

# ios 调试与发布

## 概念介绍

### 1、开发者账号类型

iOS



#### iOS Developer Program Individual

\$99 / Year

For an individual developer who will be creating iOS apps for distribution on the App Store.

#### iOS Developer Program Company

\$99 / Year

For a company with a development team who will be creating iOS apps for distribution on the App Store.

**Note:** A D-U-N-S Number is required.

#### iOS Developer Enterprise Program

\$299 / Year

For a company who will be creating proprietary, in-house iOS apps.

**Note:** A D-U-N-S Number is required.

#### iOS Developer University Program

Free

For higher education institutions looking to introduce iOS development into their curriculum.

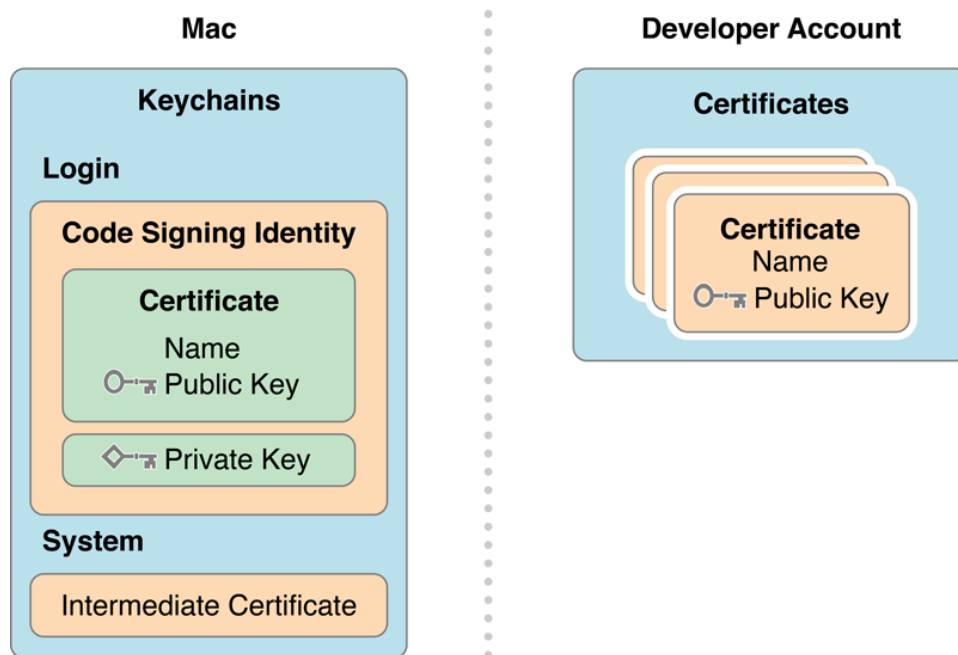
iso 开发者账号分为 个人(individual), 公司(company), 企业(enterprise)三种类型, 还有一种比较特殊的是教育机构开发者账号, 免费的。个人和公司都称为标准账号, 99 美金一年, 企业账号是 299 美金。个人账号只能有一个开发者, 公司账号可以允许邀请多个开发者协作开发(比如可以共享开发平台等, 这个对于需要多人协作开发的好处很多)。企业账号, 可以发布 in-house 内部 APP, 其 app 只能用于内部员工使用, 是无法对外公开的。教育机构开发者账号是面对学校这些教育机构的, 所以, 普通情况下大家都是选择个人或者公司账号。

### 2、代码签名(code signing)

代码签名的重要性: 给应用程序签名是为了让用户相信这个 APP 是已经通过苹果验证的开发者发布的, 从而能够识别这个 APP 是否是伪造的或者给别人修改过的。这样能够保证只有经过签名的应用程序才能保证他的来源是可信任的, 并且代码是完整的, 未经修改的。

我们申请一个 Certificate 之前, 需要先在本地申请一个 Certificate Signing Request (CSR) 文件, 而这个过程实际上是生成了一对公钥和私钥, 保存在你 Mac 的 Keychain 中。代码签名正是使用这种基于非对称密钥的加密方式, 用私钥进行签名, 用公钥进行验证。如下图所示, 在你 Mac 的 keychain 的 login 中存储着相关的公钥和私钥, 而证书中包含了公钥。你

