

# 小程序开发体验：3元实现小程序从0到上线

## 前言

最近一段时间我用本文说的小程序开发方式上线了两个小程序：“Android开发测评”和“宝宝游戏力”。

# GitChat



我不是前端程序员，前端开发的经验也不多，且工作上也不需要我开发微信的小程序。那么，我为什么会热衷于此呢？

我觉得小程序的开发是精益创业较好的一种方式，试错成本较低。具体原因如下：

1. 你不需要关注用户管理系统，而且可以快速聚集用户和借助微信传播。
2. 开发的方式就JS+类HTML的方式，效率高速度快。
3. 小程序属于井喷时期，不能让自己太OUT了。

可能你不一定想创业，但通过快速低成本现实一个想法也是一件快事。而且快速的实现一个想法，并让用户使用和提供反馈，是很多移动端开发难于实现的（相对来说没有那么方便）。

微信小程序除了在用户上和其他公众平台（公众号）一样可以通过API获取到微信用户的基本信息，还帮助我们解决了设备适配的问题，我们可以不用关心是IOS还是Android，也不用关心屏幕的大小。这就让我们可以抽出身来，把有限的精力集中我们的核心业务上面。

快速的试错一个想法，行的话好好做，不行的话咱再试试别的。而微信小程序正好符合我的预期。

## 快速上手

相信看到这篇文章前，你肯定已经用过微信小程序了，如果没有，那么到微信“发现”->“小程序”中随便找个小程序体验一下。要是你体验过了，我们就开始快速进入开发环节。

有兴趣的朋友可以微信扫描下面的二维码，感受一下我的小程序。



微信扫一扫，使用小程序

“天下武功，唯快不破”。先把理论讲清楚了再上案例，那太慢了，关于小程序开发没有比自己亲自完成一款小程序的部署上线更好的学习方式了。

### 开发前准备：

1. 小程序账号（个人和公司的都行），如无，你需要新注册一个。

地址：<https://mp.weixin.qq.com/>。

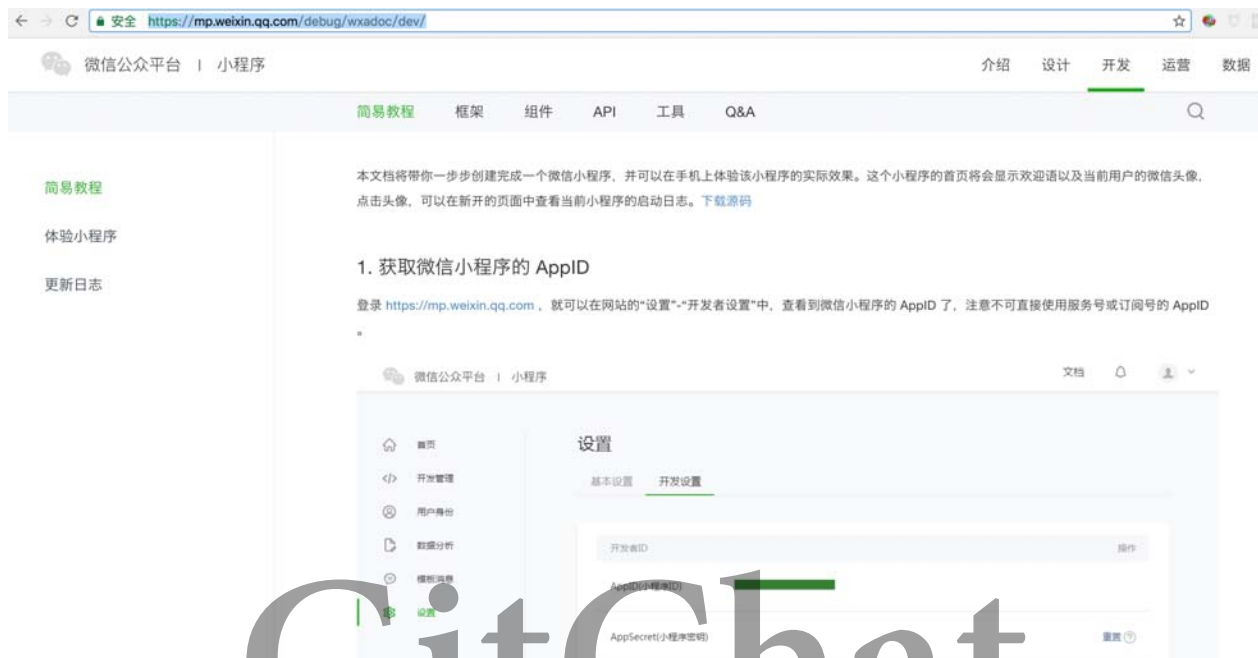
2. 下载安装“微信web开发者工具”，

下载地址：<https://mp.weixin.qq.com/debug/wxadoc/dev/devtools/download.html>

3. 会一点JavaScript、Html、CSS，至少能看懂一些语法。

4. 通读一下官方的开发指导文档。

官方开发指导文档：<https://mp.weixin.qq.com/debug/wxadoc/dev/>



## 大体方案

微信小程序并不是一个原生的移动应用，它必须运行在微信的应用框架上。如果你了解Facebook的React Native，你也可以把它理解成微信的React Native应用。

