

# 我是开发者

我只是个小开发仔 ~



首页 音乐 归档 留言 友链

## Git 忽略规则 .gitignore 配置建议, 拯救强迫症

by hext · 2018-04-16 · 默认分类

### .gitignore 文件

.gitignore 文件是用来描述 Git 忽略规则的文件, 用来防止你提交不必要的文件进入版本控制, 比如个人设置, 编译后的文件等, 一般来说只有源码才能加入版本控制.

### GitHub 的建议忽略规则

很多强迫症患者, 建完项目, 要创建 .gitignore 文件时, 就开始纠结了, 总觉得自己填入的规则不全面, 网上找的也不尽相同. 这时候不如我们看看 GitHub 的建议吧, 在 GitHub 有一个 .gitignore 文件模板集合:

<https://github.com/github/gitignore>

大概分为 语言的 和 环境的, 一级目录中的都是 各种语言的 gitignore, Global 目录下都是各种环境的全局 gitignore, 如系统的、和开发工具的忽略规则.

### iOS 项目的配置建议

以在 macOS 下使用 Xcode 开发 iOS 项目为例:

首先需要 按语言给项目配置 .gitignore 文件, 如 Objective-C 可参考: [Objective-C.gitignore](#) , Swift 可参考: [Swift.gitignore](#) . 在项目文件夹下新建 .gitignore 文件, 然后把上面对应语言的忽略规则复制进去保存即可.

然后需要配置全局的忽略规则, 忽略系统自动生成的一些文件或文件夹. 如 macOS 系统的 .DS\_Store 等, 可参考: [Global/macOS.gitignore](#) . 在用户目录①下新建 .gitignore\_global 文件, 然后把上面对应的忽略规则复制进去保存即可.

还有一类忽略规则, 用来忽略开发工具产生的一些文件, 一般是能自动生成的文件, 比如一些编译后的二进制文件, 或者是用户个人相关的偏好设置文件, 这些都不用提交到版本控制. Xcode 可参考: [Xcode.gitignore](#) . 开发工具的忽略规则可以写到全局忽略文件中, 也可以写到具体项目的忽略文件中. 但是为了防止你配置到了全局忽略文件, 而协作者没有配置, 然后往仓库提交了不必要文件的问题, 所以大多人都习惯将开发工具的相关忽略规则写到具体项目的忽略文件中②.

注:

①:

macOS 系统的用户目录为: /Users/你的用户名

Windows 系统的用户目录为: C:\Users\你的用户名

② [Objective-C.gitignore](#) 或者 [Swift.gitignore](#) 文件中已经同步包含了 [Xcode.gitignore](#) 的忽略规则, 所以可以不用关心这个文件

[阅读全文 »](#)

## ijkplayer 的编译、打包 framework 和 https 支持

by hext · 2018-04-10 · 默认分类

[ijkplayer](#) 是 [bilibili](#) 开源的一款优秀的播放器, 基于 FFmpeg, 支持 iOS/Android, 点播/直播, 以及多种编码. 官方并不直接提供 framework 框架包, 这样大家可以根据需要选择配置, 自己编译并打包. 但是也让很多第一次使用的小伙伴一脸懵逼, 并且默认配置只支持 http, 如果想播放 https 链接的视频, 那么需要做额外的配置

本文为 iOS 的编译打包过程, 环境为 Mac + Xcode

## 准备工作

安装 homebrew, git, yasm. (如果已经安装好可以跳过, 不清楚的再来一遍也无妨)

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
brew install git
brew install yasm
```

## 获取 ijkplayer 源码

在一个合适的位置新建一个文件夹, 假设为桌面, 文件夹名为 ijkplayer.

打开终端, 输入下面的指令

```
# 进入到刚刚新建的文件夹内
cd ~/Desktop/ijkplayer/

# 获取ijkplayer源码
git clone https://github.com/Bilibili/ijkplayer.git ijkplayer-ios

# 进入源码目录
cd ijkplayer-ios

# 切换分支 (目前为k0.8.8, 可以自行去GitHub查看最新版本号)
git checkout -B latest k0.8.8
```

## 配置编解码器格式支持

默认为最少支持, 如果足够你使用, 可以跳过这一步. 否则可以改为以下配置:

- module-default.sh 更多的编解码器/格式

- `module-lite-hevc.sh` 较少的编解码器/格式(包括hevc)
- `module-lite.sh` 较少的编解码器/格式(默认情况)

