

# ios下JS与OC互相调用（八） -- Cordova详解+实战



Haley\_Wong (/u/a8cf6d63e889) [+ 关注](#)

2016.09.29 11:22\* 字数 2038 阅读 4351 评论 13 喜欢 28 阅读 4351 评论 13 喜欢 28

(/u/a8cf6d63e889)

## 扯两句，可以跳过

由于项目中Cordova相关功能一直是同事在负责，所以也没有仔细的去探究Cordova到底是怎么使用的，又是如何实现JS 与 OC 的交互。所以我基本上是从零开始研究和学习Cordova的使用，从上篇在官网实现命令行创建工程，到工程运行起来，实际项目中怎么使用Cordova，可能还有一些人并不懂，其实我当时执行完那些命令后也不懂。后来搜索了一下关于Cordova 讲解的文章，没有找到一篇清晰将出如何使用Cordova，大多都是讲如何将Cordova.xcodeproj拖进工程等等。我不喜欢工程里多余的东西太多，其实并不需要将Cordova 整个工程拖进去，只需要一部分就够了，下面我会一一道来。

## 1.新建工程，添加Cordova 关键类

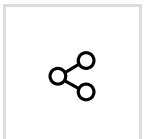
我这里用Xcode 8 新建了一个工程，叫 JS\_OC\_Cordova ,然后将Cordova关键类添加进工程。

有哪些关键类呢？

这里添加 config.xml 、 Private 和 Public 两个文件夹里的所有文件。工程目录结构如下：



然后运行工程，😏😏😏 ，你会发现报了一堆的错误：



为什么有会这么多报错呢？

原因是Cordova 部分类中，并没有 `#import <Foundation/Foundation.h>`，但是它们却使用了这个库里的NSArray、NSString 等类型。

为什么用在终端里用命令行创建的工程就正常呢？

那是因为用命令行创建的工程里已经包含了pch 文件，并且已经import 了 Foundation框架。截图为证：

其实这里有两种解决方案：1、在报错的类里添加上 `#import <Foundation/Foundation.h>`；2、添加一个pch 文件，在pch文件里加上 `#import <Foundation/Foundation.h>`。

我选择第二种方案：

再次编译、运行，依然报错。What the fuck 🤯🤯🤯 !!!

不用急，这里报错是因为Cordova的类引用错误，在命令行创建的工程里Cordova 是以子工程的形式加入到目标工程中，两个工程的命名空间不同，所以import 是用 类似这样

的方式 `#import <Cordova/CDV.h>`，但是我们现在是直接的目标工程里添加Cordova，所以要把 `#import <Cordova/CDV.h>` 改为 `#import "CDV.h"`。其他的文件引用报错同理。

当然，如果想偷懒，也可以从后面我给的示例工程里拷贝，我修改过的Cordova库。

## 2.设置网页控制器，添加网页

首先将 `ViewController` 的父类改为 `CDVViewController`。如下图所示：

这里分两种情况，加载本地HTML 和远程HTML 地址。

**\*\* 加载本地HTML \*\***

加载本地HTML，为了方便起见，首先新建一个叫 `www` 的文件夹，然后在文件夹里放入要加载的HTML和 `cordova.js`。

这里把 `www` 添加进工程时，需要注意勾选的是create foler references,创建的是蓝色文件夹。

最终的目录结构如下：

上面为什么说是方便起见呢？

先说答案，因为 `CDVViewController` 有两个属性 `wwwFolderName` 和 `startPage`，`wwwFolderName` 的默认值为 `www`，`startPage` 的默认值为 `index.html`。