微信应用的框架会把小程序“包”从微信的服务器下载、加载然后运行。

开发上线小程序的流程大体是：

1. 开发并生成一个小程序的“包”。
2. 上传这个“包”到微信的服务器。
3. 部署小程序需要调用的业务服务器；（如果你的小程序不需要访问网络，这步可以免了）。
4. 发布上线。

如过你开发一个简单的工具类小程序，完全不需要后端，那你可以直接忽视第3步发布小程序。

那如果用到自己的业务服务器给小程序提供服务，该怎么做呢？

这时就涉及到微信的一些限制了，微信小程序的网络请求必须走HTTPS协议，所以你还需要为你的域名申请一个SSL证书。这就意味着你需要搭建一个支持HTTPS的后台服务，并配置好你的证书。

也就是：你要有一台WEB服务器（虚拟或独占）+ HTTPS服务 + 配置好SSL证书。

对于非Web后端开发人员来说，上面的任何一步都可能让你放弃。你很容易就和小程序可开发失之交臂。

## 3元方案

幸运的是网上有很多帮助我们解决“麻烦”事情的服务，对比了一下，我选择腾讯云的“微信小程序解决方案”：



腾讯云基于QQ、微信、腾讯游戏等海量业务的技术锤炼，从基础架构到精细化运营，从平台实力到生态能力建设，腾讯云将之整合并面向市场，使之能够为企业和创业者提供集云计算、云数据、云运营于一体的云端服务体验。

网址：<https://www.qcloud.com/>

白话文就是：腾讯云是腾讯的子公司，你可以信赖。

在腾讯云的微信小程序解决方案中，为我们一键解决了HTTPS部署、SSL证书、和负载均衡等问题，而且还帮我给出了一个可行的会话管理解决方案。我们只需要在微信web开发者工具开发小程序，在业务服务器（业务CVM）上完成后台的开发，为小程序段提供业务服务即可。



## 购买小程序解决方案

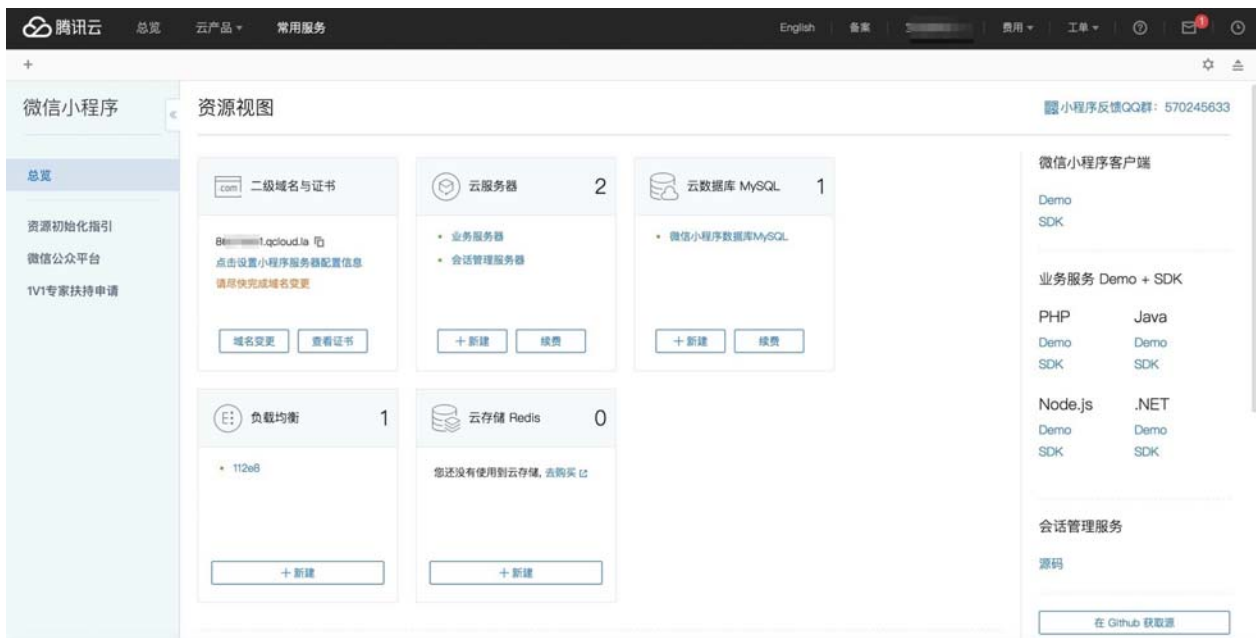
填写你的小程序配置页中的AppId和AppSecret

选择你比较熟悉或喜欢的后台开发语言，wafer Demo项目会被自动部署到这个服务器上。

3元可以试用两个月

**注意：**3元这样的活动腾讯会不定期举办，而且可能会要求你的小程序进行实名认证。如果你不喜欢或者认证不方便的话，可以直接选购买，第一次购买费用大约是90多元一个月。做为尝试这个费用并不算高，对吧。

支付成功后腾讯云需要10分钟左右的时间来配置我们的服务器。之后我们可以在腾讯云的管理中心看到腾讯云给我们配置的资源，我们快速开发只要关注“云服务器”中的“业务服务器”。