只能用私钥来进行签名，所以如果没有了私钥，就意味着你不能进行签名了，所以就无法使用这个证书了，此时你只能 `revoke` 之前的证书再申请一个。因此在申请完证书时，最好导出并保存好你的私钥。当你想与其他人或设备共享证书时，把私钥传给它就可以了。私钥保存在你的 Mac 中，而苹果生成的 Certificate 中包含了公钥。当你用自己的私钥对代码签名后，苹果就可以用证书中的公钥来进行验证，确保是你对代码进行了签名，而不是别人冒充你，同时也确保代码的完整性等。



### 3、证书(Certificates)

在开发 iOS 应用的时候，我们需要签名证书（开发证书）来验证，并允许我们在真机上对 App 进行测试。另外，在发布 App 到 App store 的时候，我们也需要证书(发布证书)来做验证。所以在给你的 app 签名之前，你应该准备好已经创建了调试证书或者发布证书。

证书主要分为两类：Development 和 Production，Development 证书用来开发和调试应用程序，Production 主要用来发布应用程序（根据证书种类有不同作用），下面是证书的分类信息：

## Development

- iOS App Development**  
Sign development versions of your iOS app.
  - Apple Push Notification service SSL (Sandbox)**  
Establish connectivity between your notification server and the Apple Push Notification service sandbox environment. A separate certificate is required for each app you develop.
- 

## Production

- App Store and Ad Hoc**  
Sign your iOS app for submission to the App Store or for Ad Hoc distribution.
- Apple Push Notification service SSL (Production)**  
Establish connectivity between your notification server and the Apple Push Notification service production environment. A separate certificate is required for each app you distribute.
- Pass Type ID Certificate**  
Sign and send updates to passes in Passbook.
- Website Push ID Certificate**  
Sign and send updates for Websites.

- **Development**

App Development: 用来开发和真机调试应用程序。

Push Development: 用来调试 Apple Push Notification

- **Production**

In-House : 用来发布 In-House 的应用程序。(企业账号才能发布 In-House)

App Store and Ad Hoc: 用来发布提交 App Store 的应用程序和发布测试应用程序。

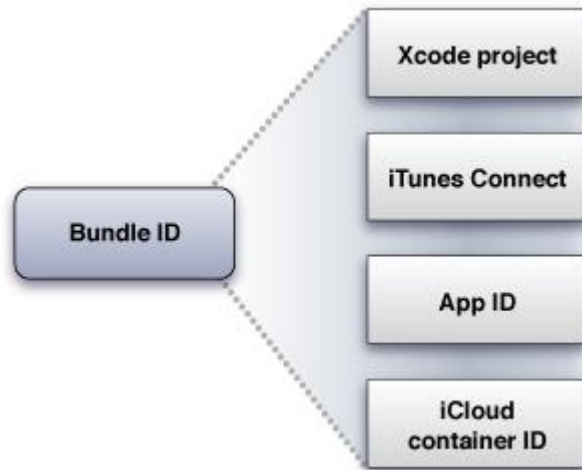
Push Production: 用来在发布版本中使用 Apple Push Notification。

Pass Type ID Certificate: 用来在发布版本中使用 Passbook

Website Push ID Certificate: 用来在发布版本中使用 safari 的 push

## 4、Bundle ID 与 App ID

Bundle ID 能够精确的识别一个 APP，这个和 Android 的包名非常类似，相当于一个 APP 的唯一标识。 Bundle ID 的使用涵盖从一个 APP 的整个开发过程到可以安装到设备以及当 APP 要发布面向消费者的时候提供给操作系统来使用。例如，当使用 Game Center and In-App Purchase 这些服务必须使用 bundle ID 来唯一标识你的 APP。系统使用此字符串来确定哪些偏好设置来适合这个 app，同样地，启动服务也用 Bundle ID 来定位一个 APP 可以打开一个特定的文件。Bundle ID 也可以用来验证 APP 的签名。下图列出 Bundle ID 被使用到的地方。



App ID 用于标识一个或者一组 App，App ID 应该是和 Xcode 中的 Bundle ID 是一致的或者匹配的。App ID 主要有以下两种：

- **Explicit App ID:** 唯一的 App ID，这种 App ID 用于唯一标识一个应用程序，例如 com.ABC.demo1，标识 Bundle ID 为 com.ABC.demo1 的程序。
- **Wildcard App ID:** 通配符 App ID，用于标识一应用程序。例如\*可以表示所有应用程序，而 com.ABC.\*可以表示以 com.ABC 开头的所有应用程序。