在 CDVViewController 的 viewDidLoad 方法中，调用了与网页相关的三个方法：

- loadSetting 、 - createGapView 、 - appUrl 。

先看 - loadSetting ，这里会对 wwwFolderName 和 startPage 设置默认值，代码如下：

```
- (void)loadSettings
{
    CDVConfigParser* delegate = [[CDVConfigParser alloc] init];

    [self parseSettingsWithParser:delegate];

    // Get the plugin dictionary, whitelist and settings from the delegate.
    self.pluginsMap = delegate.pluginsDict;
    self.startupPluginNames = delegate.startupPluginNames;
    self.settings = delegate.settings;

    // And the start folder/page.
    if(self.wwwFolderName == nil){
        self.wwwFolderName = @"www";
    }
    if(delegate.startPage && self.startPage == nil){
        self.startPage = delegate.startPage;
    }
    if (self.startPage == nil) {
        self.startPage = @"index.html";
    }

    // Initialize the plugin objects dict.
    self.pluginObjects = [[NSMutableDictionary alloc] initWithCapacity:20];
}
```

要看 - createGapView ，是因为这个方法内部先调用了一次 - appUrl ，所以关键还是 - appUrl 。源码如下：

```
- (NSURL*)appUrl
{
    NSURL* appURL = nil;

    if ([self.startPage rangeOfString:@"://"].location != NSNotFound) {
        appURL = [NSURL URLWithString:self.startPage];
    } else if ([self.wwwFolderName rangeOfString:@"://"].location != NSNotFound)
    {
        appURL = [NSURL URLWithString:[NSString stringWithFormat:@"%s/%s", self.w
wwFolderName, self.startPage]];
    } else if([self.wwwFolderName hasSuffix:@".bundle"]){
        // www folder is actually a bundle
        NSBundle* bundle = [NSBundle bundleWithPath:self.wwwFolderName];
        appURL = [bundle URLForResource:self.startPage withExtension:nil];
    } else if([self.wwwFolderName hasSuffix:@".framework"]){
        // www folder is actually a framework
        NSBundle* bundle = [NSBundle bundleWithPath:self.wwwFolderName];
        appURL = [bundle URLForResource:self.startPage withExtension:nil];
    } else {
        // CB-3005 strip parameters from start page to check if page exists in re
sources
        NSURL* startURL = [NSURL URLWithString:self.startPage];
        NSString* startFilePath = [self.commandDelegate pathForResource:[startURL
path]];

        if (startFilePath == nil) {
            appURL = nil;
        } else {
            appURL = [NSURL fileURLWithPath:startFilePath];
            // CB-3005 Add on the query params or fragment.
            NSString* startPageNoParentDirs = self.startPage;
            NSRange r = [startPageNoParentDirs rangeOfCharacterFromSet:[NSCharact
erSet characterSetWithCharactersInString:@"?#"] options:0];
            if (r.location != NSNotFound) {
                NSString* queryAndOrFragment = [self.startPage substringFromIndex
:r.location];
                appURL = [NSURL URLWithString:queryAndOrFragment relativeToURL:ap
pURL];
            }
        }
    }

    return appURL;
}
```

此时运行效果图：

#### **\*\* 加载远程HTML \*\***

项目里一般都是这种情况，接口返回H5地址，然后用网页加载H5地址。  
只需要设置下 `self.startPage` 就好了。

这里有几个需要注意的地方：

1. `self.startPage` 的赋值，必须在`[super viewDidLoad]`之前，否则`self.startPage` 会被默认赋值为`index.html`。
2. 需要在 `config.xml` 中修改一下配置，否则加载远程H5时，会自动打开浏览器加载。  
需要添加的配置是：

```
<allow-navigation href="https://*/*" />
<allow-navigation href="http://*/*" />
```

3. 远程H5中也要引用 `cordova.js` 文件。
4. 在 `info.plist` 中添加 `App Transport Security Setting` 的设置。

运行效果图：

### 3.创建插件， 配置插件

在插件中实现JS要调用的原生方法， 插件要继承自 CDVPlugin ， 示例代码如下：

```
#import "CDV.h"

@interface HaleyPlugin : CDVPlugin

- (void)scan:(CDVInvokedUrlCommand *)command;

- (void)location:(CDVInvokedUrlCommand *)command;

- (void)pay:(CDVInvokedUrlCommand *)command;

- (void)share:(CDVInvokedUrlCommand *)command;

- (void)changeColor:(CDVInvokedUrlCommand *)command;

- (void)shake:(CDVInvokedUrlCommand *)command;

- (void)playSound:(CDVInvokedUrlCommand *)command;

@end
```