## 配置小程序的服务器域名

点击“二级域名与证书”中的配置小程序服务器信息那一项，有一键配置的，当然你也可以自己拷贝到微信公众平台的设置里。小程序是在微信的框架上运行的，微信限制了小程序对网络的访问，如果没有在微信公众平台给小程序设置好合法的域名，小程序是不能发起网络请求的。



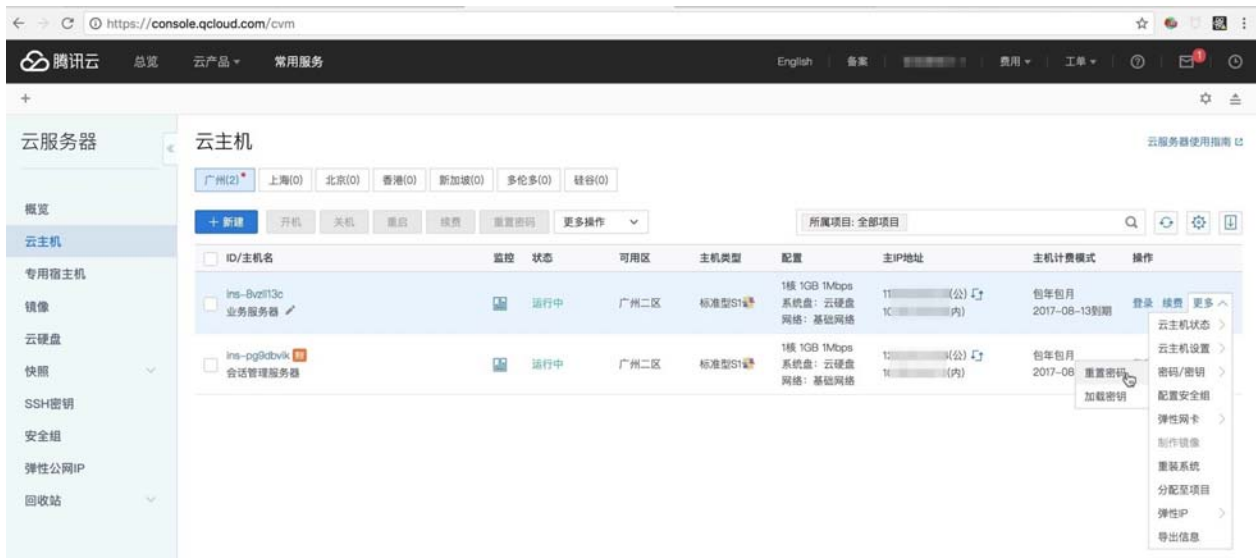
配置好后只要使用这个小程序AppId的小程序应用都可以访问我们设置的域名。

## 登录业务服务器

业务服务器配置好后，我们需要在腾讯云的后台给它设置一个登陆密码，重启后我们就可以在本地通过SSH管理和控制这台服务器了。

修改密码：





SSH登录：

```
ssh /Users/lan/Documents/weixin/babyapp — ssh root@11
...babyapp — fish /Users/lan/Documents/weixin/babyapp — -fish
ssh /Users/lan/Documents/we...yapp — ssh root@11
Last login: Tue Jun 13 11:29:11 on ttys005
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
[x] ~/D/w/babyapp on master ssh root@119.29.252.25 11:46:25
The authenticity of host '119.29.252.25 (119.29.252.25)' can't be established.
ECDSA key fingerprint is SHA256:HL/...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '119.29.252.25' (ECDSA) to the list of known hosts.
root@119.29.252.25's password:
Permission denied, please try again.
root@119.29.252.25's password:
Last login: Tue Jun 13 11:46:12 2017

root@10.10.10.10:~
# ls
anaconda-ks.cfg

root@10.10.10.10:~
# |
```