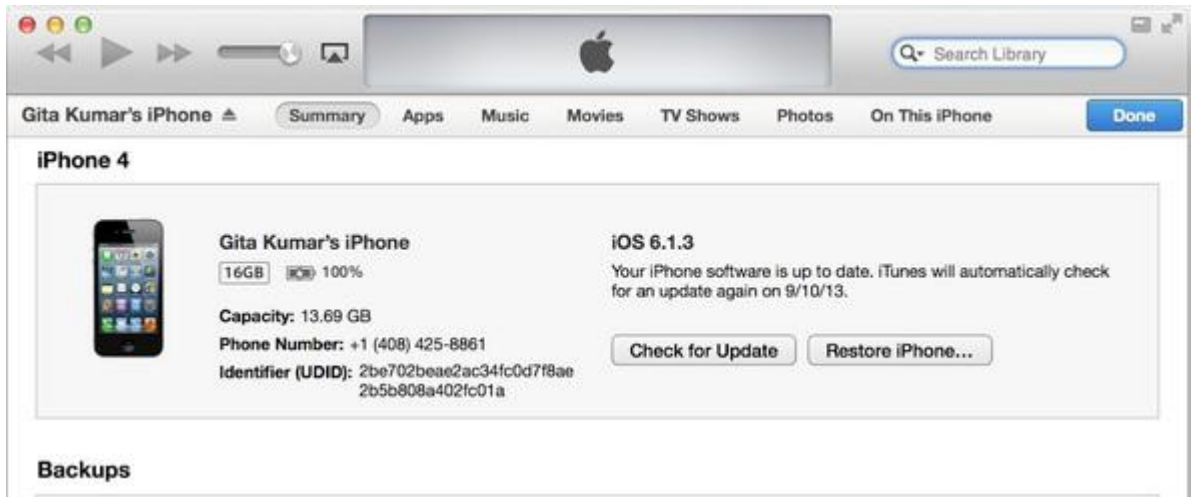
每创建一个 App ID，我们都可以设置该 App ID 所使用的 APP Services，也就是其所使用的额外服务。每种额外服务都有着不同的要求，例如，如果要使用 Apple Push Notification Services，则必须是一个 explicit App ID，以便能唯一标识一个应用程序。下面是目前所有可选的服务和相应的配置要求。

	Create Explicit App ID	Enable App ID	Set Entitlements	Edit Info.plist	Configure iTunes Connect
APNs	✓	✓			
Game Center	✓	✓			✓
iCloud		✓	✓		
In-App Purchase	✓	✓			✓
Data protection		✓			
Newsstand				✓	✓
Passbook		✓			
Routing apps				✓	✓

## 5、添加设备（Devices）

当需要将 APP 在真实 IOS 设备上运行，如果你的设备没有越狱的话，你需要将设备通过 xcode 或者开发者管理中心来注册你的设备。Devices 中包含了该账户中所有可用于开发和测试的设备。每个账户中的设备数量限制是 100 个。Disable 一台设备也不会增加名额。每台设备使用 UDID 来唯一标识。UDID（Unique Device Identifier）是区分物理设备的唯一标识。你的 iPhone 肯定有一个与众不同的 UDID，通常 UDID 会是一个 40 位十六进制字符串。你可以通过 xcode 或者 itunes 来查看设备的 UDID，如图：