```
# 进入 config 目录
```

```
cd config
```

```
# 删除当前的 module.sh 文件
```

```
rm module.sh
```

```
# 可根据需要替换为`module-default.sh`, `module-lite-hevc.sh`, `module-lite.sh`
```

```
# 创建软链接 module.sh 指向 module-lite-hevc.sh
```

```
ln -s module-lite-hevc.sh module.sh
```

```
cd ..
```

```
cd ios
```

```
sh compile-ffmpeg.sh clean
```

## 获取 ffmpeg 并初始化

```
cd ..
```

```
./init-ios.sh
```

## 添加 https 支持

最后会生成支持 https 的静态文件 `libcrypto.a` 和 `libssl.a`, 如果不需要可以跳过这一步

```
# 获取 openssl 并初始化
```

```
./init-ios-openssl.sh
```

```
cd ios
```

```
# 在模块文件中添加一行配置 以启用 openssl 组件
```

```
echo 'export COMMON_FF_CFG_FLAGS="$COMMON_FF_CFG_FLAGS --enable-openssl"' >> ../config/module.sh
```

```
./compile-ffmpeg.sh clean
```

## 编译

```
# 如果下一步提示错误`xcrun: error: SDK "iphoneos" cannot be located`, 请执行`sudo xcode-select --switch /Applications/Xcode.app/Contents/Developer/`, 再重新执行下一步
```

```
# 编译openssl, 如果不需要https可以跳过这一步  
./compile-openssl.sh all
```

```
# 编译ffmpeg  
./compile-ffmpeg.sh all
```

ps: 如果提示错误:

```
./libavutil/arm/asm.S:50:9: error: unknown directive  
    .arch armv7-a  
    ^  
make: *** [libavcodec/arm/aacpsdsp_neon.o] Error 1
```

最新的 Xcode 已经弱化了对 32 位的支持, 解决方法:

在 compile-ffmpeg.sh 中删除 armv7 , 修改如:

```
FF_ALL_ARCHS_IOS8_SDK="arm64 i386 x86_64"
```

再重新执行出现错误的命令: ./compile-ffmpeg.sh all

## 打开 IJKMediaPlayer 项目

用命令:

```
open IJKMediaPlayer/IJKMediaPlayer.xcodeproj
```

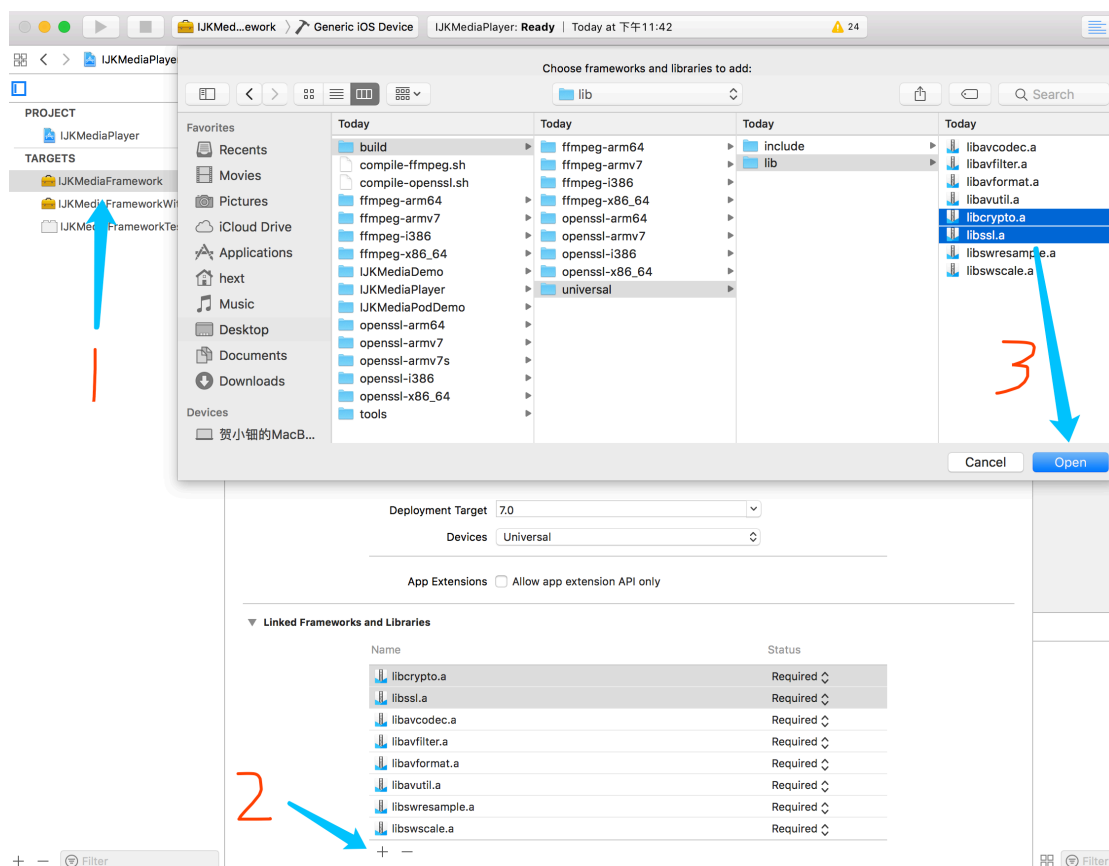
或者手动用 Xcode 打开 ios 目录下的 IJKMediaPlayer 项目.

添加 openssl 相关包以支持 https

如果不使用 https, 可以跳过此步, 直接开始打包 framework

如果使用 https, 那么需要手动给 IJKMediaFramework 添加 libcrypto.a 和 libssl.a 文件, 默认不会添加

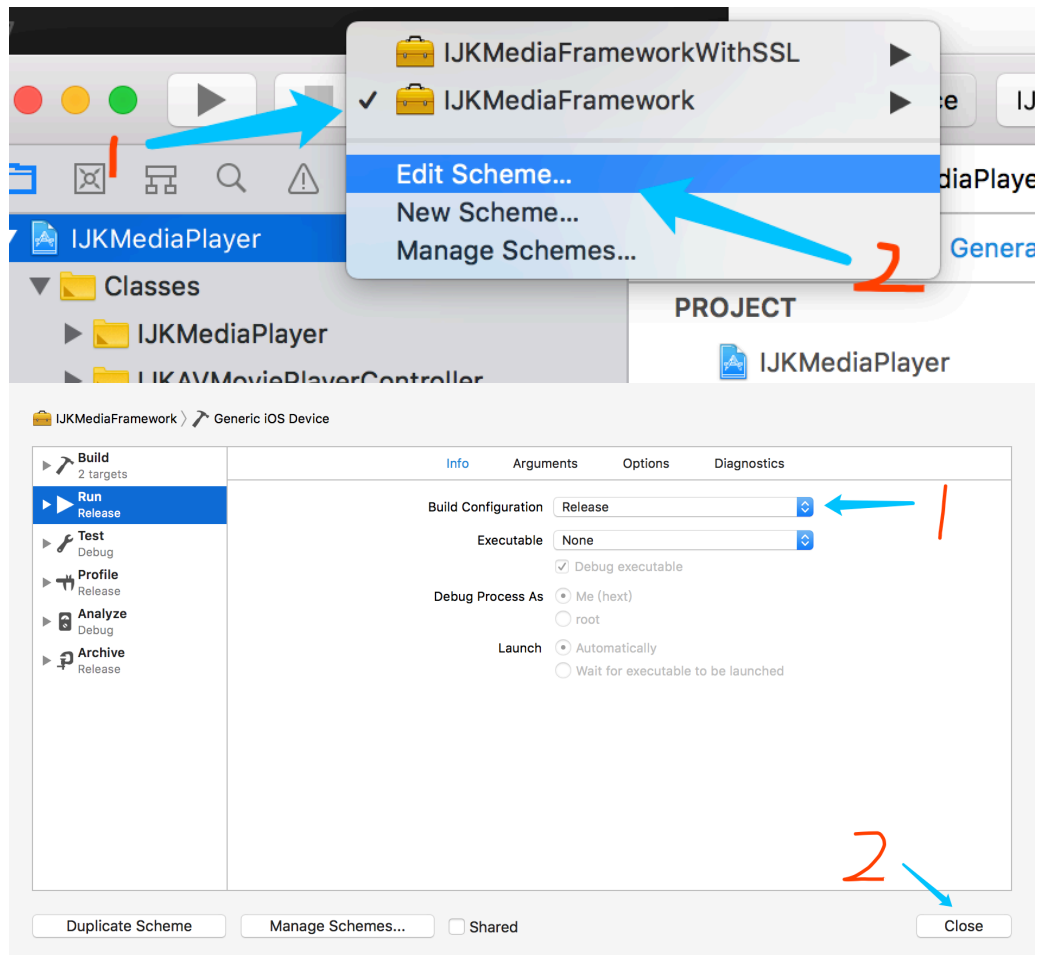