我是MAC系统，Windows系统的话需要自己安装SSH工具。

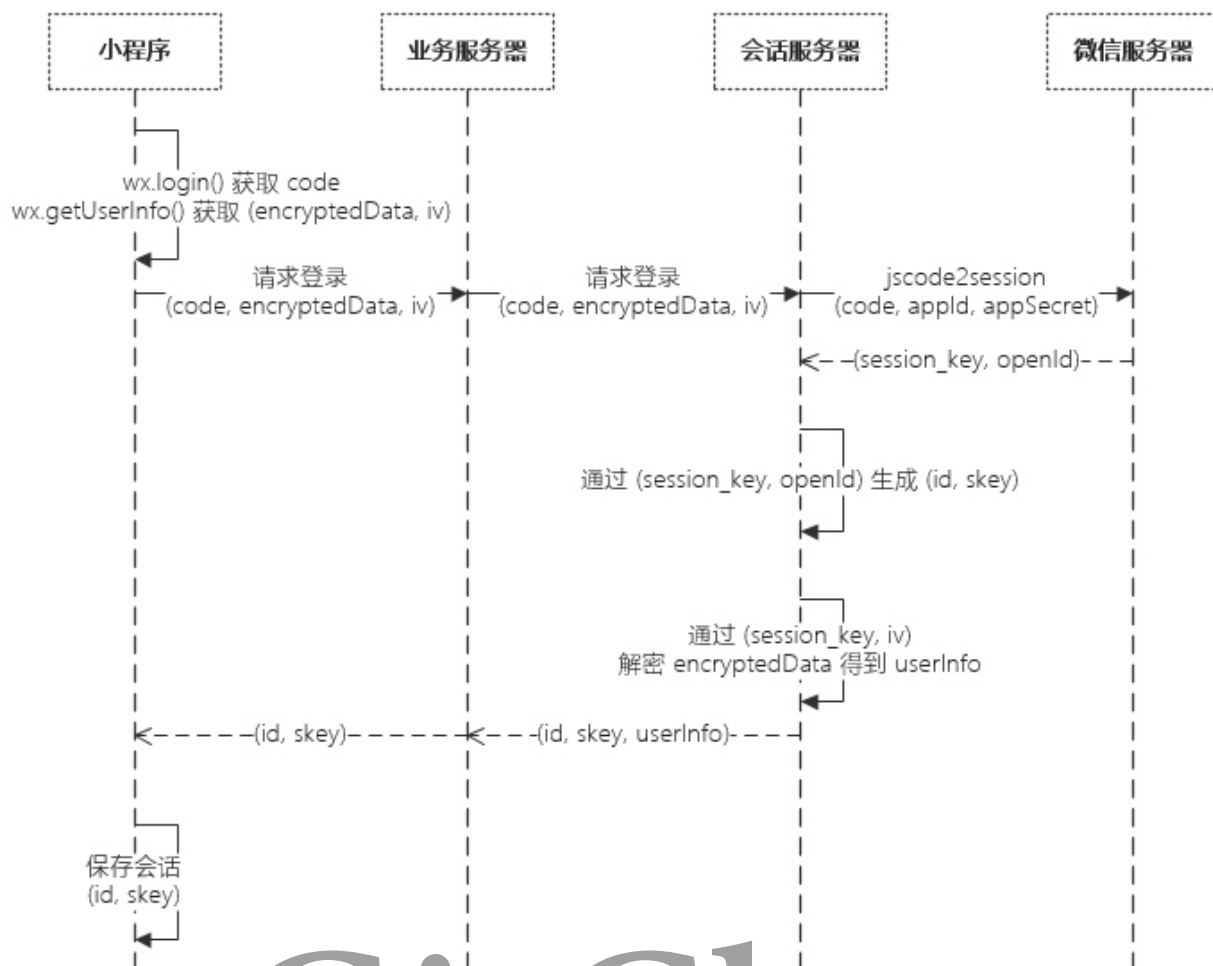
到这里，我们完成了对微信公众平台（小程序端）和腾讯云业务服务器的配置，下边就要开始从腾讯云自动部署的Demo（Wafer项目）的修改开始。

会话服务方面的知识可以从下面的地址了解，目前我们先快速完成上线是第一要务。

会话服务和业务服务：

<https://github.com/tencentyun/wafer/wiki/%E4%BC%9A%E8%AF%9D%E6%9C%8D%E5%8A%A1>





# GitChat

wafer

Wafer 是腾讯云面向广大开发者提供的小程序开发全栈资源套件，套件提供小程序会话管理服务和 WebSocket 信道服务，部署方式具备良好的弹性伸缩能力，可以快速应对业务的爆发增长，同时具备较低的开发门槛。

项目地址：<https://github.com/tencentyun/wafer>

腾讯云提供的微信小程序解决方案会帮我们自动部署wafer到服务器，示例代码部署在业务服务器的“/data/release/node-weapp-demo”目录。

业务服务器我选的是Node.js的开发语言，解决方案为我们部署好了Node.js和PM2:

运行示例的 Node 版本：v4.6.0

Node 进程管理工具：pm2

## 测试wafer的客户端和后端

先暂停掉业务服务器上的Weapp，使用SSH登录上业务服务器，可以查看自动部署的weapp服务端应用（即wafer的后端）：

```
ssh /Users/lan — ssh root@1:
Last login: Tue Apr 25 18:51:45 on ttys001
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
x> ~ ssh root@1:
18:59:21
root@1: s password:
Last login: Tue Apr 25 18:51:54 2017 from 1:

root@1:~
# pm2 list
```

App name	id	mode	pid	status	restart	uptime	cpu	mem	w
atching									
weapp enabled	0	fork	10788	online	63	45h	0%	42.9 MB	

```
Use `pm2 show <id|name>` to get more details about an app

root@1:~
#
```