配置插件， 是在config.xml的 widget 中添加自己创建的插件。  
如下图所示：

关于插件中方法的实现有几个注意点：  
1、如果你发现类似如下的警告：

```
THREAD WARNING: ['scan'] took '290.006104' ms. Plugin should use a background thread.
```

那么直需要将实现改为如下方式即可：

```
[self.commandDelegate runInBackground:^(
    // 这里是实现
)];
```

示例代码：

```
- (void)scan:(CDVInvokedUrlCommand *)command
{
    [self.commandDelegate runInBackground:^(
        dispatch_async(dispatch_get_main_queue(), ^{
            UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"原生弹窗"
            message:nil delegate:nil cancelButtonTitle:@"知道了" otherButtonTitles:nil, nil]
            ;

            [alertView show];

        })];
    }];
}
```

2、如何获取JS 传过来的参数呢？

CDVInvokedUrlCommand 参数，其实有四个属性，分别是 arguments 、 callbackId 、 className 、 methodName 。其中 arguments ，就是参数数组。看一个获取参数的示例代码：

```
- (void)share:(CDVInvokedUrlCommand *)command
{
    NSUInteger code = 1;
    NSString *tip = @"分享成功";
    NSArray *arguments = command.arguments;
    if (arguments.count < 3) {;
        code = 2;
        tip = @"参数错误";
        NSString *jsStr = [NSString stringWithFormat:@"shareResult('%@')",tip];
        [self.commandDelegate evalJs:jsStr];
        return;
    }

    NSLog(@"从H5获取的分享参数:%@",arguments);
    NSString *title = arguments[0];
    NSString *content = arguments[1];
    NSString *url = arguments[2];

    // 这里是分享的相关代码.....

    // 将分享结果返回给js
    NSString *jsStr = [NSString stringWithFormat:@"shareResult('%@','%@','%@')",title,content,url];
    [self.commandDelegate evalJs:jsStr];
}
```

3、如何将Native的结果回调给JS ？

这里有两种方式： 第一种是直接执行JS， 调用UIWebView 的执行js 方法。示例代码如下：

```
// 将分享结果返回给js
NSString *jsStr = [NSString stringWithFormat:@"shareResult('%@','%@','%@')",title,content,url];
[self.commandDelegate evalJs:jsStr];
```

第二种是，使用Cordova 封装好的对象 CDVPluginResult 和API。使用这种方式时，在JS 调用原生功能时，必须设置执行成功的回调和执行失败的回调。即设置 cordova.exec(successCallback, failCallback, service, action, actionArgs) 的第一个参数和第二个参数。像这样：

```
function locationClick() {
    cordova.exec(setLocation,locationError,"HaleyPlugin","location",[]);
}
```

然后，Native 调用JS 的示例代码：

```
- (void)location:(CDVInvokedUrlCommand *)command
{
    // 获取定位信息.....

    // 下一行代码以后可以删除
    //  NSString *locationStr = @"广东省深圳市南山区学府路XXXX号";
    NSString *locationStr = @"错误信息";

    //  NSString *jsStr = [NSString stringWithFormat:@"setLocation('%@')",locationStr];
    //  [self.commandDelegate evalJs:jsStr];

    [self.commandDelegate runInBackground:^(
        CDVPluginResult *result = [CDVPluginResult resultWithStatus:CDVCommandStatus_ERROR messageAsString:locationStr];
        [self.commandDelegate sendPluginResult:result callbackId:command.callbackId];
    )];
}
```

## 4.JS 调用Native 功能

终于到重点了，JS想要调用原生代码，如何操作呢？我用本地HTML 来演示。  
首先，HTML中需要加载 cordova.js ，需要注意该js 文件的路径，因为我的 cordova.js 与HTML放在同一个文件夹，所以src 是这样写：

```
<script type="text/javascript" src="cordova.js"></script>
```

