

02

OPEN ORIENTED

凹凸实验室

# React Native 原生应用打包与发布

Mark



- ▶ iOS 应用打包和发布
- ▶ Android 应用打包和发布

- ▶ iOS 应用打包和发布
- ▶ Android 应用打包和发布

▸ iOS 应用打包和发布



# 打包

Xcode 概念

Project / Workspace / Target / Scheme...

环境变量

debug / release

打包命令

# Xcode 概念



Aurora.xcodeproj

## Project

iOS 项目工程文件

包含 APP 所有文件资源位置的文件夹



Aurora.xcworkspace

## Workspace

多个 Project 的集合

包括 APP 代码和依赖库文件信息



Aurora.xcodeproj

## Project

一般情况下使用 project 文件



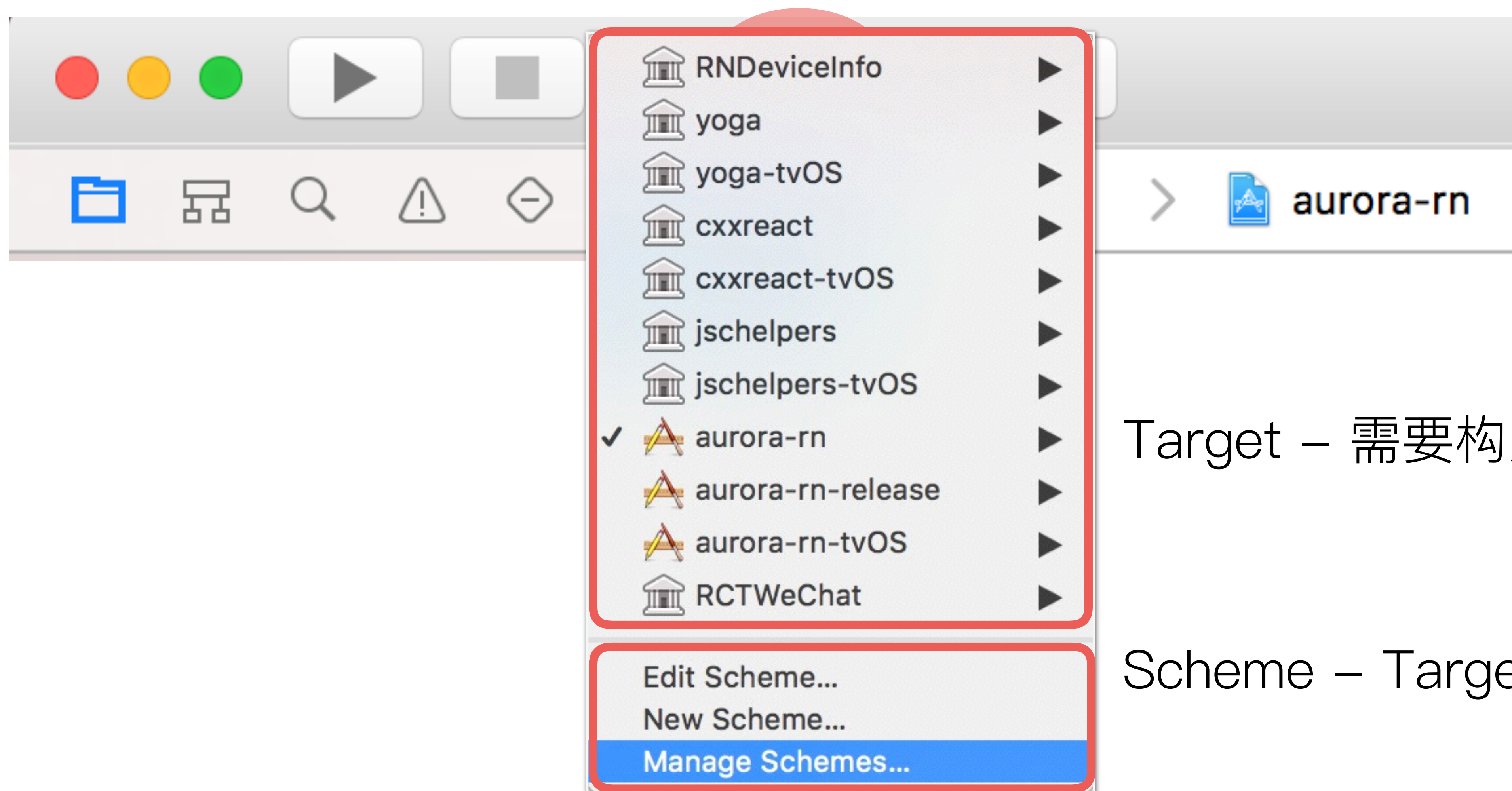
Aurora.xcworkspace

## Workspace

CocoaPods 项目需要使用 workspace 文件

注：CocoaPods 是第三方依赖库管理工具，类似于 NPM





Target – 需要构建的项目

Scheme – Target 构建配置

## Xcode 环境变量

## Debug

- 包含程序调试信息
- 可执行断点等调试指令
- RN 会显示警告信息

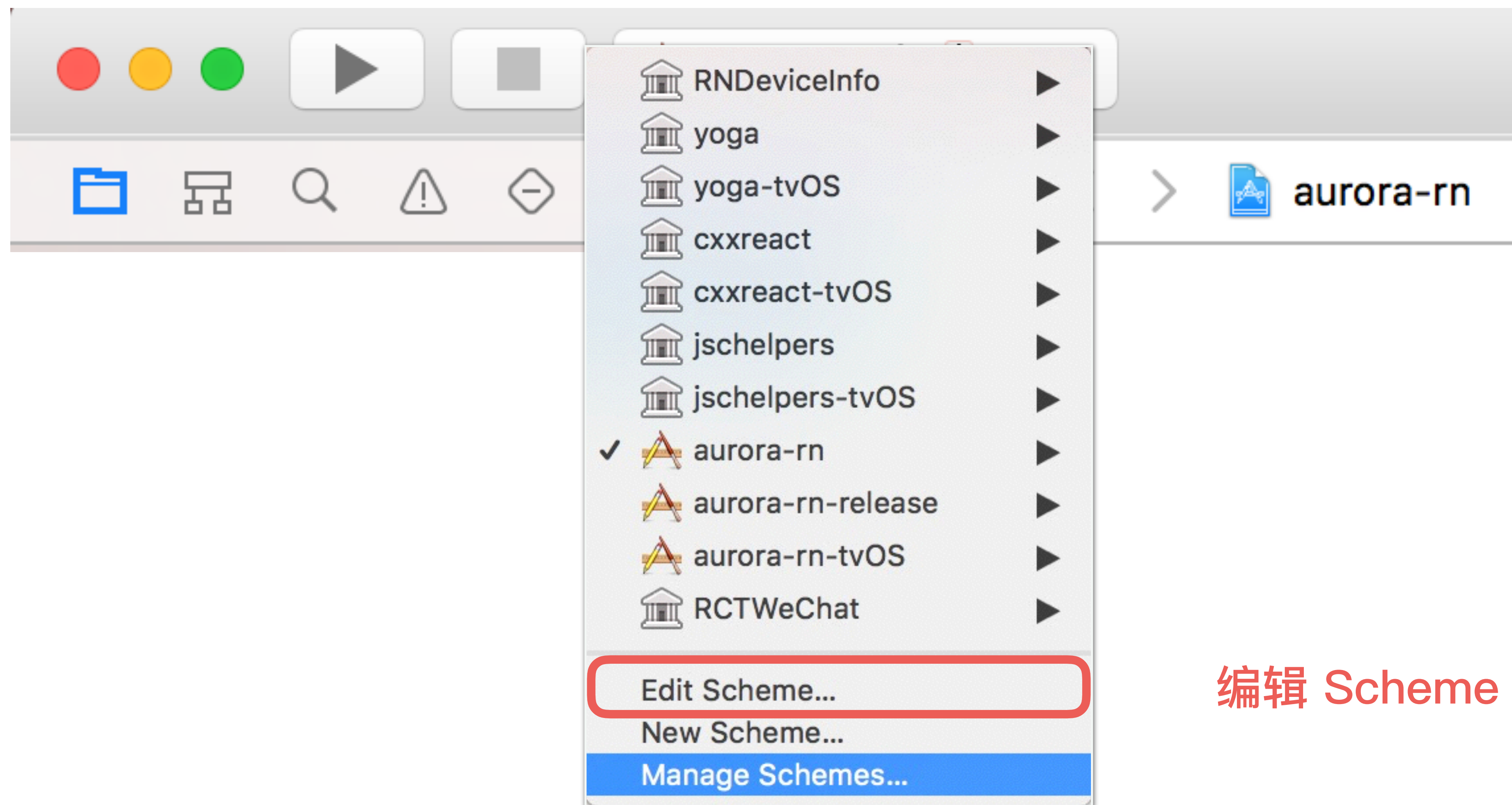
## Release

- 不包含任何调试信息
- 压缩体积小、运行速度快