在业务服务器的SSH下输入：

```
pm2 stop weapp
```

在微信web开发者工具，导入“wafer-client-demo”的代码，导入时要填写正确AppId:

微信web开发者工具 v0.17.172600

< 返回

添加项目

AppID wx 小程序ID

填写小程序AppID 无 AppID

项目名称

项目目录 选择

取消 添加项目

小程序端Demo地址：<https://github.com/tencentyun/wafer-client-demo>

关掉后用微信小程序工具测试：



发现访问出错，重启（`pm2 start process.json`）刚刚的业务服务器的weapp，访问成功的话，说明我们从小程序端到业务服务端已经走通了。

## 代码协同和管理

我使用GitHub进行代码管理和同步：本机开发和修改代码后提交Github，业务服务器从Github更新代码。当然你可以使用其他的git平台或服务器，其他的代码管理如SVN都可以。

但没有代码的版本控制和管理不是一个好习惯，过一段时间你可能都会忘了自己的代码放哪里。

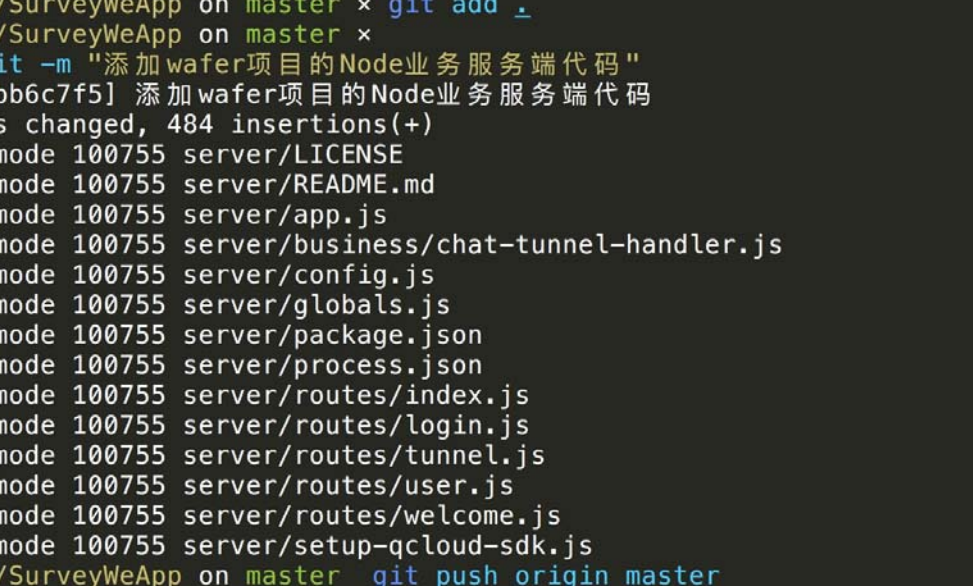
下载Node业务服务端的源码到本地：<https://github.com/tencentyun/wafer-node-server-demo>

注意修改两个地方：

1. config.js的port端口改变80的，因为腾讯云的负载均衡服务器指向这个端口。
2. process.json的”name”那项可以换一个名子，和刚刚我们stop掉的weapp区分开。

然后提交到Github作为我们自己的项目框架：

```


  SurveyWeApp — fish /Users/lan/Documents/weixin/SurveyWeApp — -fish
  README.md client server
  x> ~/D/w/SurveyWeApp on master x git add .
  x> ~/D/w/SurveyWeApp on master x
  git commit -m "添加wafer项目的Node业务服务端代码"
  [master bb6c7f5] 添加wafer项目的Node业务服务端代码
  14 files changed, 484 insertions(+)
  create mode 100755 server/LICENSE
  create mode 100755 server/README.md
  create mode 100755 server/app.js
  create mode 100755 server/business/chat-tunnel-handler.js
  create mode 100755 server/config.js
  create mode 100755 server/globals.js
  create mode 100755 server/package.json
  create mode 100755 server/process.json
  create mode 100755 server/routes/index.js
  create mode 100755 server/routes/login.js
  create mode 100755 server/routes/tunnel.js
  create mode 100755 server/routes/user.js
  create mode 100755 server/routes/welcome.js
  create mode 100755 server/setup-qcloud-sdk.js
  x> ~/D/w/SurveyWeApp on master git push origin master
  Counting objects: 19, done.
  Delta compression using up to 8 threads.
  Compressing objects: 100% (19/19), done.