然后，在HTML中创建几个按钮，以及实现按钮的点击事件，示例代码如下：

```
<input type="button" value="扫一扫" onclick="scanClick()" />
    <input type="button" value="获取定位" onclick="locationClick()" />
    <input type="button" value="修改背景色" onclick="colorClick()" />
    <input type="button" value="分享" onclick="shareClick()" />
    <input type="button" value="支付" onclick="payClick()" />
    <input type="button" value="摇一摇" onclick="shake()" />
    <input type="button" value="播放声音" onclick="playSound()" />
```

点击事件对应的关键的JS代码示例：

```
function scanClick() {
    cordova.exec(null,null,"HaleyPlugin","scan",[]);
}

function shareClick() {
    cordova.exec(null,null,"HaleyPlugin","share",['测试分享的标题','测试分享的内容','http://m.rblcmall.com/share/openShare.htm?share_uuid=shdfxdfdsfsdfs&share_url=http://m.rblcmall.com/store_index_32787.htm&imagePath=http://c.hiphotos.baidu.com/image/pic/item/f3d3572c11dfa9ec78e256df60d0f703908fc12e.jpg']);
}

function locationClick() {
    cordova.exec(setLocation,locationError,"HaleyPlugin","location",[]);
}

function locationError(error) {
    asyncAlert(error);
    document.getElementById("returnValue").value = error;
}

function setLocation(location) {
    asyncAlert(location);
    document.getElementById("returnValue").value = location;
}
```

JS 要调用原生，执行的是：



```
// successCallback : 成功的回调方法
// failCallback : 失败的回调方法
// server : 所要请求的服务名字, 就是插件类的名字
// action : 所要请求的服务具体操作, 其实就是Native 的方法名, 字符串。
// actionArgs : 请求操作所带的参数, 这是个数组。
cordova.exec(successCallback, failCallback, service, action, actionArgs);
```

cordova, 是 cordova.js 里定义的一个 var 结构体, 里面有一些方法以及其他变量, 关于exec, 可以看 iOSExec这个js 方法。

大致思想就是, 在JS中定义一个数组和一个字典(键值对)。

数组中存放的就是:

```
callbackId与服务、操作、参数的对应关系转成json 存到上面全局数组中。
var command = [callbackId, service, action, actionArgs];

// Stringify and queue the command. We stringify to command now to
// effectively clone the command arguments in case they are mutated before
// the command is executed.
commandQueue.push(JSON.stringify(command));
```

而字典里存的是回调, 当然回调也是与callbackId对应的, 这里的callbackId与上面的callbackId是同一个:

```
callbackId = service + cordova.callbackId++;
cordova.callbacks[callbackId] =
    {success:successCallback, fail:failCallback};
```

**\*\* iOSExec 里又是如何调用到原生方法的呢? \*\***

依然是做一个假的URL 请求, 然后在UIWebView的代理方法中拦截请求。

JS 方法 iOSExec 中会调用 另一个JS方法 pokeNative, 而这个 pokeNative, 看到他的代码实现就会发现与UIWebView 开启一个URL 的操作是一样的:

```
function pokeNative() {
    // CB-5488 - Don't attempt to create iframe before document.body is available
    if (!document.body) {
        setTimeout(pokeNative);
        return;
    }

    // Check if they've removed it from the DOM, and put it back if so.
    if (execIframe && execIframe.contentWindow) {
        execIframe.contentWindow.location = 'gap://ready';
    } else {
        execIframe = document.createElement('iframe');
        execIframe.style.display = 'none';
        execIframe.src = 'gap://ready';
        document.body.appendChild(execIframe);
    }
    failSafeTimerId = setTimeout(function() {
        if (commandQueue.length) {
            // CB-10106 - flush the queue on bridge change
            if (!handleBridgeChange()) {
                pokeNative();
            }
        }
    }, 50); // Making this > 0 improves performance (marginally) in the normal case (where it doesn't fire).
}
```

