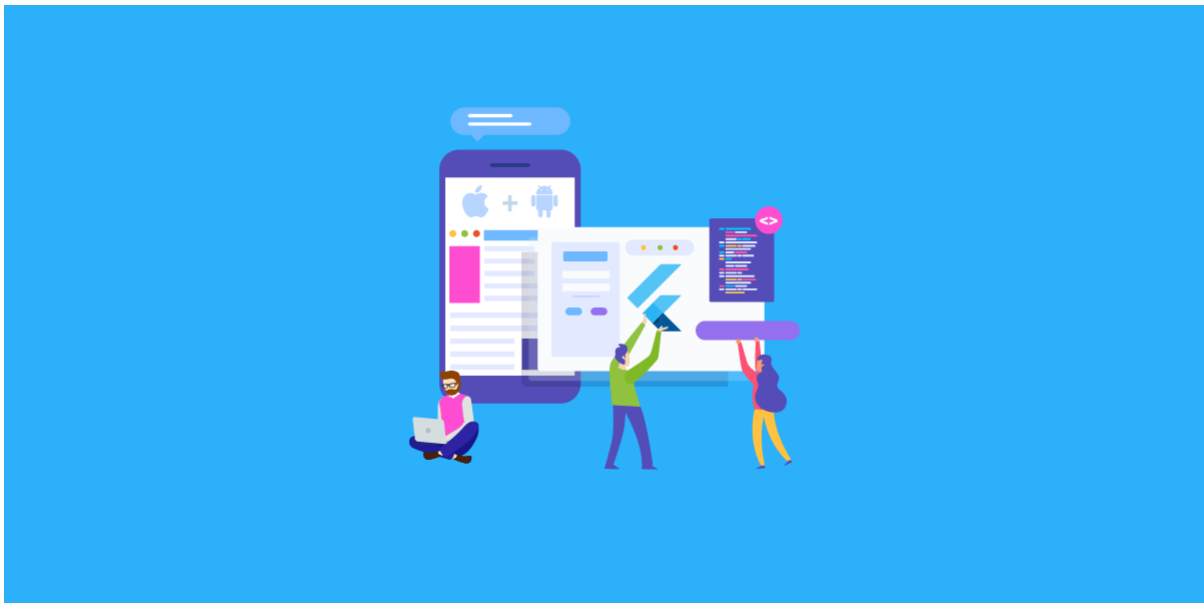


【赠送内容】编译和打出iOS应用



主要步骤和流程

- 申请iOS企业级或个人开发者账号
 - 通过链接<https://developer.apple.com/programs/>进行注册
- 在App Store Connect注册你的APP
 - 为你的APP注册Bundle ID
- 检查 Xcode 项目设置
- 更新应用的开发版本
- 添加APP启动图标
- 设置签名账号
- 创建构建存档
- 在 TestFlight 发布你的应用
- 在 App Store 发布你的应用

打包并发布 iOS 应用

预先准备

在开始发布你的 app 的进程之前，确保你已经看过了 Apple 的 [App Store 审核指南](#)。

想要发布你的 app 到 App Store，你需要注册 [Apple Developer Program](#)。你可以在苹果的 [选择会员资格\(开发者类型\)](#) 中查看到关于多种不同会员类型的选择。

在 App Store Connect 上注册你的 App

在 [App Store Connect](#) 管理你的应用生命周期（以前是 iTunes Connect）。你需要定义你的应用名称、描述以及截图，然后设置价格并管理发布到应用商店和 TestFlight。

[App Store Connect](#)（曾经的 iTunes Connect）是你将会管理应用生命周期的地方。你将定义应用的名称和描述以及截图，设置价格，并管理发布到 App Store 和 Testflight。

注册你的 app 需要两步：登记唯一的套装 ID（Bundle ID），并在你的 App Store Connect 中创建一个 app。

关于更多 App Store Connect 的细节，查看 [App Store Connect 指南](#)。

注册 Bundle ID

每一个 iOS 应用都与一个在 Apple 登记的唯一的套装 ID 关联。要为你的应用登记一个套装 ID，请参考下面的步骤：

1. 在你的开发者账号页面打开 [App IDs](#) 页面。
2. 点击 + 来创建一个新的套装 ID。
3. 输入一个 App 名称，选择 **Explicit App ID**，然后输入一个 ID。
4. 选择你的 App 将要使用的服务，然后点击 **继续**
5. 在下一页，确认细节并点击 **注册** 来注册你的 Bundle ID。