ps: 这两个依赖库的目录为: ijkplayer-ios/ios/build/universal/lib, 只有进行了上面跟 openssl 相关的操作, 才会在这个目录下有生成 libcrypto.a 和 libssl.a



## 打包 framework

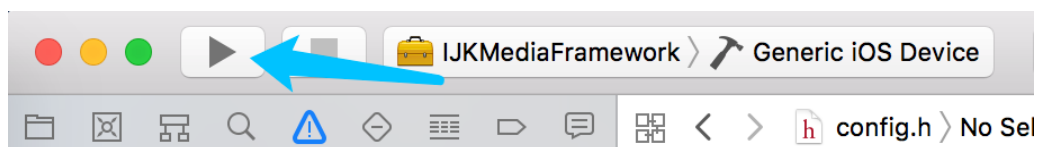
大家会发现除了 IJKMediaFramework 这个 target, 还有一个叫 IJKMediaFrameworkWithSSL, 但是不推荐使用这个, 因为大部分基于 ijkplayer 的第三方框架都是使用的前者, 你把后者导入项目还是会报找不到包的错误, 就算你要支持 https 也推荐使用前者, 然后按照上一步添加 openssl 即可支持

### 1. 配置 Release 模式如果下图



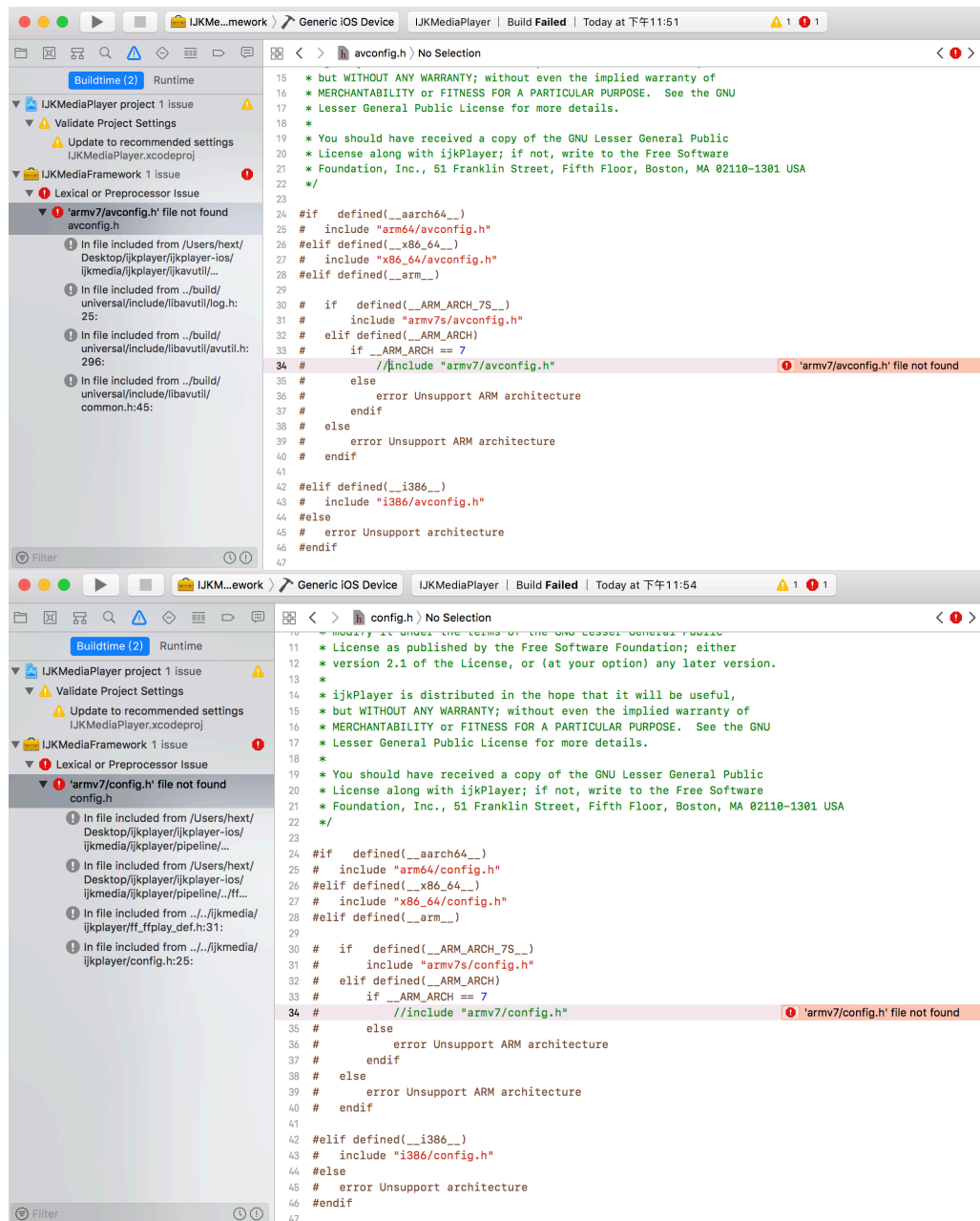
## 2. 打包真机 framework

选择你连接的手机或者 Generic iOS Device

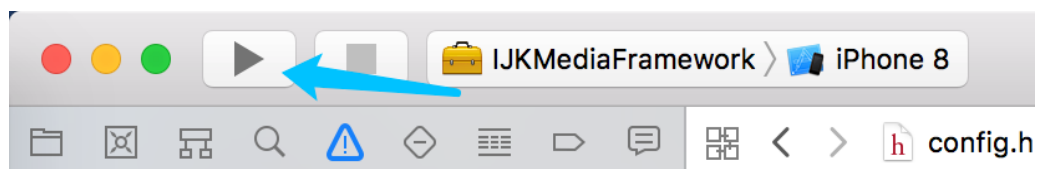


如图操作，然后按键 `command+b` 编译即可

如果之前的步骤删除了 `compile-ffmpeg.sh` 中 `armv7`，这里会报错，我们直接注释掉就好



### 3. 打包模拟器 framework



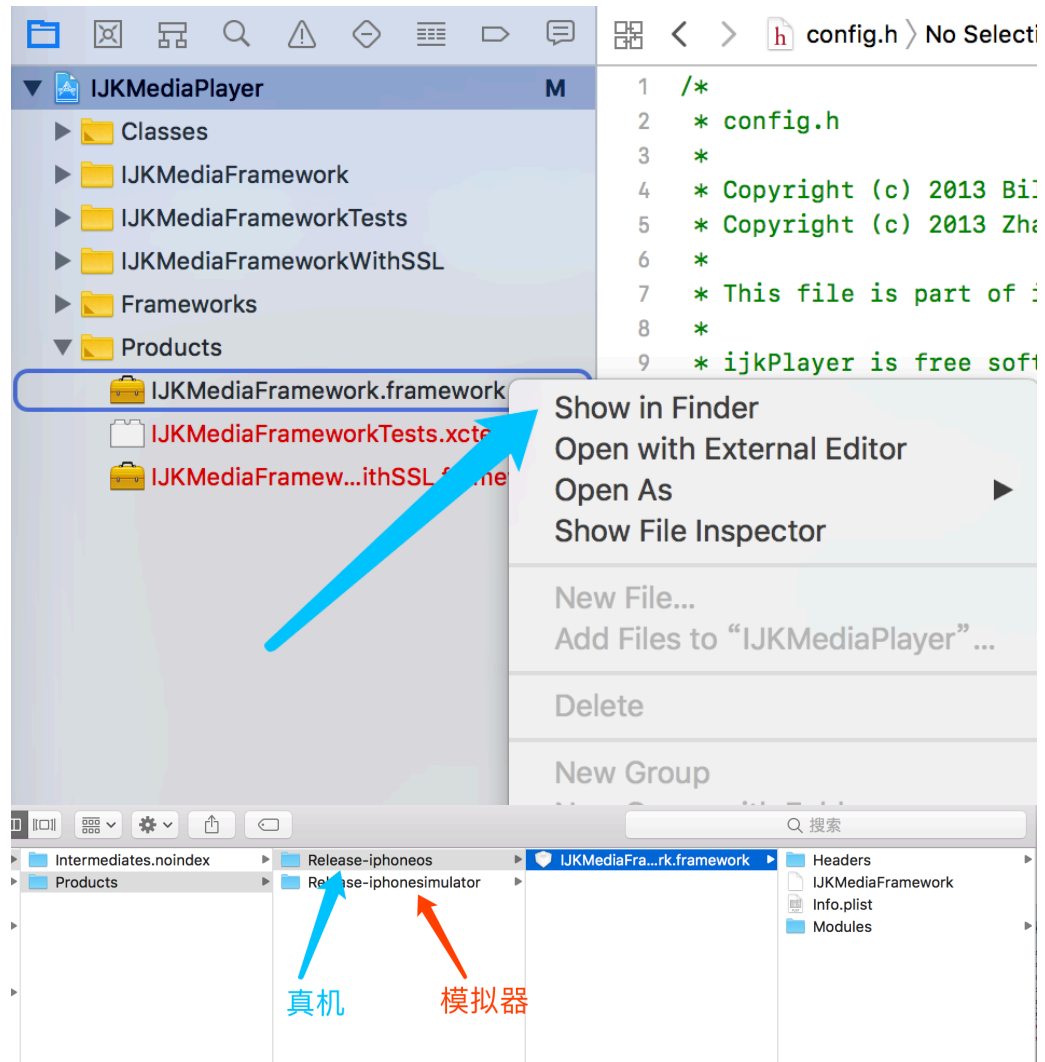