- RN 无法实时更新，需重新编译

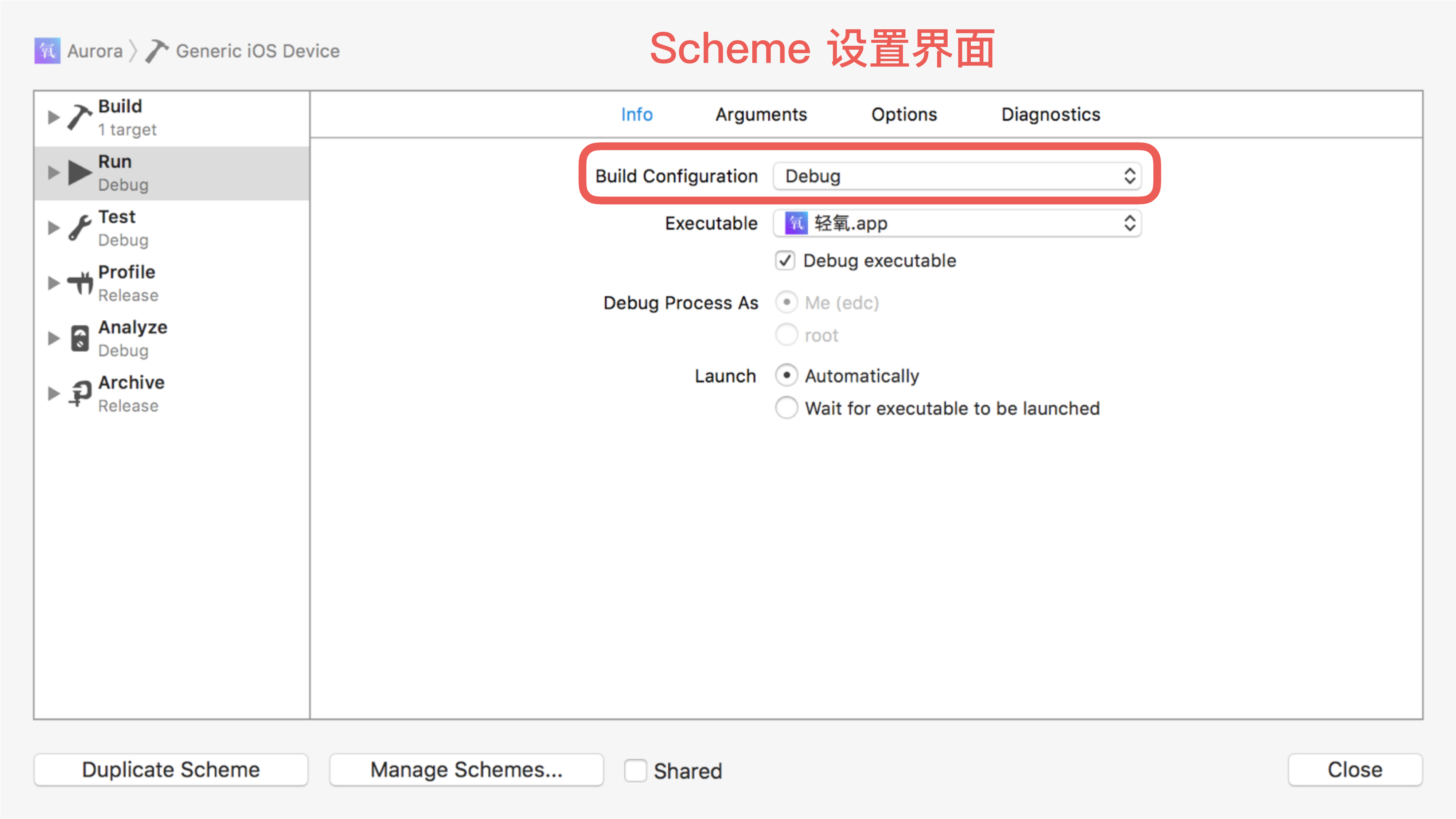


# 环境变量切换

02



编辑 Scheme



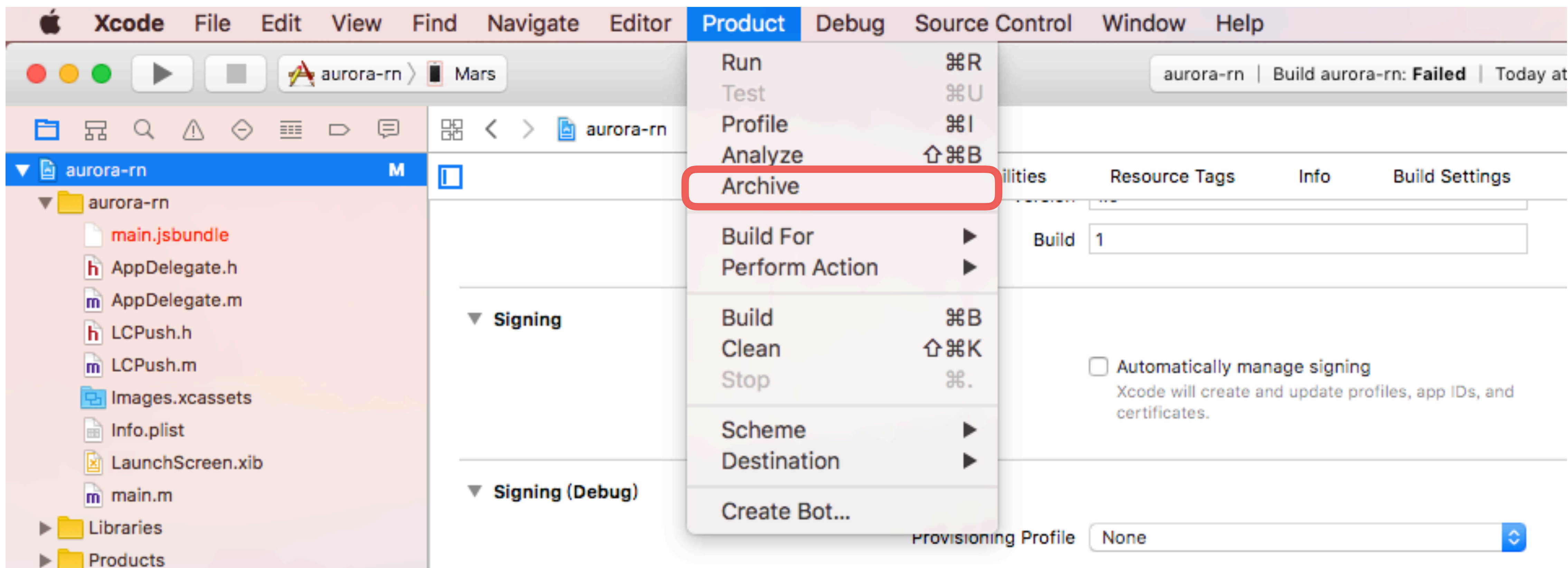


# Xcode 打包

# Xcode 打包

02

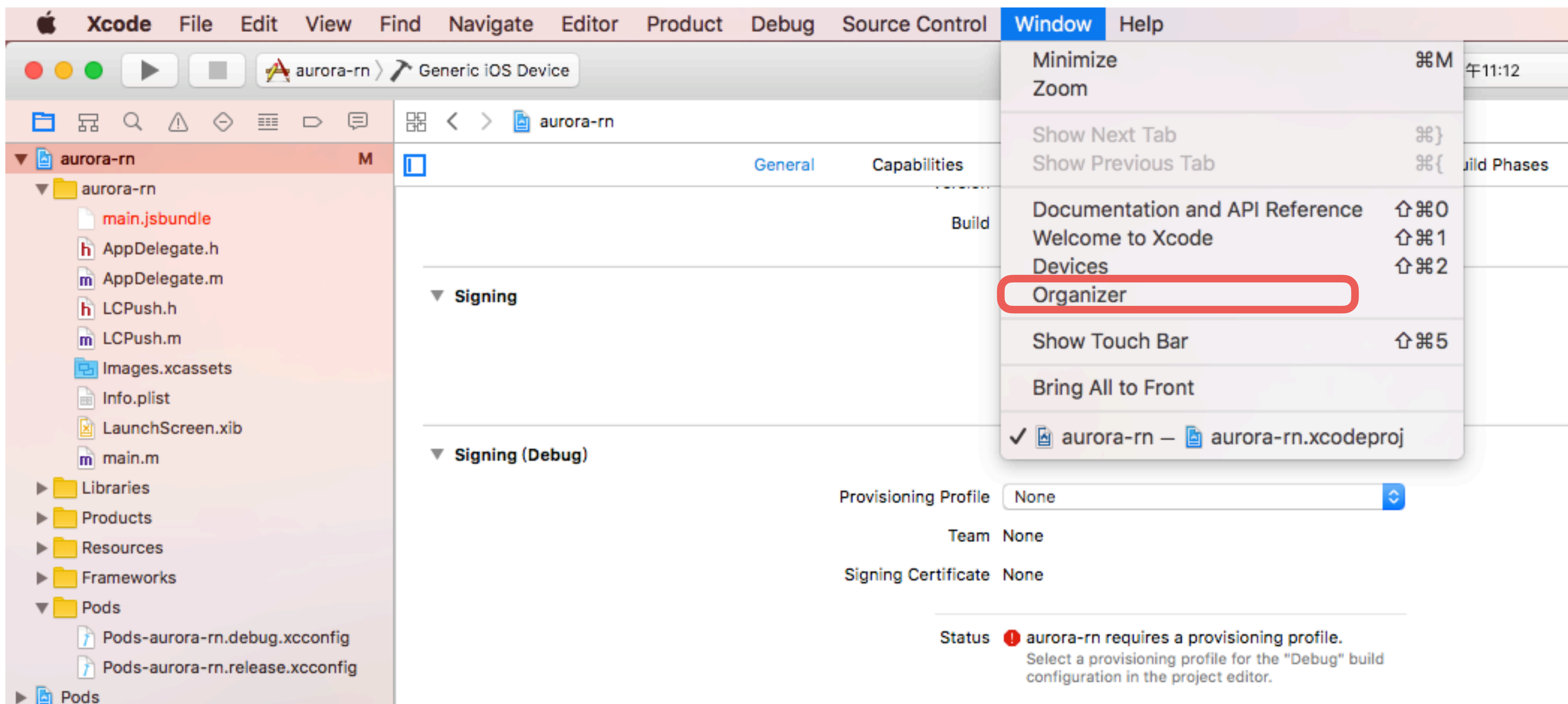
## 1. 使用 Archive 命令生成 archive 包



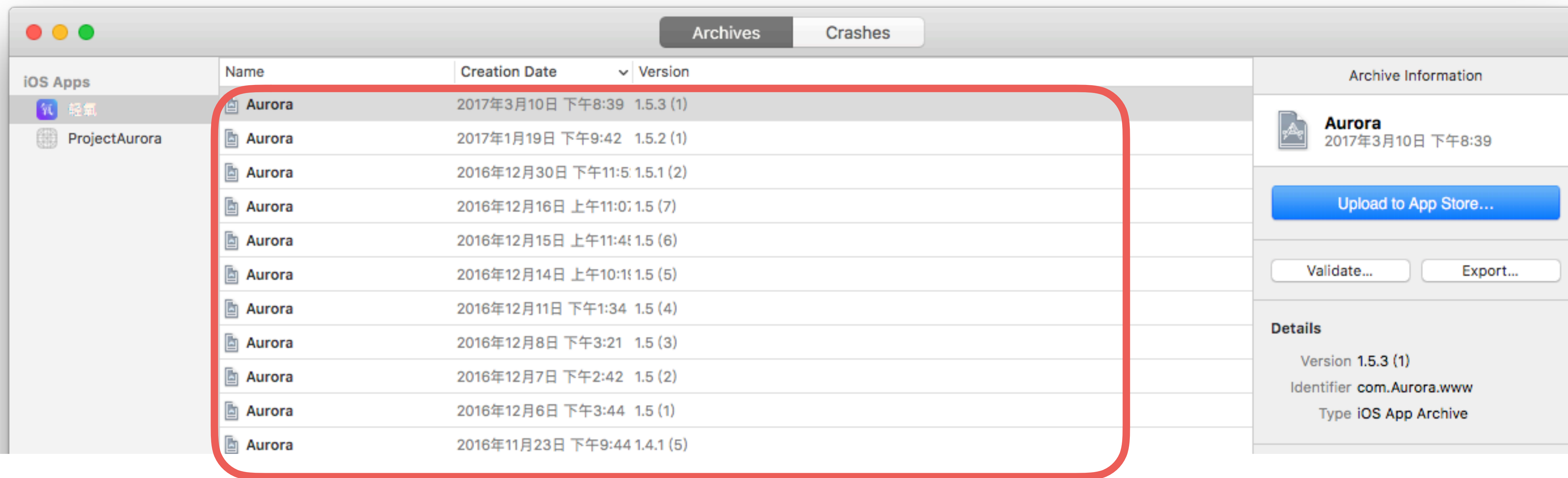
# Xcode 打包

02

## 2. 打开 Organizer, 查看打包记录



## 3. 根据生成时间找到对应的打包结果



发布到 App Store



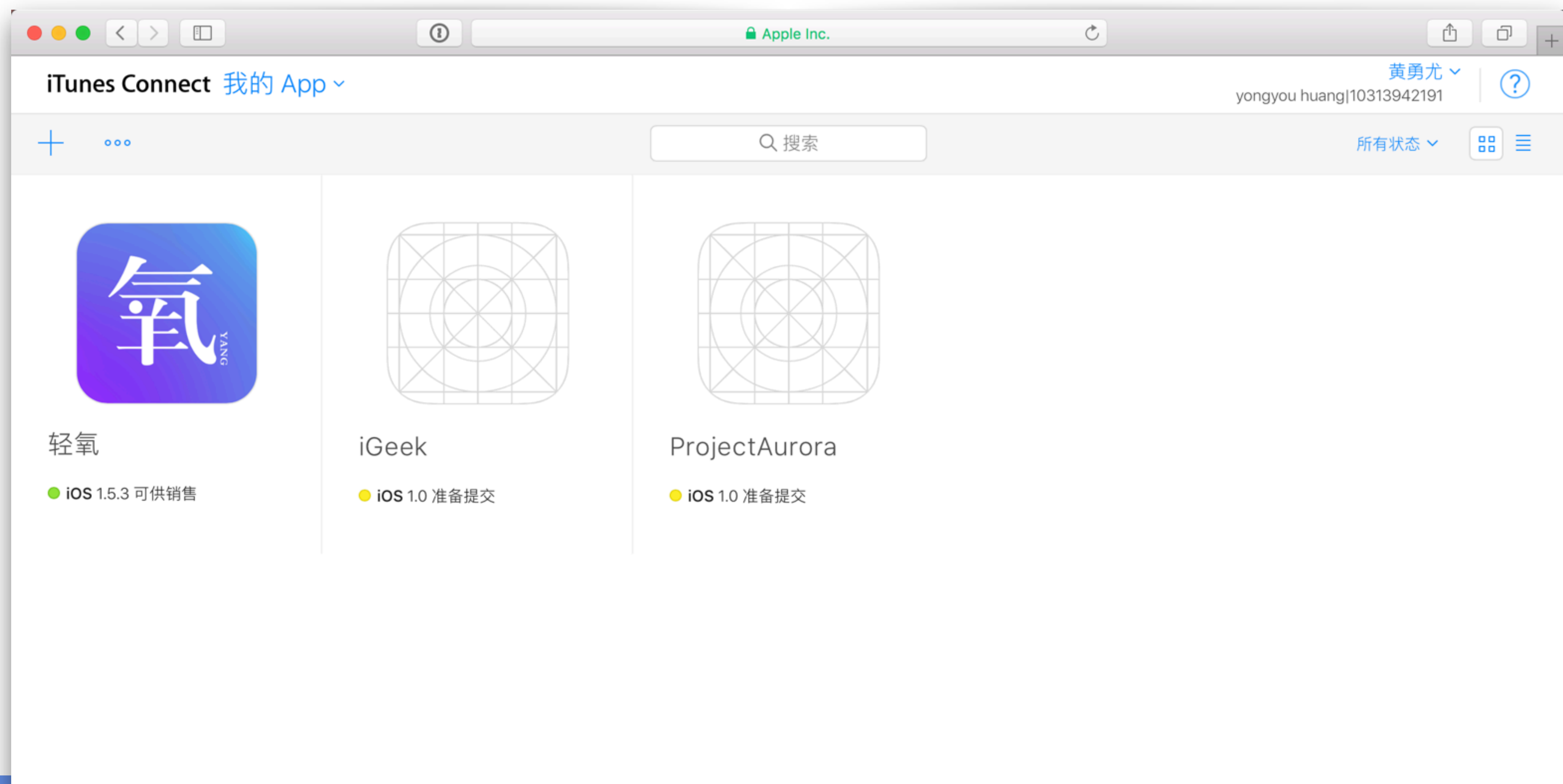
