

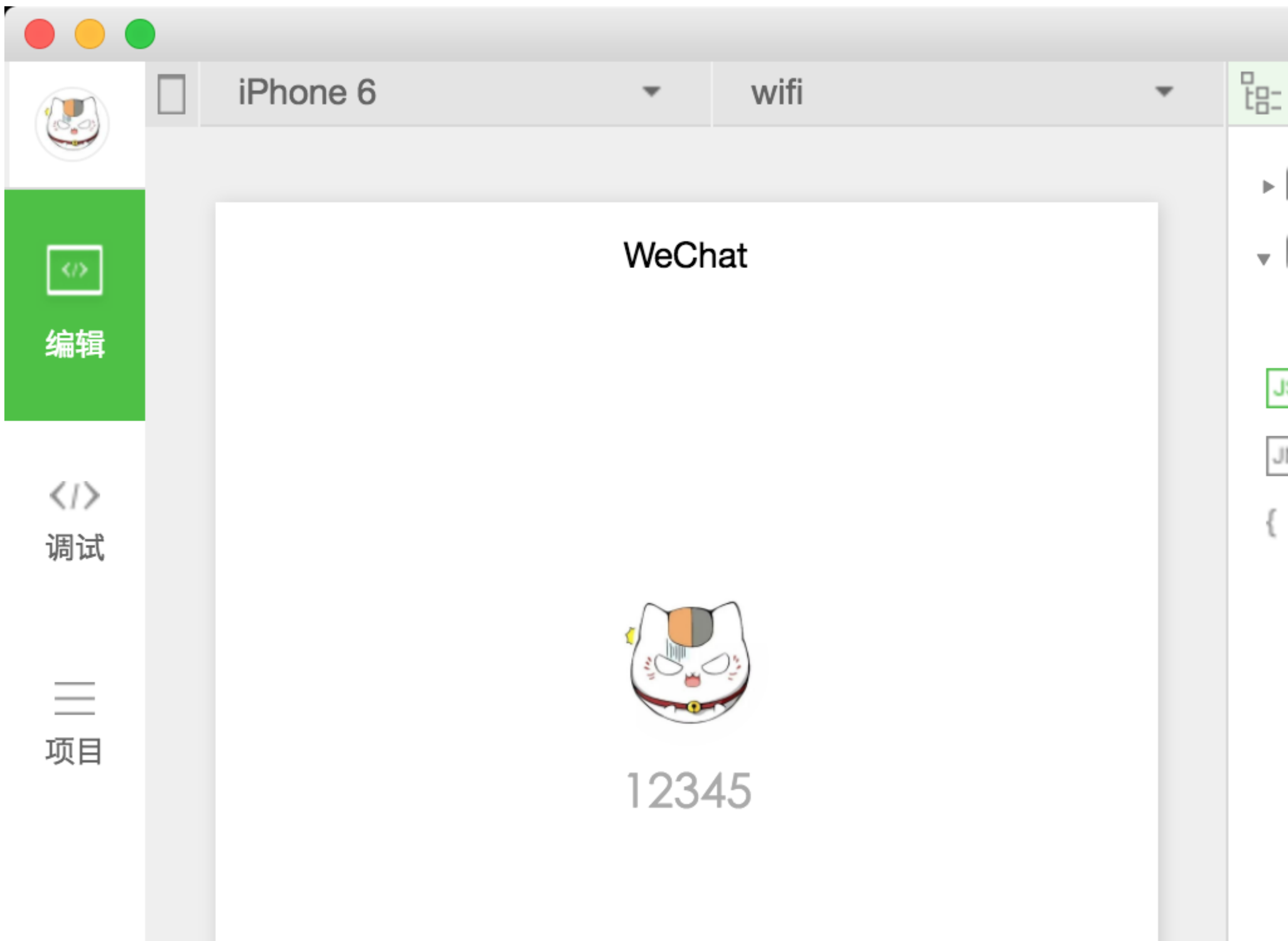
微信小程序初接触与踩坑日记

一：微信小程序初接触

微信主要做了这么几件事：

1. 不怎么流通的数据流
2. 指定 ui
3. 破破烂烂的 ide

尤其自带的 ide 简直不能忍受



想要用其他 ide (编辑器) 写微信小程序 , 我们需要做下面的事 :

1. 将 html 编译成 wxml
2. 将 sass 编译成 wxss
3. 由于 json 的难阅读所以使用 yaml 进行配置

gulp 配置

```
var gulp = require('gulp')
var fs = require('fs')
var sass = require('gulp-sass')
var rename = require('gulp-rename')
var yaml = require('gulp-yaml')
var babel = require('gulp-babel')
var image = require('gulp-image');

gulp.task('wxml', function () {
  gulp.src('./app/**/*.html')
    .pipe(rename(function (path) {
      path.extname = '.wxml'
    }))
    .pipe(gulp.dest('build/'))
})

gulp.task('sass', function () {
  gulp.src('./app/**/*.scss')
    .pipe(sass({outputStyle: 'expanded'}))
    .pipe(rename(function (path) {
      path.extname = '.wxss'
    }))
    .pipe(gulp.dest('build'))
})
```

```
})

gulp.task('yaml', function () {
  gulp.src('./app/**/*.yaml')
    .pipe(yaml({ space: 2 }))
    .pipe(gulp.dest('build'))
})

gulp.task('js', function () {
  gulp.src('./app/**/*.js')
    .pipe(babel({
      presets: ['es2015']
    }))
    .pipe(gulp.dest('build'))
})

gulp.task('image', function () {
  gulp.src(['./app/**/*.png', './app/**/*.jpg'])
    .pipe(image())
    .pipe(gulp.dest('build'))
})

gulp.task('default', ['wxml', 'sass', 'yaml', 'js'], function () {
  gulp.watch('./app/**/*.html', ['wxml'])
  gulp.watch('./app/**/*.scss', ['sass'])
  gulp.watch('./app/**/*.yaml', ['yaml'])
  gulp.watch('./app/**/*.js', ['js'])
})
```