如图操作，然后 `command+b` 编译即可

### 4. 合并 framework

如果只需要真机运行或者模拟器运行, 可以不用合并, 直接找到对应的 framework 导入项目即可; 一般我们为了方便会合并 framework, 这样就同时支持模拟器和真机运行。



先找到生成 framework 的目录:



准备合并:

打开终端, 先 cd 到 Products 目录下

然后执行:

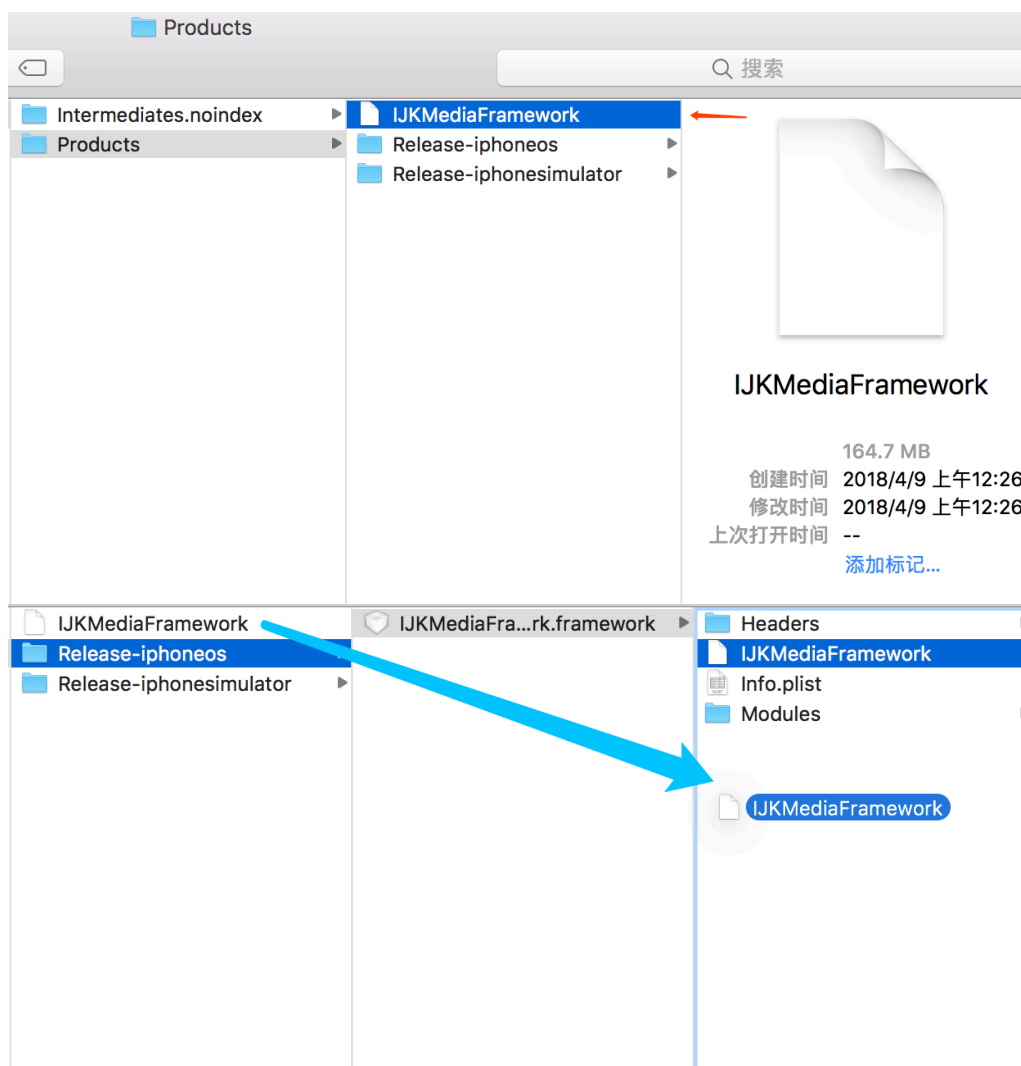
```
lipo -create 真机framework路径 模拟器framework路径 -output 合并的文件路径
```