## Apple 开发者账号

<https://developer.apple.com>

- \* 免费开发者
- \* 个人及公司组织    \$99/y
- \* 企业版                \$299/y

iTunes Connect <https://itunesconnect.apple.com>

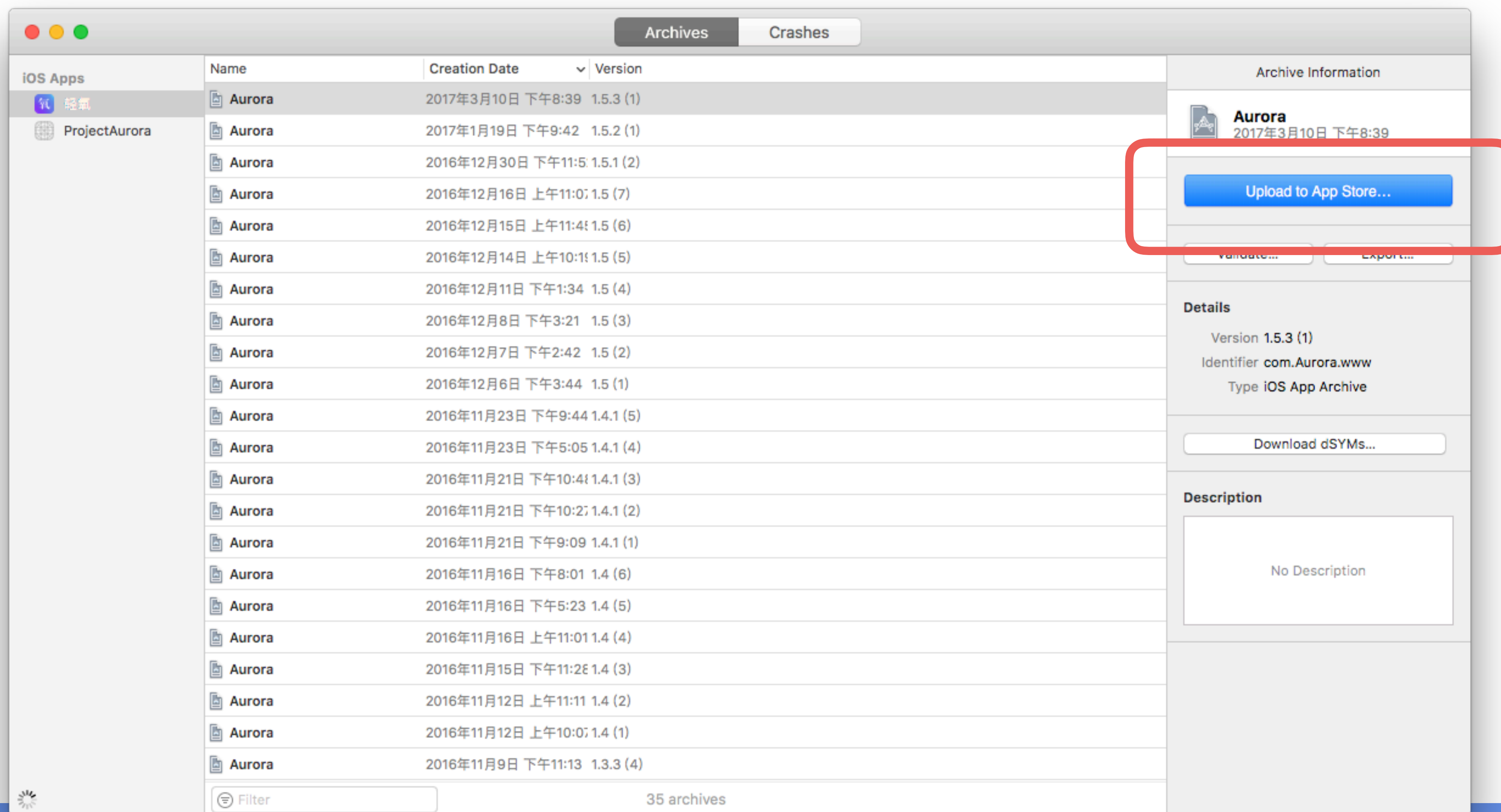
创建应用 · 发布版本 · 管理测试

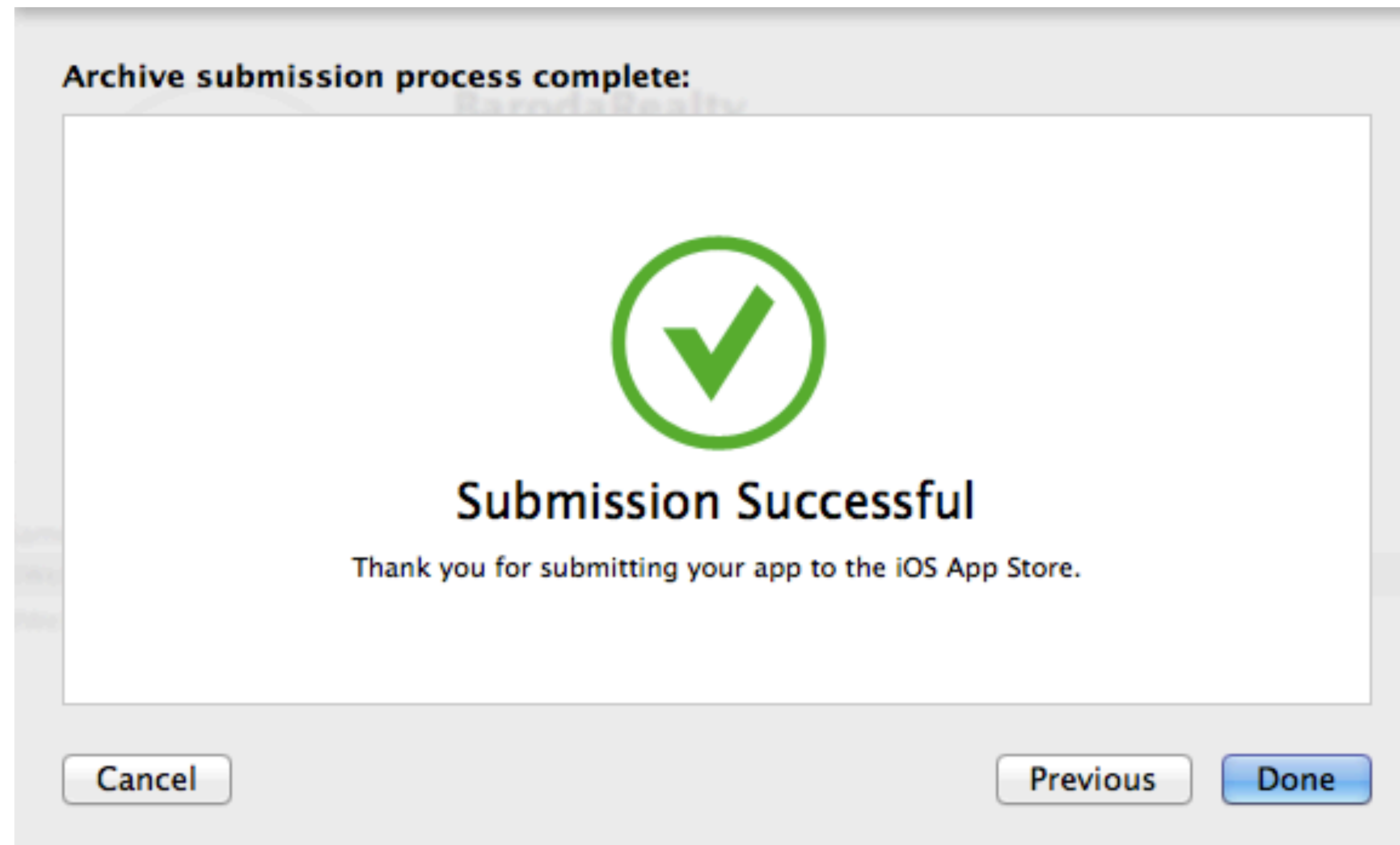


# 发布到 App Store

02

在 Organizer 中选择打包记录，一键上传至 App Store





## 真机测试

单机测试

Xcode

不需提前打包

公开测试