看到这里, 我们只需要去搜索一下拦截URL 的代理方法, 然后验证我们的想法接口。

我搜索 webView:shouldStartLoadWithRequest:navigationType 方法, 然后打上断点, 看如下的堆栈调用:

关键代码是这里，判断url 的scheme 是否等于 gap 。

```
        if ([[url scheme] isEqualToString:@"gap"]) {
            [vc.commandQueue fetchCommandsFromJs];
            // The delegate is called asynchronously in this case, so we don't have t
o use
            // flushCommandQueueWithDelayedJs (setTimeout(0)) as we do with hash chan
ges.
            [vc.commandQueue executePending];
            return NO;
        }
```

fetchCommandsFromJs 是调用js 中的 nativeFetchMessages()，获取 commandQueue 里的json 字符串；  
executePending 中将json 字符串转换为 CDVInvokedUrlCommand 对象，以及利用 runtime，将js 里的服务和 方法，转换对象，然后调用objc\_msgSend 直接调用执行，这样就进入了插件的对应的方法中了。

这一套思想与 WebViewJavascriptBridge 的思想很相似。

## 5. Native 调用 JS 方法

这个非常简单，如果是在控制器中，那么只需要像如下这样既可：

```
- (void)testClick
{
    // 方式一：
    NSString *jsStr = @"asyncAlert('哈哈啊哈')";
    [self.commandDelegate evalJs:jsStr];
}
```

这里的 evalJs 内部调用的其实是 UIWebView 的 stringByEvaluatingJavaScriptFromString 方法。

## 6.如果你在使用Xcode 8时，觉得控制台里大量的打印很碍眼，可以这样设置来去掉。

首先:

然后，添加一个环境变量：

好了，到这里关于Cordova 的讲解就结束了。

示例工程的github地址： JS\_OC\_Cordova ([https://link.jianshu.com?t=https://github.com/Haley-Wong/JS\\_OC/tree/master/JS\\_OC\\_Cordova](https://link.jianshu.com?t=https://github.com/Haley-Wong/JS_OC/tree/master/JS_OC_Cordova))

Have Fun!



Haley\_Wong (/u/a8cf6d63e889)

写了 79307 字，被 2076 人关注，获得了 1488 个喜欢  
(/u/a8cf6d63e889)写了 79307 字，被 2076 人关注，获得了 1488 个喜欢

+ 关注

Brighter! Brighter! Brighter!

♡ 喜欢 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-like-button) | 28



更多分享

(<http://cwb.assets.jianshu.io/notes/images/6046005>



下载简书 App ▶

随时随地发现和创作内容



(/apps/download?utm\_source=nbc)

被以下专题收入，发现更多相似内容



## Android使用Cordova进行混合开发 (/p/0a6b05abca0f?utm\_campaign=...

导语 这篇文章主要介绍了如何在Android平台上使用Cordova 的command-line interface (CLI)接口开发一个JS/HTML程序Cordova 提供了一组设备相关的API， 通过这组API， 移动应用能够以JavaScript访问原生的设

 jorstinchan (/u/a515733281e5?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## iOS开发最常用三方库集合OC版(三方库、插件、博客等等) (/p/4145a80de...

Swift版本[点击这里](#)欢迎加入QQ群交流: 594119878 最新更新日期: 17-11-01 About A curated list of iOS objective-C ecosystem. How to Use Simply presscommand+F+"xxx...

 ylgwhyh (/u/e94387132cd6?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/20e7fa4a0fb2?



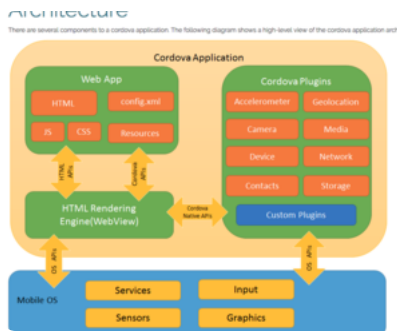
utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## Mac下使用Cordova在iOS应用中自定义插件的两种方式 (/p/20e7fa4a0fb2...