```
lipo -create Release-iphoneros/IJKMediaFramework.framework/IJKMediaFramework Release-iphonesimulator/IJKMediaFramework.framework/IJKMediaFramework -output IJKMediaFramework
```

合并完成:

可以看到这里生成了一个大概两倍大小的文件, 将生成的 IJKMediaFramework 文件替换掉 真机framework 中的

IJKMediaFramework 文件，然后这个替换掉文件的真机framework 就是我们需要的 通用的framework 了。



## 集成 framework 到项目中

- 导入 framework

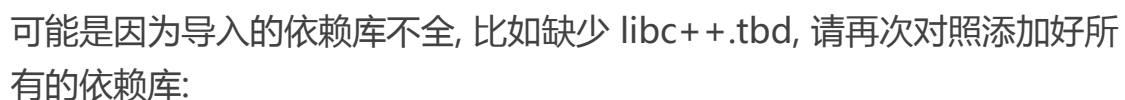
直接将 IJKMediaFramework.framework 拖入到工程中即可

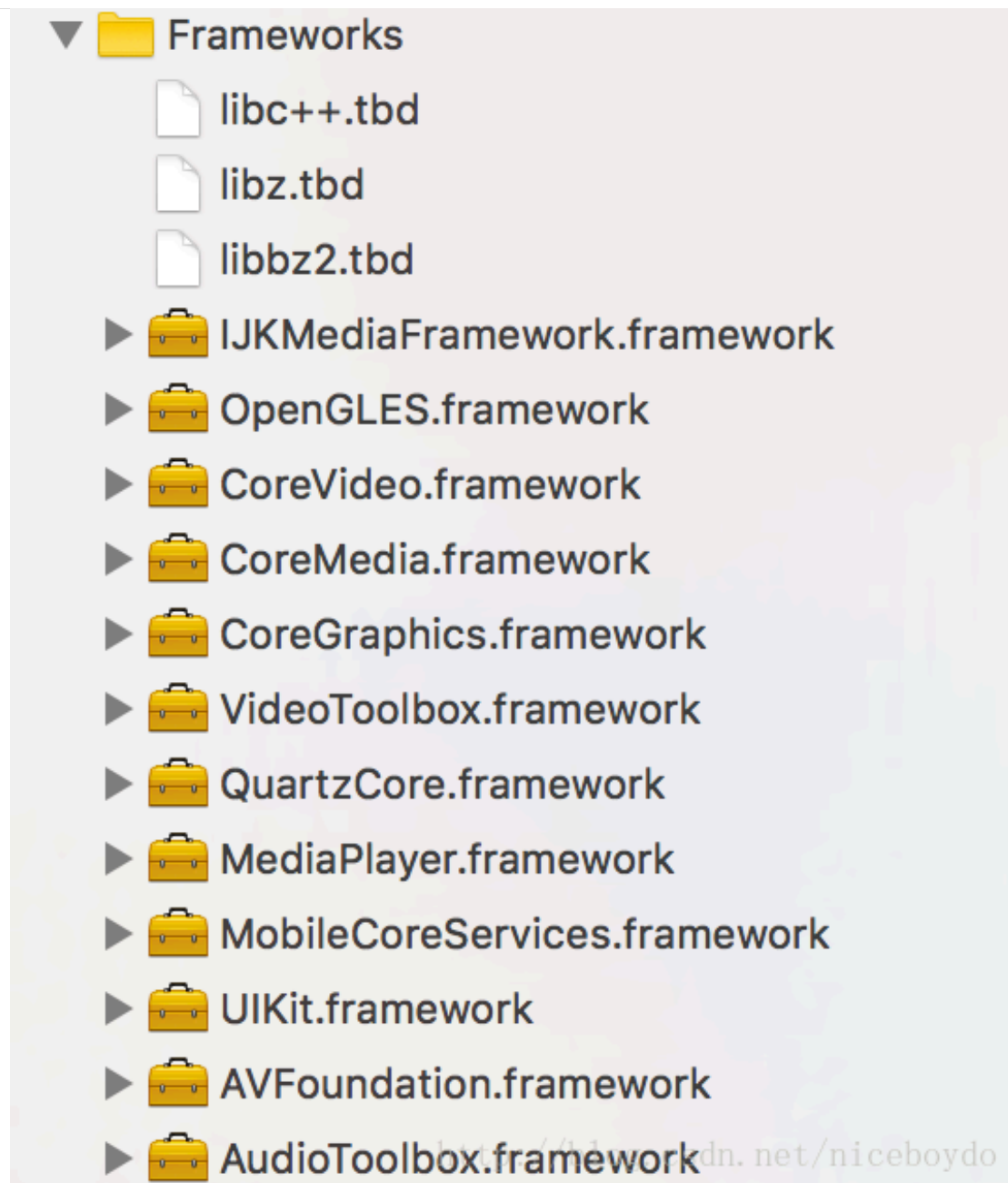
注意记得勾选 Copy items if needed 和 对应的 target

- 添加下列依赖到工程

1. libc++.tbd ( 编译器选 gcc 的请导入 libstdc++.tbd )
2. libz.tbd
3. libbz2.tbd
4. AudioToolbox.framework

- 导入 ijkplayer 头文件运行一下项目, 如果遇到了类似这样的错误:





ios   ijkplayer

[阅读全文 »](#)

## 使用 Python 把多个 MP4 合成一个视频

by hext · 2018-02-03 · 默认分类

这两天群里有个小伙伴有一个需求,就是把很多个视频文件 合并成一个. 期间也找了各种软件,如格式工厂,但是只能一次合成50个文件,小伙伴有几千个文件需要合成,太繁琐;又比如会声会影,这个剪辑是很强大,但是软件

也很大, 对电脑配置要求也高. 我只需要拼接功能, 割鸡焉用牛刀?

## 人生苦短 我用 Python

这两天刚好在学习用 Python 写自动化测试. 💡 转念一想, Python 也很擅长图形处理, 那处理视频也不在话下吧, 于是就上网搜了搜, 果然找到了简单的办法~

### 开始安装使用

主要是利用 moviepy 这个库, 里面提供了丰富的功能, 我们只需要使用简单的拼接函数

python

[阅读全文 »](#)

