模块，push 模块保留



(4)在模块库中搜索 FNScanner 点击右上角加号即可添加模块



(5)添加自定义模块，其中这里上传的 Android 和 IOS 的库是基于 APICloud 平台的 Android 和 IOS 库，库的名称都是 mico2.zip。



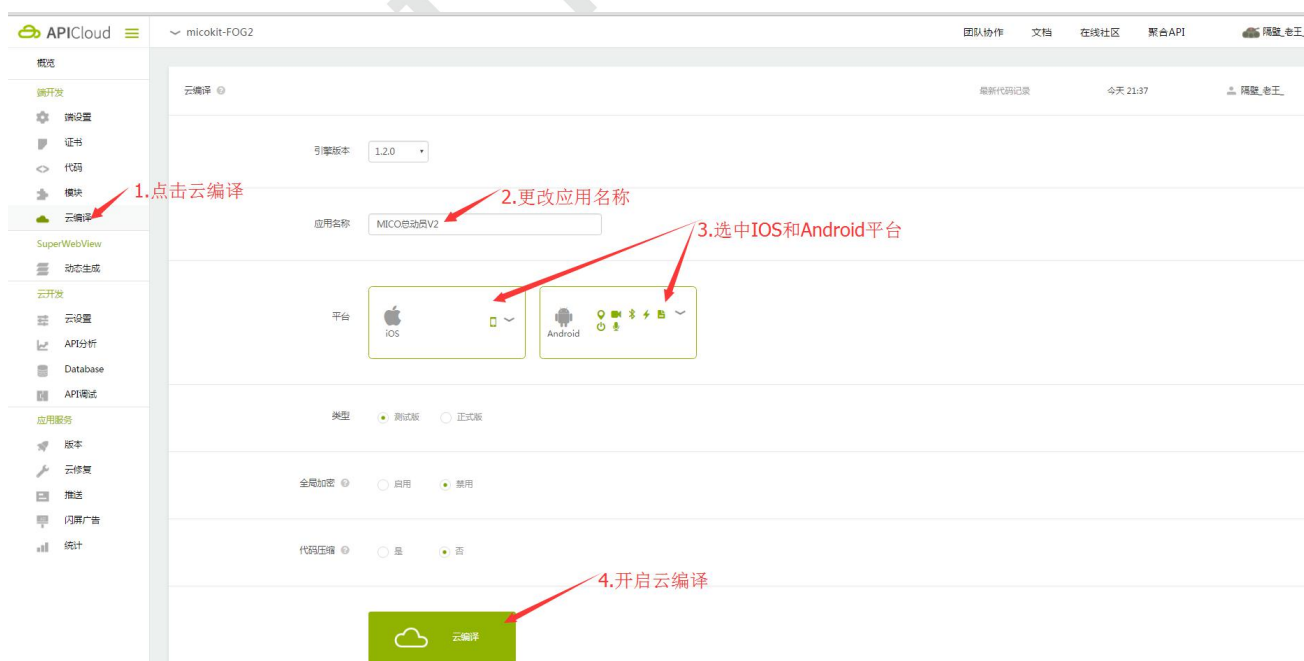
(6)添加自定义的模块到项目中



(7)已添加模块的中会有三个关键模块，分别为 FNScanner、mico2、push



3.6 云端编译



3.7 扫码下载 APP

(1)排队等待编译完成之后，会生成 IOS 和 Android 平台的两个二维码。

(2)因为 IOS 版本是测试版本，装在手机上的 APP 有效期只有 7 天。

(3)IOS 手机装完 APP 之后，打开 APP 时候可能会提示未受到信任。需要开发者在 IOS 里面的设置->通用->描述文件与设备管理中信任该应用。



IOS 版本 APP 二维码



Android 版本 APP 二维码

3.8 APP 的使用

详情见 APP 的使用说明文档。

4. 服务与支持

如需技术支持或产品咨询，请在办公时间拨打电话咨询上海庆科信息技术有限公司。

办公时间：

星期一至星期五 上午：9:00~12:00，下午：13:00~18:00

网址 : <http://mxchip.com/>

联系电话：+86-21-52655026

联系地址：上海市普陀区金沙江路 2145 弄 5 号楼 9 楼

邮编 : 200333

Email : sales@mxchip.com