在 App Store Connect 创建一个应用记录

在 App Store Connect 中注册你的应用：

接下来，你需要在 App Store Connect 注册你的应用：

1. 在你的浏览器里打开 [App Store Connect](#)。
2. 在 App Store Connect 的落地页，点击 **My Apps**。
3. 在我的 app 页面的顶部左侧，点击 +，然后选择 **New App**。
4. 在出现的表单中填写你的 app 细节。在平台部分，确保 iOS 被选中。由于 Flutter 暂时不支持 tvOS，保持该选项为未选。点击 **Create**。
5. 跳转到你的应用详情，然后从侧边栏选择 **App Information**。
6. 在基础信息部分，选择你在前一步注册的套装 ID。
想要获取更多信息，可以看这个帮助页面 [添加 App 至您的帐户](#)。

检查 Xcode 项目设置

在这一步，你需要在 Xcode 工作空间检查绝大多数重要设置。关于更多的步骤和描述，查看 [为 App 分发做准备](#)。

在 Xcode 中跳转到你的目标设置：

1. 在 Xcode 中，打开你的 App 的 ios 目录中的 `Runner.xcworkspace`
2. 想要看你的 app 设置，在 Xcode 的项目导航栏中选择 **Runner**
3. 选择 **General** tab

接下来，你需要验证最重要的配置：

在 **Identity** 部分：

Display Name
应用的名字。

Bundle Identifier

在 App Store Connect 注册的 App ID。

在 **Signing & Capabilities** 部分：

Automatically manage signing

是否需要 Xcode 自动管理 app 签名和设置。这个默认被设置为 `true`，对于绝大多数 App 来说都是适用的。对于更复杂的场景，查看 [代码签名指南](#)。

Team

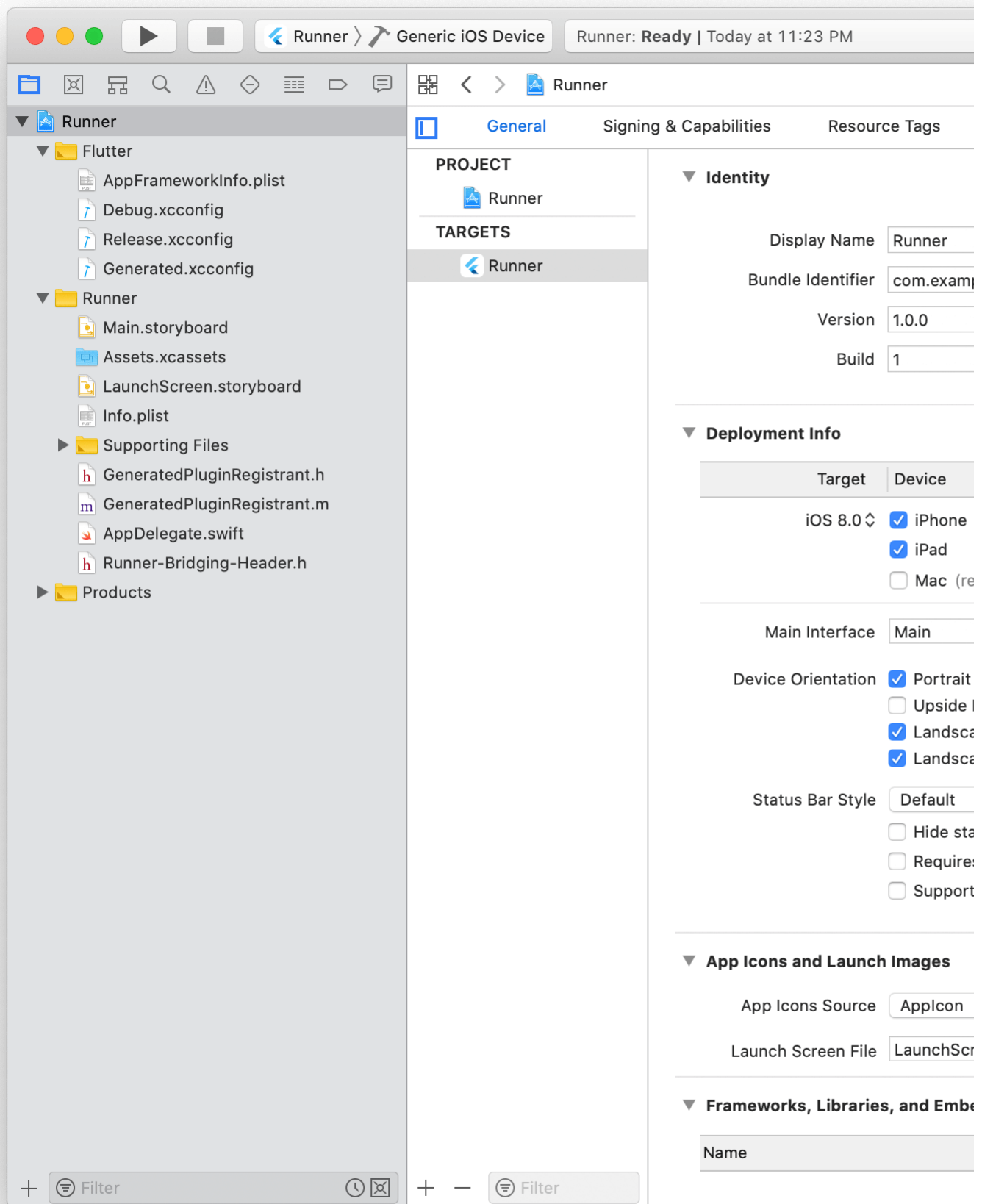
选择关联到你注册的 Apple 开发者账户的团队。如果需要，选择 **Add Account...**，然后更新选项。