TestFlight

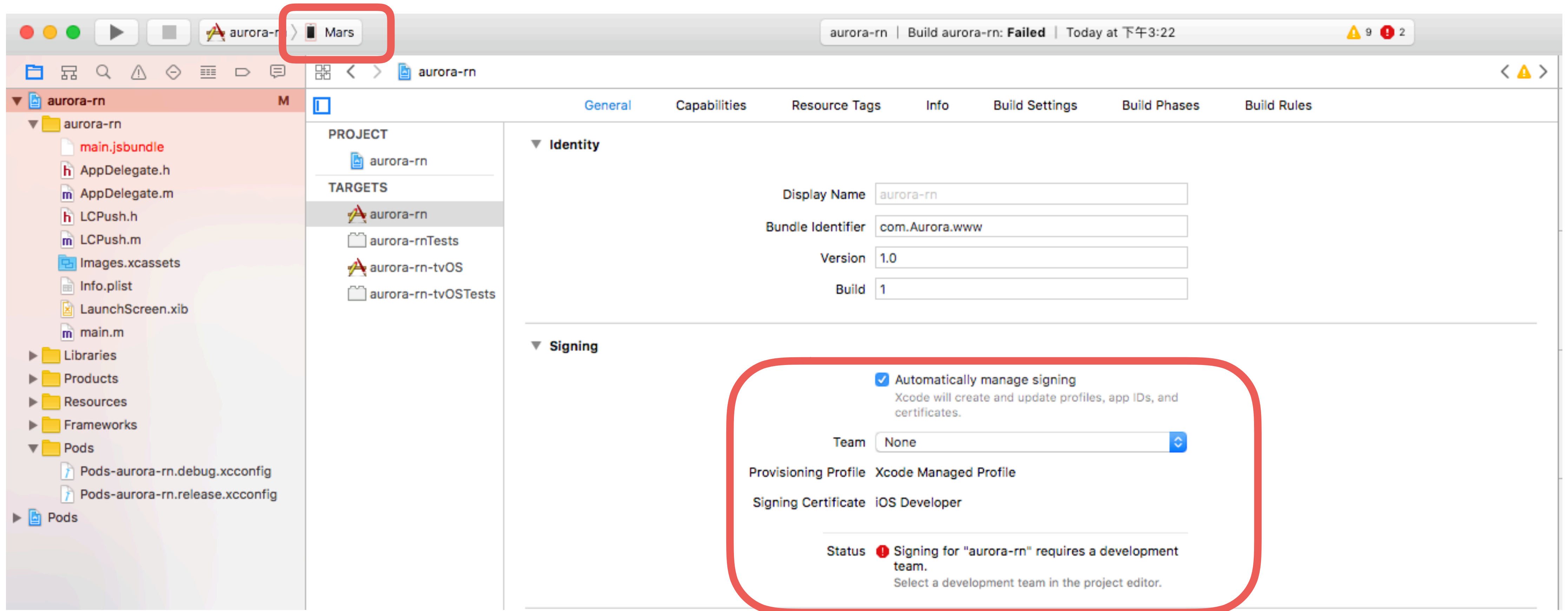
第三方测试平台



# 单机测试

02

- ▶ Xcode: Product -> Run



▶ TestFlight 内部测试



正在测试

测试信息

内部测试

外部测试

TESTFLIGHT 构建版本

iOS

关于 TestFlight 数据

内部测试

在您的公司内使用 TestFlight App 测试构建版本。Beta 版 App 无需审核。

iOS

选择版本进行测试

内部测试员 (4)

编辑

电子邮件	姓名
b[redacted]@qq.com	WongLevin
z[redacted]h@qq.com	朱奕腾

▶ TestFlight 外部测试



● ● ● < > ⓘ Apple Inc. ⓘ ⓘ +