## Mac + iPhone 实现微信跳一跳自动化教程

by hext · 2018-01-28 · 默认分类

### 前言

最近微信跳一跳这个小游戏火爆, 由于朋友排行版的存在, 激发了我们的上(pan)进(bi)心. 除了正常方式爆肝提高分数以外, 也有各种技术宅的非正常手段出来:

1. 抓包拦截请求, 直接改游戏结束之后, 提交到服务器的分数
2. 物理外挂, 用机械手指和摄像头, 识别距离, 计算点击时间, 控制机械手指按下去的时间
3. 利用手机调试模式, 屏幕截图, 识别方块距离, 计算点击时长, 调用系统函数模拟点击屏幕

第 3 种方法相当于第 2 种方法的软件版, 本文用的就是这种方式

*ps: 目前方法 1 的提交漏洞已经被修复, 方法 3 特别是方法 2 原理上是模*

拟人的操作 (代码中有故意随机偏移落地点和停顿间隔), 理论上微信是没法区别的, 但是微信貌似有一套成长曲线检测方法, 如果你的进步曲线不科学, 还是会把你分数清零的

ios    跳一跳    自动化    python    iphone

[阅读全文 »](#)

## 夕甲甲——孔乙己之C++版

by hext · 2017-12-13 · 默认分类

OOP代码的格局，是和别的编程模式不同的：首先要有一个构造函数；基类里只定义了函数的形式，可以随时通过派生增加不同的实现。那些程序员们，每每学会了继承和多态，便可以接一个项目，——这是十年前的事，现在至少要懂得设计模式，——靠柜台外站着，写些代码换一个鸡蛋灌饼吃；倘若懂一些多线程的知识，便可以在鸡蛋灌饼里加火腿肠，或者里脊了，如果懂得语言的实现原理，那就能做一些底层代码的优化，换一盘炒菜。但这些程序员，多是初级水平，大抵没有这样深入。只有懂得内存模型的，才能接一些底层的项目，打开调试器和剖析器，慢慢的坐着优化代码。

我从十二岁起，便在村口的githuboverflow工厂里当学徒，师傅说，我逻辑思维太差，怕学不会静态类型语言，就写些shell脚本吧。shell脚本虽然容易写，但语法容易含混不清的地方也不少。字符串和数组的功能很弱，条件判断的语法也很奇特，有时要对特殊字符手动做两重转义，然后通过。在这种情况下，写脚本也很为难。所以过了几天，师傅又说我干不了这事。幸亏荐头的情面大，辞退不得，便改为专管测试的一种无聊职务了。

[阅读全文 »](#)

>

© 2018 我是开发者.  
湘ICP备15002355号

Powered by Typecho. Theme Is  
Terse. 