在 **Build Settings** 部分：

iOS Deployment Target

你的 App 将会支持的最低版本的 iOS，Flutter 支持 iOS 8.0 及以后的版本。如果你的 App 包含使用了 iOS 8 不支持的 API 的 Objective-C 或 Swift 代码，请设置到合适的版本。

你项目的 **General** tab 应该看起来像是这样的：



更多关于 App 签名新的介绍，查看文档 [创建、导出、和删除签名证书](#)。

更新应用的开发版本

如果你在 Xcode 工程里更改了 Deployment Target，你需要打开 Flutter app 的 `ios/Flutter/AppframeworkInfo.plist` 文件并修改 `MinimumOSVersion` 值与之匹配。

更新应用版本号

每个应用默认的初始版本号是 1.0.0。若要更新它，请转到 `pubspec.yaml` 文件并更新以下内容：

```
version: 1.0.0+1
```

版本号由三个点分隔的数字组成，例如上面样例中的 1.0.0。然后是可选的构建号，例如上面样例中的 1，以 + 分隔。

版本号与构建号都可以在 Flutter 打包时分别使用 `--build-name` 和 `--build-number` 重新指定。

在 iOS 中，当 `build-number` 用作 `CFBundleVersion` 的时候，`build-name` 用作 `CFBundleShortVersionString`。阅读关于 iOS 版本控制的更多信息请参考 Apple 开发者网站提供的 [Core Foundation Keys](#)。

添加应用图标

当你创建一个新的 Flutter 应用时，则会创建一个默认的图标。在这一步，你将使用你自己的图标替换占位图标：

1. 回顾 [iOS 的 App Icon](#) 指南。
2. 在 Xcode 项目导航栏，选择 `Runner` 目录中的 `Assets.xcassets`，更新占位图标为你自己的 app 的图标。
3. 通过执行 `flutter run` 来验证你的图标是否已经被替换。

创建一个构建归档 (build archive)

在这一步，你将创建一个构建归档，并上传到 App Store Connect。

在开发过程中，你将使用 `debug` 模式来完成构建、调试并测试。当你准备好通过 App Store 或 TestFlight 交付你的 app 给用户时，你需要准备一个 **release** 构建。这时你也许想要 [混淆你的 Dart 代码](#) 以加大反编译难度。混淆你的代码需要在 build 的时候添加一些标志，并维护其他文件以消除反编译的堆栈跟踪。

在命令行中你的应用目录下执行如下步骤：

1. 执行 `flutter build ios` 来创建一个 release 构建 (`flutter build` 默认指向 `--release`)。
2. 为了确保 Xcode 刷新了发布模式的配置，关闭并重新打开你的 Xcode。对于 Xcode 8.3 及以后版本，这一步不是必须的。