iTunes Connect 我的 App ▾ 轻氧 ▾ yongyou huang|10313942191 黄勇尤 ▾ ⓘ

App Store 功能 TestFlight 活动 App 分析 销售和趋势

正在测试

测试信息

内部测试

外部测试

TESTFLIGHT 构建版本

iOS

关于 TestFlight 数据 ⓘ

外部测试 ⓘ

让使用 TestFlight App 的任何人测试您的构建版本。构建版本可能需要 Beta 版 App 审核部门的批准。

开始测试

iOS 向测试添加构建版本

外部测试员 (5) ⓘ

Q 搜索

编辑

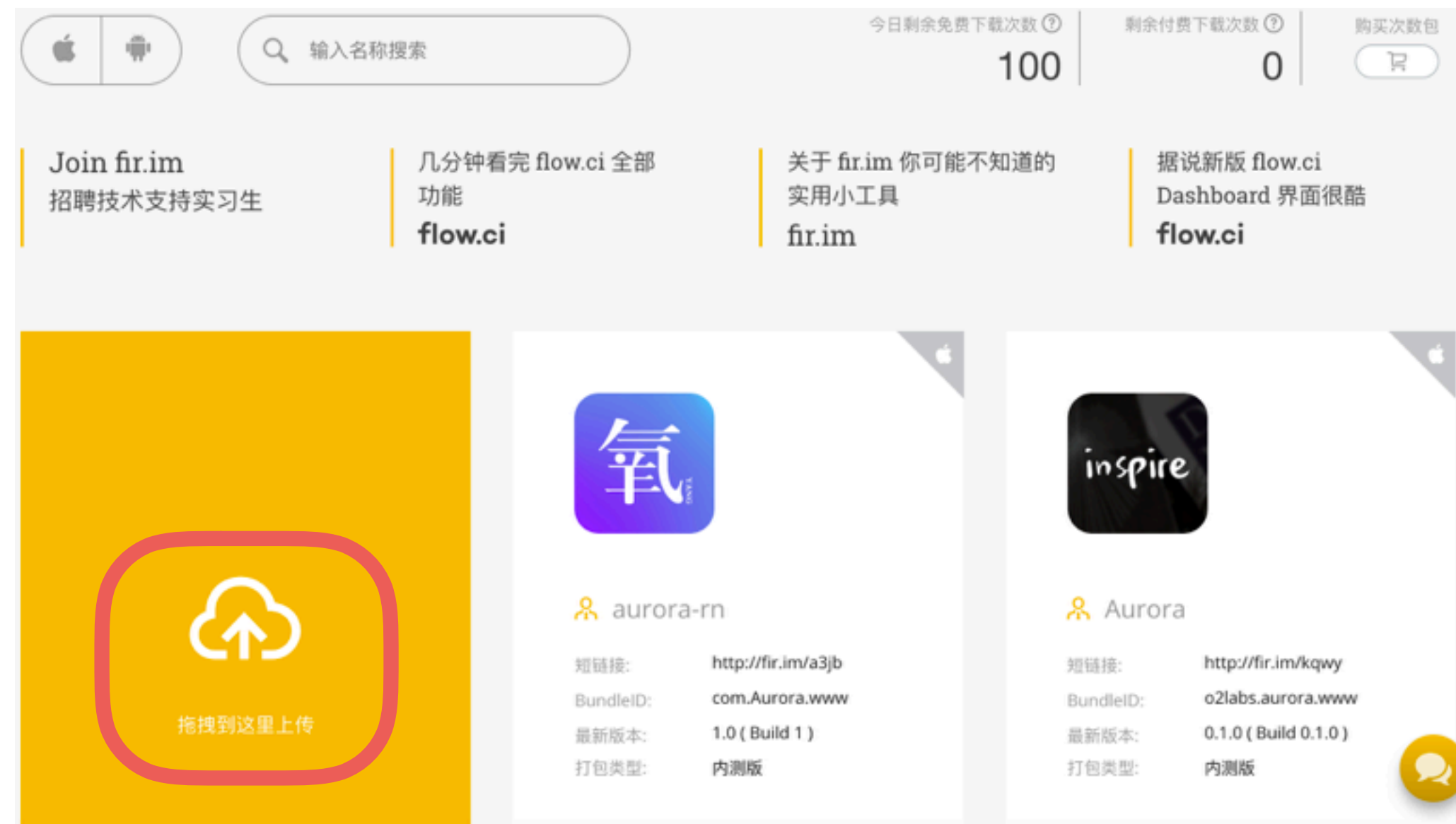
电子邮件	姓名 ^
j@gmail.com	feeling
57@qq.com	AV
@qq.com	ray