```

在业务服务器上下载同步我们Github上的业务服务端代码。先切换到“/data/release/”目录（自动部署的node-weapp-demo就放在这个目录）：

```
root@1[REDACTED]:~  
# cd /data/release/  
  
root@1[REDACTED]:/data/release  
# ls  
node-weapp-demo  
  
root@1[REDACTED]:/data/release  
# git clone git@github.com:goeasyway/SurveyWeApp.git  
正克隆到 'SurveyWeApp'...  
The authenticity of host 'github.com (1[REDACTED]3)' can't be established.  
RSA key fingerprint is 16:[REDACTED].  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'github.com,1[REDACTED]3' (RSA) to the list of known hosts.  
remote: Counting objects: 22, done.  
remote: Compressing objects: 100% (19/19), done.  
remote: Total 22 (delta 1), reused 22 (delta 1), pack-reused 0  
接收对象中: 100% (22/22), 7.46 KiB | 0 bytes/s, done.  
处理 delta 中: 100% (1/1), done.  
  
root@1[REDACTED]:/data/release  
# |
```

我使用了私有的Github创库，提前需要配置SSH的key，如果你们使用公开的创库应该不需要配置。

切换到server所在目录，执行：“npm install”，安装必要的依赖包：



```
ssh /Users/lan — ssh root@100.700.110.700
root@100.700.110.700: /data/release/SurveyWeApp/server
# ls
app.js      config.js  LICENSE    process.json  routes
business    globals.js package.json README.md     setup-qcloud-sdk.js

root@100.700.110.700: /data/release/SurveyWeApp/server
# npm install
npm WARN package.json wafer-node-server-demo@1.0.0 No repository field.
lodash@4.17.4 node_modules/lodash

morgan@1.8.1 node_modules/morgan
├── on-headers@1.0.1
├── basic-auth@1.1.0
├── depd@1.1.0
├── on-finished@2.3.0 (ee-first@1.1.1)
└── debug@2.6.1 (ms@0.7.2)

body-parser@1.17.1 node_modules/body-parser
├── content-type@1.0.2
├── bytes@2.4.0
├── depd@1.1.0
├── on-finished@2.3.0 (ee-first@1.1.1)
└── qs@6.4.0
```

用pm2启动服务：pm2 start process.json（记得先暂停weapp）

```
ssh /Users/lan — ssh root@100.700.110.700
root@100.700.110.700: /data/release/SurveyWeApp/server
# pm2 start process.json
[PM2] [WARN] Applications weapp2 not running, starting...
[PM2] App [weapp2] launched (1 instances)

┌──────────┬──────────┬──────────┬──────────┬──────────┬──────────┬──────────┬──────────┬──────────┬──────────┬──────────┐
│ App name  │ id        │ mode      │ pid       │ status    │ restart   │ uptime    │ cpu       │ mem       │            │ w         │
├──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┘
│ atching  │           │           │           │           │           │           │           │           │           │           │
├──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┘
│ weapp    │ 0         │ fork      │ 0          │ stopped   │ 63        │ 0          │ 0%        │ 0 B       │            │           │
├──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┘
│ enabled  │           │           │           │           │           │           │           │           │           │           │
├──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┘
│ weapp2   │ 1         │ fork      │ 23743      │ online    │ 0         │ 0s         │ 33%       │ 9.6 MB    │            │           │
├──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┘
│ enabled  │           │           │           │           │           │           │           │           │           │           │
├──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┼──────────┘

Use `pm2 show <id|name>` to get more details about an app

root@100.700.110.700: /data/release/SurveyWeApp/server
#
```

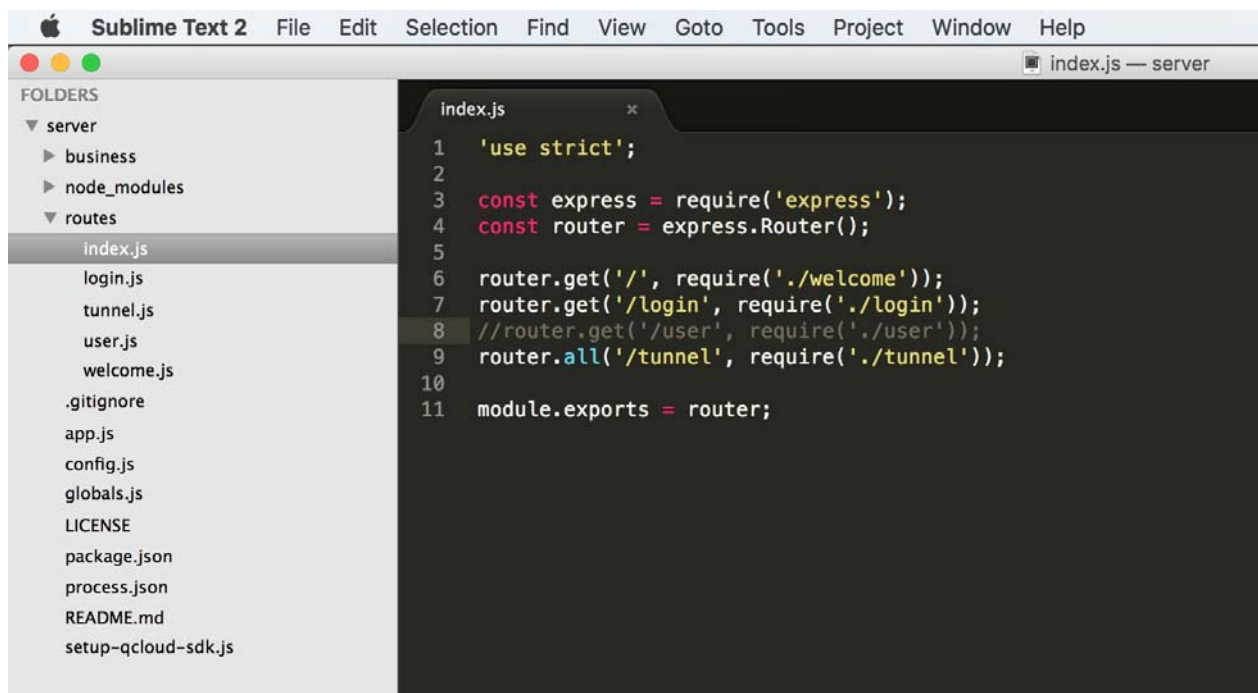
重新用小程序客户端测试业务服务器，登录成功，业务服务器正常运行，这时候小程序端链接的后端服务不再是自动部署的wafer后端，而是我们自己提交的后端。