在 Xcode 中，配置 app 的版本，并开始构建：

1. 在 Xcode 中，打开你应用 `ios` 目录中的 `Runner.xcworkspace`
2. 选择 **Product > Scheme > Runner**
3. 选择 **Product > Destination > Any iOS Device**
4. 在 Xcode 项目导航栏中选择 **Runner**，然后在设置界面侧边栏选择 **Runner** 目标。
5. 在 Identity 部分，更新 **Version** 为你想要发布的用户可见的版本号。
6. 在 Identity 部分，更新 **Build** 标示为一个唯一的 Build 数字，用来在 App Store Connect 上追踪。每一个上传都需要一个独立的 Build 数字。

最后，创建一个构建归档并将其上传到 App Store Connect：

1. 选择 **Product > Archive** 来生成一个构建归档

提示

在 Flutter 1.24.0-6.0 以及之后的版本你可以通过运行 ``flutter build ipa`` 创建一个归档 (archive)。然后打开 Xcode 中的 ``build/ios/archive/MyApp.xcarchive`` 目录以验证并发布你的应用。运行 ``flutter bui`

2. 在 Xcode 组织界面的侧边栏中，选择你的 iOS app，然后选择你刚刚生成的构建归档。
3. 点击 **Validate...** 按钮。如果报告了任何问题，记录下他们并重新开始一个新的构建。在你上传一个归档前，可以一直使用同一个 Build ID。
4. 当这个归档校验成功以后，点击 **Upload to App Store...**。你可以在 [App Store Connect](#) 中应用详情页面的 Activities 标签页查看你的构建状态。

提示

当你最后在 `**Distribute App**` 导出应用时，Xcode 将会创建一个含有你的应用 IPA 以及 ``ExportOptions.plist`` 文件的文件夹。在 Flutter 1.24.0-6.0 版本后，你可以无需启动 Xcode，通过运行 ``flutter build ipa`

当你的构建已经通过了校验，可以将你的构建通过 TestFlight 发布给你的测试人员或将其发布到 App Store 的时候，你会在 30 分钟内收到一封来信提醒你。

更多信息可以查看 [上传一个 App 到 App Store Connect \(Upload an app to App Store Connect\)](#)

在 TestFlight 发布你的应用

[TestFlight](#) 允许开发者发布他们的 app 给内部或外部的测试人员。这个步骤为可选，接下来的内容将引导你将发布你的构建到 TestFlight。

1. 在 [App Store Connect](#) 中，你的应用的详情页面跳转到 TestFlight Tab。
2. 在侧边栏选择 **Internal Testing**。
3. 选择要发布给测试人员的构建，然后点击 **保存**。
4. 为每一个内部测试人员添加邮件。你可以在 App Store Connect 的 **用户与角色** 页面添加额外的内部用户，他们将会出现在页面顶部的下拉菜单中。

关于更多信息，请查看 [使用 TestFlight 分发应用 \(Distribute an app using TestFlight \(iOS, tvOS, watchOS\)\)](#)。

在 App Store 发布你的应用

当你准备发布你的 app 到这个世界时，跟随下面的步骤，来提交你的 App 去审核，并将其发布到 App Store。

1. 从你的 app 在 [App Store Connect](#) 的页面中的侧边栏中选择 **Pricing and Availability**，然后完善所有的必填信息。
2. 从侧边栏选择状态。如果这是第一次发布这个 App，这个状态将会是 **1.0 Prepare for Submission**，填写所有需要填写的区域。
3. 点击 **提交审核**

Apple 将会在他们的审核过程结束后提醒你。你的 app 将会根据 **Version Release** 部分的介绍进行发布。

关于更多细节，查看 [通过 App Store 分发一个 App](#)。

当你创建了应用之后，你可以通过 Google Ads 吸引更多用户，Google Ads 平台可以通过机器学习帮助你以非常高的性价比吸引到更多用户。

1. 创建广告—我们会根据您的应用信息帮您制作广告。另外，您还可以添加图片和视频。
2. 决定推广预算—对于以提高应用安装量为主要目标的广告系列，您需要为其设置应用安装出价，也就是“目标每次安装费用”，同时设置每日推广支出预算。
3. 选择目标地区—让我们知道你希望触达哪些区域的用户。
4. 设定用户行动—决定你希望用户要做什么，比如安装，应用内操作或者目标广告支出回报率 (ROAS)。

故障排除

[分发你的 app](#) 指南提供了一个详细的关于发布一个 App 到 App Store 过程的内容。

更多参考资料

- <https://docs.flutter.dev/deployment/ios>