# 公开测试

02

- 第三方众测平台，如 Fir.im <https://fir.im>

需要上传 IPA 包



ipa? — iOS系统的软件包(iPhone application archive)

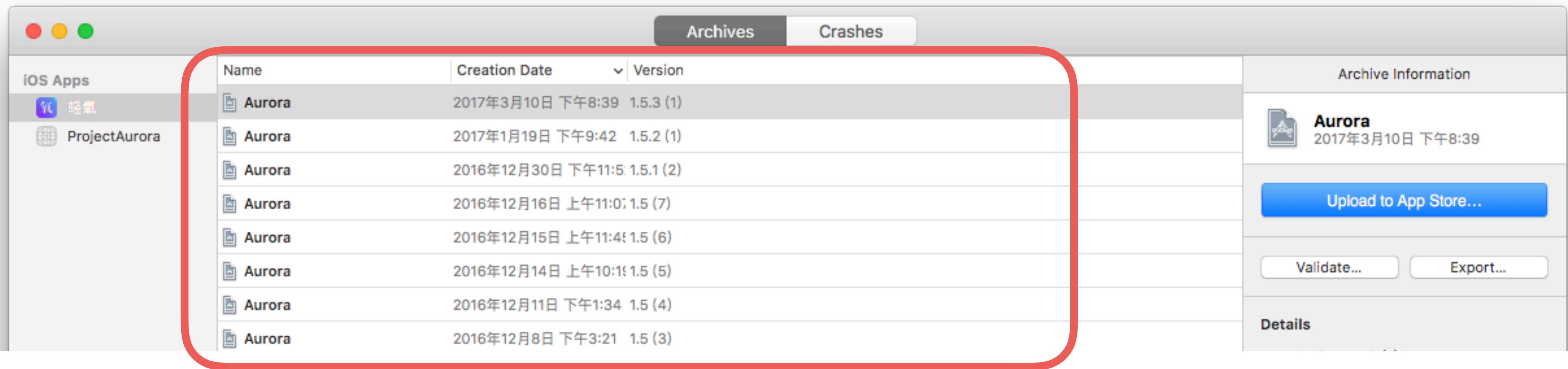
# 如何生成 IPA 包





# 如何生成 IPA 包

02



1. 选择指定 Archive, **右键进入 Finder**
2. 选择 .xcarchive 文件, **右键「显示包内容」**
3. Products/Applications/ **找到 .app**
4. **拖拽到 iTunes** 里生成
5. 进入目录 /Users/{user name}/Music/iTunes/iTunes Media/Mobile Applications **找到 .ipa**

- ▶ iOS 应用打包和发布
- ▶ Android 应用打包和发布

▸ Android 应用打包和发布

打包 ..... 发布

打包工具

真机测试

环境变量

发布渠道

打包密钥

打包命令

Gradle    安卓构建工具，类似于 NPM + Gulp

ADB    安卓调试命令行，用于真机上安装和调试应用

## Debug

- 包含程序调试信息
- 可执行断点等调试指令
- RN 会显示警告信息

## Release

- 不包含任何调试信息
- 压缩体积小、运行速度快
- RN 无法实时更新，需重新编译



发版本：

**Release**

Release 版本在真机上安装需要通过密钥进行签名

1. 生成签名密钥
2. 设置 gradle 变量
3. 添加签名到项目的 gradle 配置文件
4. 终端执行 gradle 打包命令
5. 在 android/app/build/outputs/apk 目录下找到 .apk 文件

详细步骤请参考：

<http://reactnative.cn/docs/0.42/signed-apk-android.html>

- ▶ 通过 USB 安装
  1. 连接电脑，设备开启 USB 调试模式
  2. 终端运行 `adb devices` 查看是否存在设备列表信息
  3. 终端运行 `react-native run-android` 直接安装在真机上
- ▶ 直接将 apk 发送到手机中安装

# 安卓发布渠道

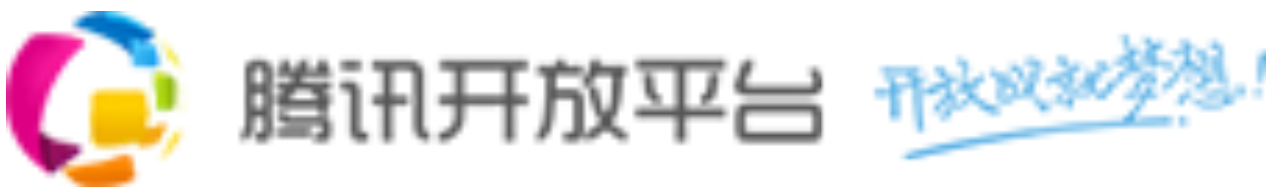
国外



国内



-----



-----



放心下，畅快玩

One more thing...



将 RN 代码转换成 Web 的技巧

# RN 编译成 Web 的方案

## A. `necolas/react-native-web`

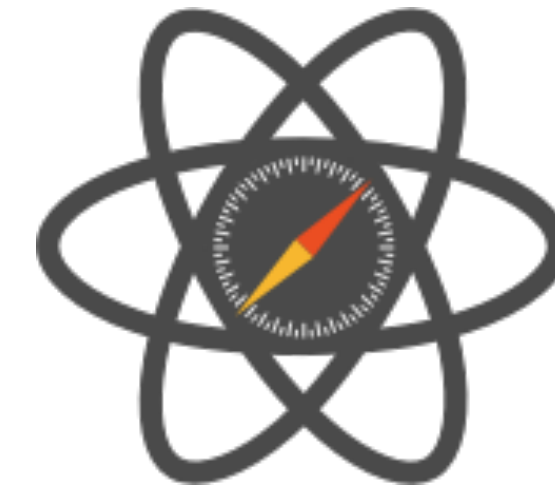
<https://github.com/necolas/react-native-web>