我们自己部署的Node业务服务端成功，下边可以开始修改业务代码，添加相应的功能。

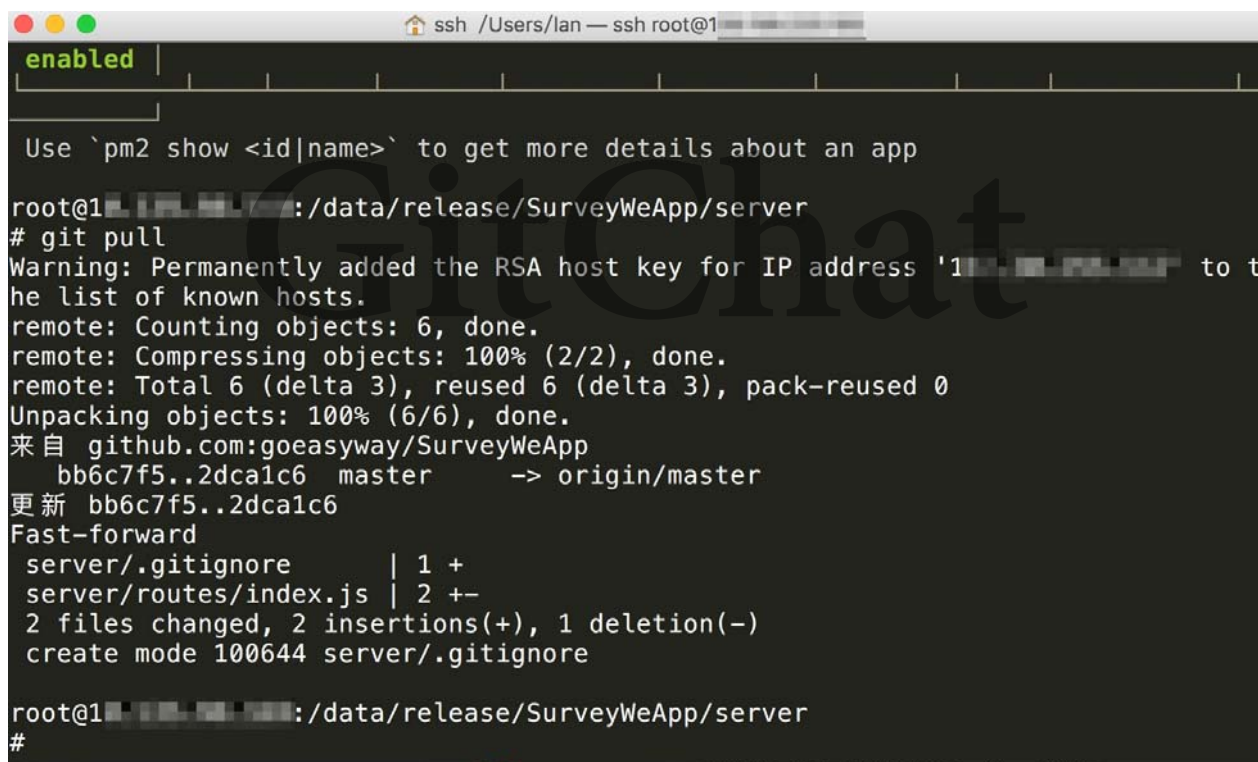
## 修改代码

我们来试一下：

修改本地的routes/index.js的一句代码，让/user这个请求接口失效：

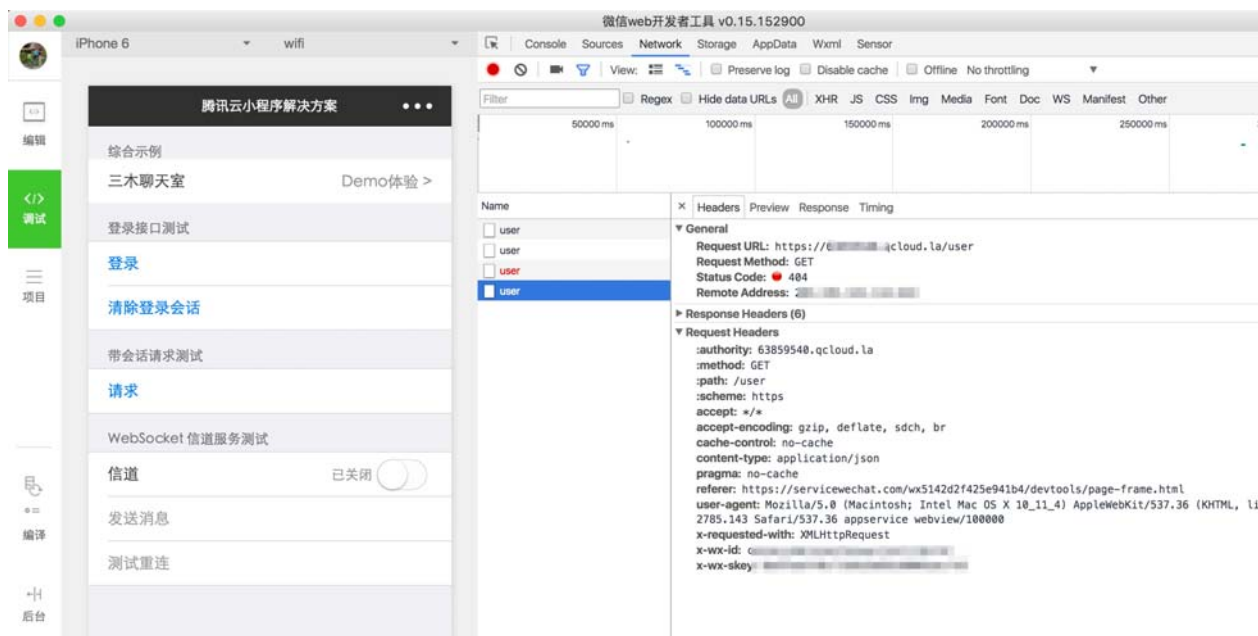


然后上专到Github，业务服务器使用git pull更新代码：



更新代码就可以，PM2会自动重新加载执行app.js。

小程序客户端点击“请求”发出请求时，返回404状态：



到此我们的整个流程走通了。现在可以根据自己的需求修改这个业务端代码，提供更多的接口，然后再修改我们的小程序端代码。

## 开发

这个就看各位发挥了，如果你不太喜欢用Node.js也可以选其他的后台语言。

小程序端的开发可以看官方的指导文档（<https://mp.weixin.qq.com/debug/wxadoc/dev/>），写的还是很简单易懂的。

不过前提需要你懂一些JavaScript、html和CSS方面的知识。

## 预览和上传

最后到了我们要上线的时候了，我们之前的开发都是在微信web开发者工具这个IDE进行的，现在我们来看看下面的流程。

微信web开发工具为我们集成小程序的打包和上传工作，你只需要在“编辑”栏完成代码开发后点击“项目”栏下的“上传”提交就好。





在确认上传前最好在手机上进行测试，点击“预览”，扫码便可以在开发者的微信（手机）上进入“开发版”的小程序。



上传到微信服务器的包还不能被最终的用户看到，选用通过微信的审核才行，有点类似IOS应用的应用商店上架的过程。