前言：本文主要是针对对webapp开发有一定基础和对Cordova有了一定的了解的人员，如果你是Cordova的初学者，建议先看看Cordova的相关介绍以及简单的一些CLI命令使用。Cordova的相关介绍以及安装推荐文

 LiuHJ (/u/92bbc62741d6?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/78e486b31953?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## iOS下JS与OC互相调用（七）--Cordova 基础 (/p/78e486b31953?utm\_c...

Cordova 简介 在介绍Cordova之前，必须先提一下PhoneGap。PhoneGap 是Nitobi软件公司2008年推出的一个框架，旨在弥补web 和iOS 之间的不足，使得web 和 iPhone SDK 之间的交互更容易。后来又加入了

 Haley\_Wong (/u/a8cf6d63e889?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

## iOS下JS与OC互相调用（五）--UIWebView + WebViewJavascriptBridge...

WebViewJavascriptBridge是一个有点年代的JS与OC交互的库，使用该库的著名应用还挺多的，目前这个库有7000+star。我去翻看了它的第一版本已经是4年前了，在版本V4.1.4以及之前，该库只有一个类和一个

 Haley\_Wong (/u/a8cf6d63e889?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/bdbd39effe20?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 一个人，拼的就是坚持.....（努力片段） (/p/bdbd39effe20?utm\_campai...

昨天临时决定想要开始修身养性，提高一下自己的阅读质量，从而让自己的精神境界有一个提高。想要摒弃一些自己不知是否正确的想法，却又不知道该如何选择。昨天匆匆忙忙的读了几页，却有了不小的收益。知

 依琳娜 (/u/04f64f79e26d?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)




(/p/84019a508c7a?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)  
**忘拉手刹“螳臂挡车”受伤！ 保险赔吗？ (/p/84019a508c7a?utm\_campaig...**

陈女士驾驶一辆蓝色奥拓回到家中，在自家门前的小巷内停好车后，就下车准备离开。刚没走出步就发现车子在倒退，她赶紧绕到车后，想用脚撑住地面，然后用身子顶住车子，想让车子停下，不料车子越溜越快，

 米米养车 (/u/b4690b60dea1?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

---

**【20170714电商资讯】 (/p/8c6bfe8c8354?utm\_campaign=maleskine...**

1、阿里巴巴股价半年大涨65%，五新战略巩固亚洲第一市值 美国当地时间周三，阿里巴巴股价收盘再创历史新高，报149美元，市值达到3768.7亿美元，亚洲第一的领先优势继续扩大。招商证券最新报告认为，阿

 A专业微博推广悍马 (/u/9018e7dbfe52?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

---

(/p/fe832dd820e9?



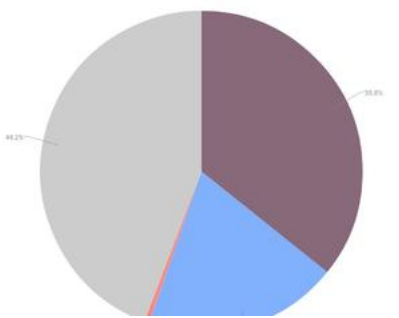
utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)  
**爱多深，苦涩就有多深 (/p/fe832dd820e9?utm\_campaign=maleskine&...**

深秋，踏香而去，菊花开得正欢，正艳，正妖娆。寻寻觅觅，不知不觉走到李清照纪念堂前。朴素的建筑，清新，低调。木匾上六个清秀流畅的书法，恰似宋词的清丽。门前两侧开遍了菊花，许是深秋的缘故吧，那

 金玉良缘传媒 (/u/89976e50b108?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

---

(/p/dc66057de591?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)  
**无题... (/p/dc66057de591?utm\_campaign=maleskine&utm\_content=n...**

2017年7月5日 周三 天气：小雨转阴 心情指数：❤ 秉着每日上来一文的心态，我勉强地又来了。今日不说闲话，只上来打个卡。任务依旧没完成，报名了精读班，坐等开课。我知道我状态没回来，和阿毛通了电

 Tera\_ (/u/f5898debeeb74?  
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)