## B. `taobaofed/react-web`

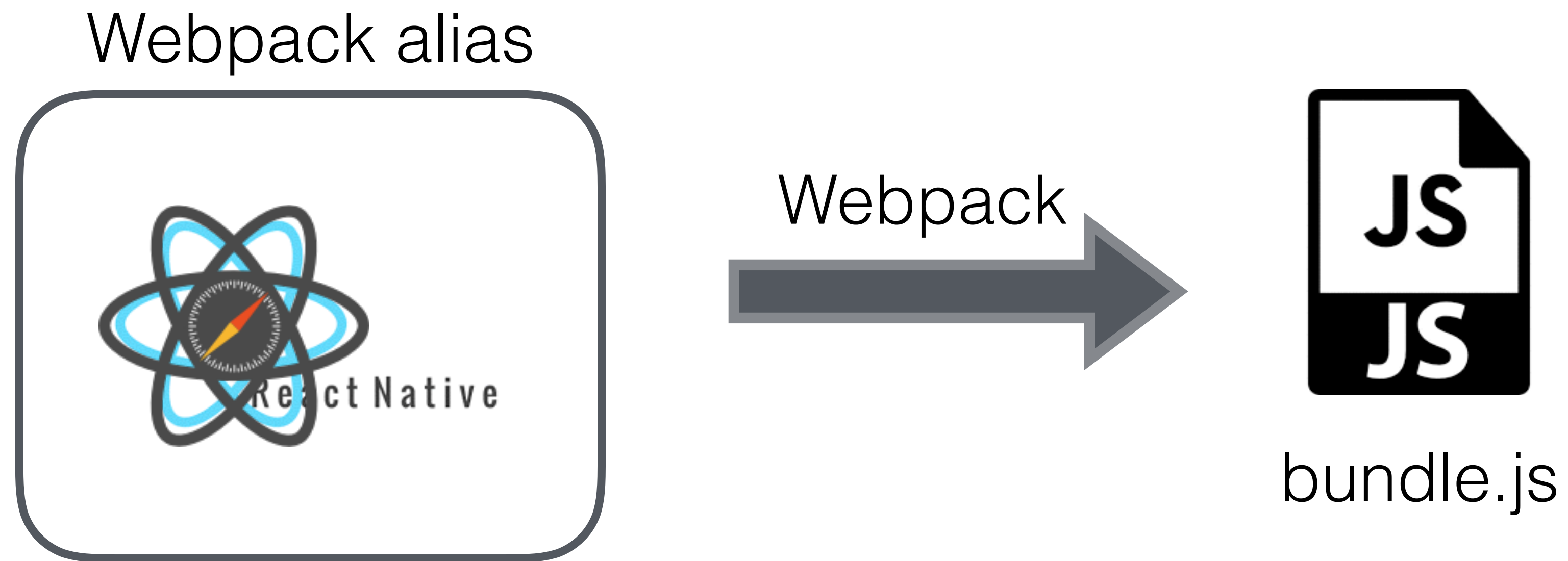
<https://github.com/taobaofed/react-web>



- View
- Text
- TextInput
- ListView
- ProgressBar
- Image
- ...



- Div
- Span
- Input
- Div
- Div
- Img
- ...



1. 安装必要的组件
2. 创建 index.web.js 的入口组件
3. 创建并配置 webpack.config.js 文件
4. 创建一个 HTML 模板
5. 命令行通过 webpack 进行打包

详细步骤请参考：

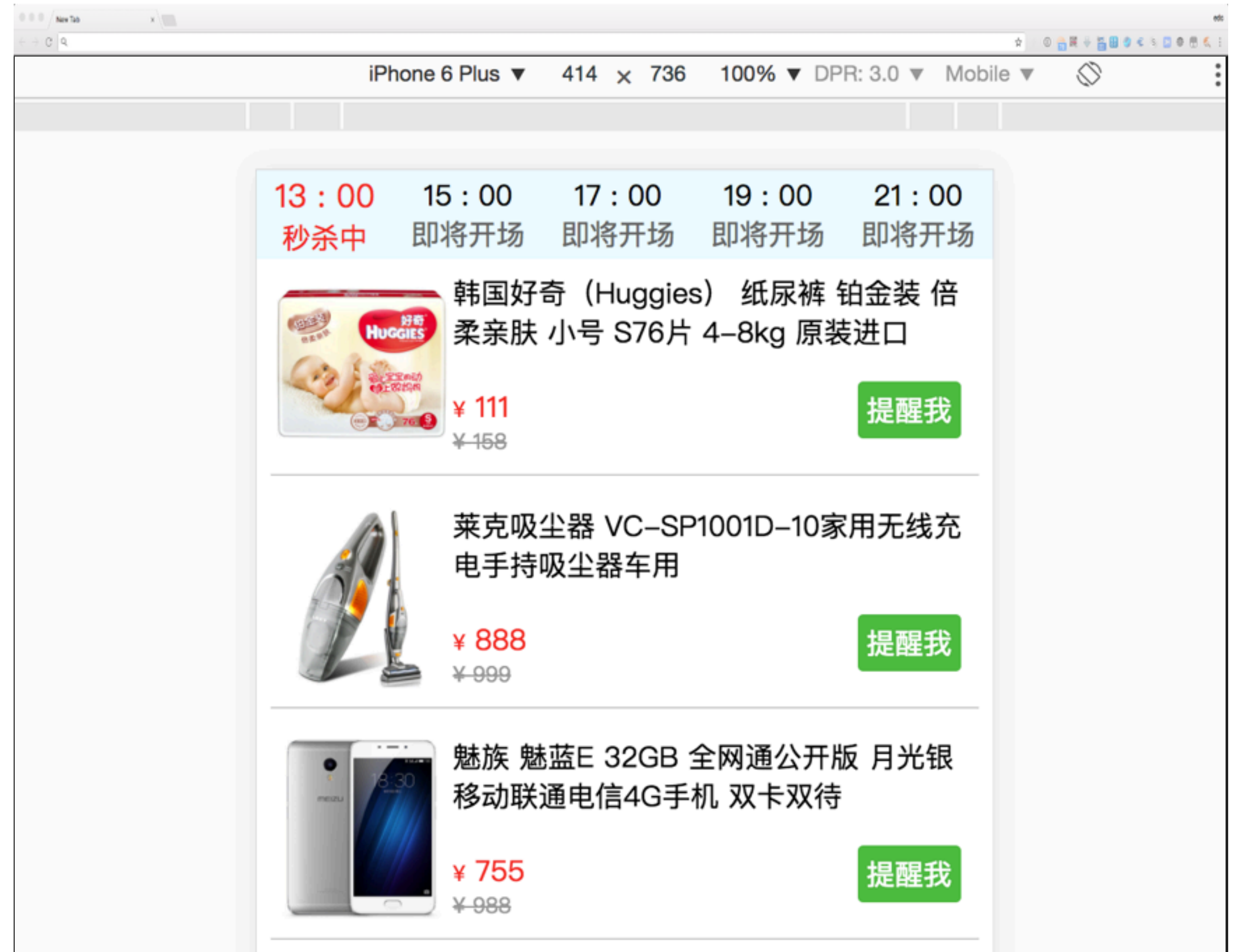
<https://github.com/o2team/react-native-web-demo>



## 秒杀 iOS 版



## 秒杀 Web 版



**T H A N K S**  
**FOR YOUR WATCHING**



OPEN ORIENTED

凹凸实验室