下面是非前端或者非 nodejs 开发者看的：

首先你得安装 nodejs 和 npm，然后执行 `npm i -g gulp-cli`。然后 git clone 项目。进入文件夹中执行 `npm install`，然后执行 `gulp` 即可进行编译。

将编译后的 `build/` 目录作为小程序主目录。完成。

atom 插件

个人花了点时间搞了一个 atom 插件。作为 atom 开发的补充，具有部分自动补全的功能，都是体力活。atom 搜索安装 wxapp 插件就好。重启编辑器后直接使用

基础配置

微信已经将目录结构定死了所以可以发挥的余地也不是很大，基本来说只要跟着他的教程走做个简单的应用还是很容易的

模板目录如下：

```
weapp/  
  app/  
    pages/  
      index/  
      logs/  
    utils/  
    app.js  
    app.scss  
    app.yml
```

```
node_modules/  
  
build/  
  
.gitignore  
  
gulpfile.js  
  
package.json
```

其中 build 为打包后的目录，微信小程序就运行该目录。

另外我个人搞了个 atom 插件（我习惯用 atom）包括了部分自动提示，还可以。

小程序示例

接下来我们将使用 v2ex 的接口来进行简单的示例。v2ex 的接口相当简洁，非常适合做示例。

接口整理

最热主题：

url: <https://www.v2ex.com/api/topics/hot.json>

method: get

节点信息

url: <https://www.v2ex.com/api/nodes/show.json>

method: get

query: name

最新主题

url:

mehtod: get

用户主页

url: https://www.v2ex.com/api/members/show.json

method: get

query: id or username

简单配置后台

由于小程序域名有所限制，所以这里自己搭个后台简单转发一下，代码如下：

```
var https = require('https')
var express = require('express')
var fs = require('fs')
var request = require('request')

var app = express()

var options = {
  key: fs.readFileSync('./wechat/wechat.key'),
  cert: fs.readFileSync('./wechat/wechat.pem')
}

app.get('/', (req, res) => {
```

```
res.json({
  name: 1
})
})

app.get('/hot', (req, res) => {
  var url = 'https://www.v2ex.com/api/topics/hot.json'
  request(url, (err, result, body) => {
    res.json(body)
  })
})

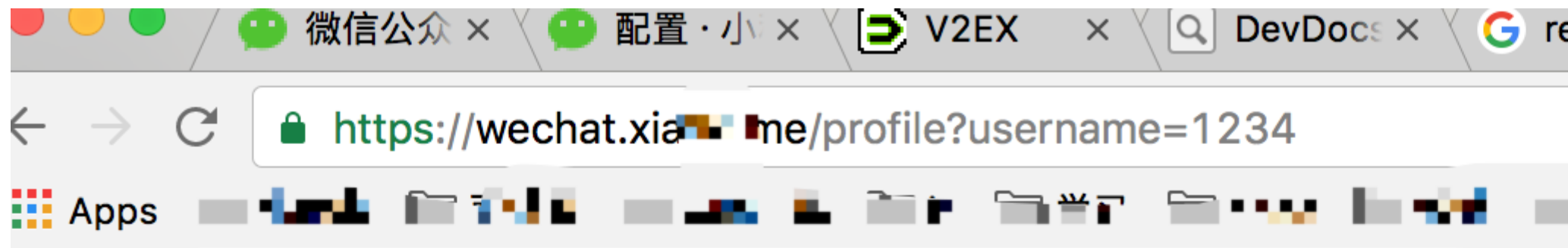
app.get('/newest', (req, res) => {
  var url = 'https://www.v2ex.com/api/topics/latest.json'
  request(url, (err, result, body) => {
    res.json(body)
  })
})

app.get('/profile', (req, res) => {
  var username = req.query.username
  var url = 'https://www.v2ex.com/api/members/show.json?username=' + username
  request(url, (err, result, body) => {
    res.json(JSON.parse(body))
  })
})

var server = https.createServer(options, app)

server.listen(443, () => console.log('server is running'))
```

这里是申请的阿里云免费证书在本地起个服务器，然后用 natapp (natapp.cn) 进行转发。



```
{  
  "status": "found",  
  "id": 126757,  
  "url": "http://www.v2ex.com/member/1234",  
  "username": "1234",  
  "website": "",  
  "twitter": "",  
  "psn": "",  
  "github": "",  
  "btc": "",  
  "location": "",  
  "tagline": "",  
}
```

如图服务器搭起来了（就是简单的转发了三个接口。而且 v2ex 的接口也实在太简单了），下面就是小程序的工作了。

二：踩坑日记

1. 如果 pages 里的 js 文件是空的，微信小程序未递补上去，然后后面的 page js 文件就为空，真是奇奇怪怪的 bug

```
pages:
- pages/index/index
- pages/categories/categories
- pages/items/items
- pages/cart/cart
# here bug is when put page at the last, it occurs that page not found
# solved! all pages must have non empty *.js, filled with page
- pages/profile/profile
```

2. image 标签有问题，有待探究
3. 不知道为何连 websocket 显示是 xhr 请求