一般审核的时间为：1-3个工作日。审核通过时，小程序的管理员微信号会收到提醒。

## 发布上线

只有“审核版本”通过审核后，你点击“提交发布”的小程序才算完成了上线。

## 线上版本

版本号	发布者	兰柳学(Even Lan)
V0.0.1	发布时间	2017-06-15 14:45:00
	描述	内测版本：宝宝积分记录功能。

详情



## 审核版本

版本号	开发者	兰柳学(Even Lan)
v0.0.2	提交审核时间	2017-06-18 22:45:00
<u>审核通过，待发布</u>	描述	修改不能创建baby用户的BUG。

这步提交用  
户才能使用



提交发布



发布成功后，用户才可以在小程序的搜索里找到你：

# GitChat



大功告成

到此你的微信小程序已经成功上线了，和可以通过公众号或者群转发等方式向用户公布你的小程序。

当然也可以发布你的二维码给你的用户，在微信公众平台设置中可以下载各个尺寸的小程序二维码。



## 开发小程序注意

等你正真感受了一下小程序的开发后，再来看开发的注意才更容易理解和记住要注意的地方。

1. 每个小程序“包”的大小是有限制的，不能超过1M。
2. 消息推送：可以使用消息模板完成，但是对条数和时间有限制。
3. 小程序同时只能有5个网络请求连接。
4. 小程序的本地存储是有大小限制的。
5. view的background-image不能使用本地（小程序“包”）中的图片资源，image控件可以。
6. API兼容性：注意小程序是有基础版本库的，不同的版本库支持的功能有差；如一些新的功能（小程序内的组件可点击分享）在旧版的微信上是不能调用的。

还有一些开发心得和体会留到在线交流时在和大家详聊。有问题的朋友可以在下边留意。