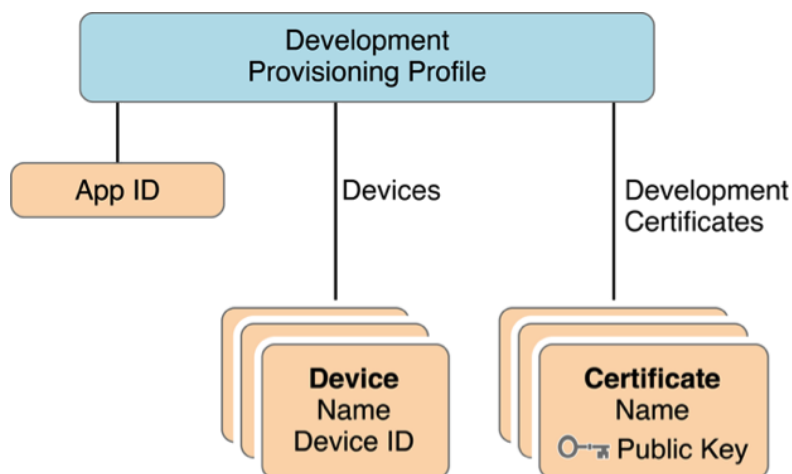




## 6、配置文件（provisioning profiles）

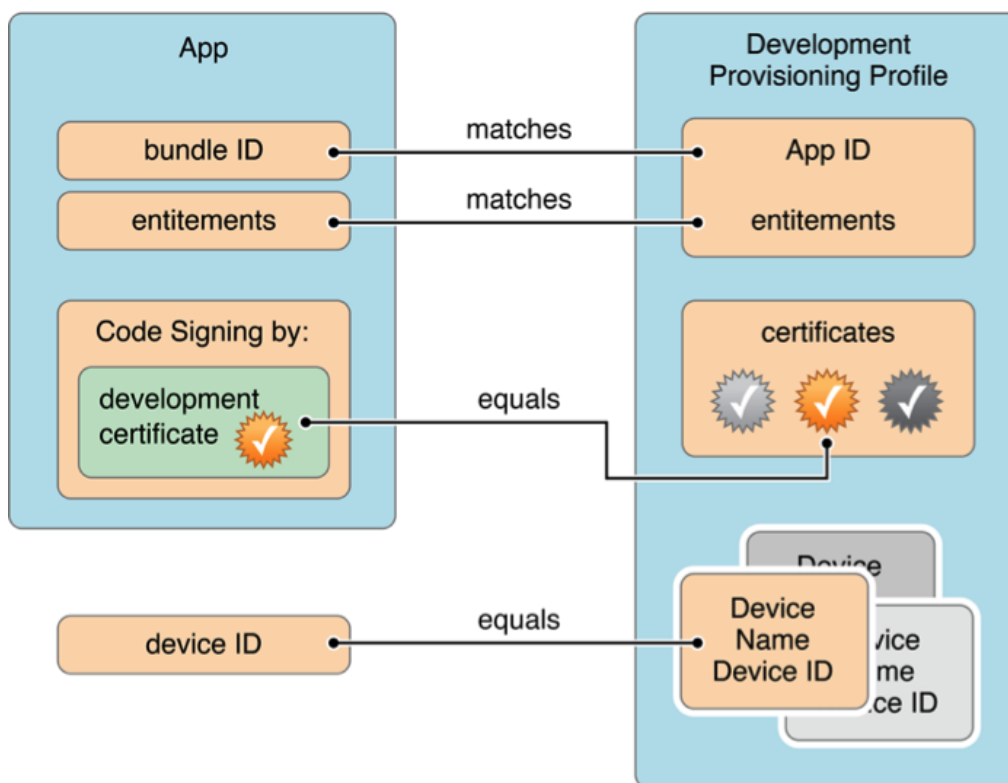
如果我们要打包或者在真机上运行一个应用程序，我们首先需要证书来进行签名，用来标识这个应用程序是合法的、安全的、完整的等等；然后需要指明它的 App ID，并且验证 Bundle ID 是否与其一致；再次，如果是真机调试，需要确认这台设备能否用来运行程序。而 Provisioning Profile 就把这些信息全部打包在一起，方便我们在调试和发布程序打包时使用，这样我们只要在不同的情况下选择不同的 profile 文件就可以了。而且这个 Provisioning Profile 文件会在打包时嵌入.ipa 的包里。

例如，如下图所示，一个用于 Development 的 Provisioning Profile 中包含了该 Provisioning Profile 对应的 App ID，可使用的证书和设备。这意味着使用这个 Provisioning Profile 打包程序必须拥有相应的证书，并且是将 App ID 对应的程序运行到 Devices 中包含的设备上去。



如上所述，在一台设备上运行应用程序的过程如下：

Your app launches on a device if:



与证书一样，Provisioning Profile 也分为 Development 和 Distribution 两种，对应上面所提到的调试证书和发布证书：

（注：前面提到不同账户类型所能创建的证书种类不同，显然 Profile 文件的种类是和你所能创建的证书种类相关的）

- Development
- Distribution

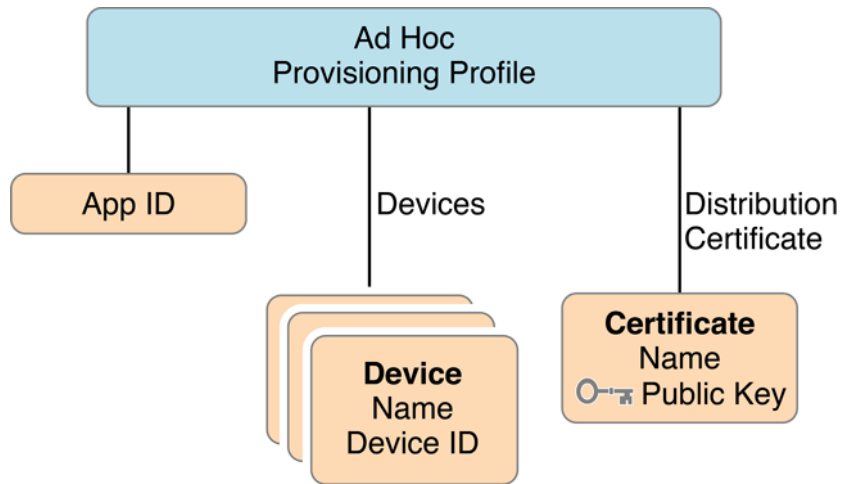
In House

Ad Hoc

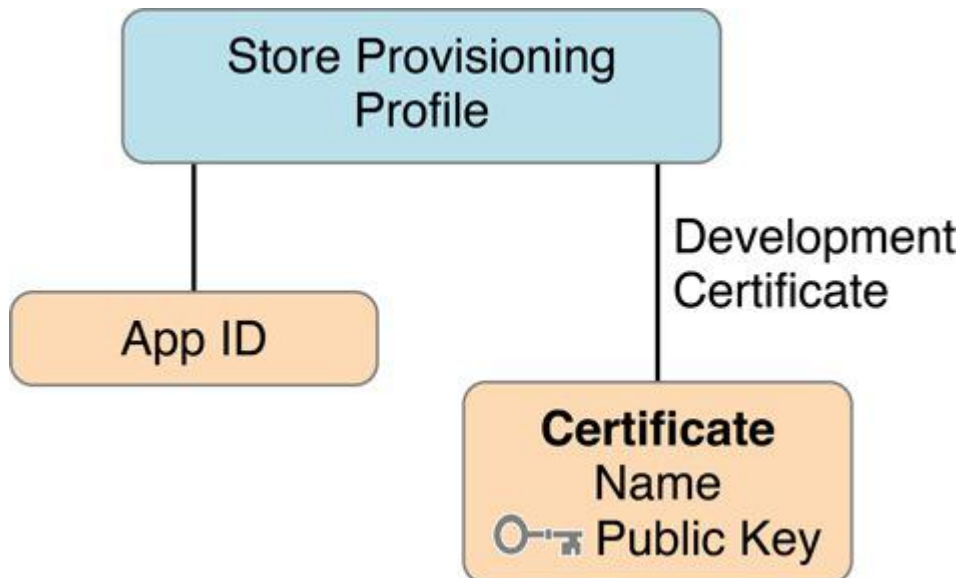
App Store

**In House:** 是企业账号发布的内部 APP，使用 In House 打包的 APP 是无法发布到 App Store 上对外销售的，也不需要经过 Apple 的评审。你可以把 “In House” 应用通过任何方式发布给你的企业员工、用户及其他你认可的任何人，没有注册设备的数量限制，尤其适合于企业应用的开发。

**Ad Hoc:** 是用来发布测试用的或者是越狱版本，Ad Hoc 的包只能运行在该账户内已经注册的可用设备或者已经越狱的设备，最多 100 个注册设备的数量限制。



App Store: 是用来提交到 App Store 上的, 发布状态的 Provisioning Profile 已经以签名的方式和 App 进行了绑定, 有一点不同的是, 发布状态的 Provisioning Profile 不需要指定 Device, 因为它不知道将被哪些设备使用。



## 开发/调试流程

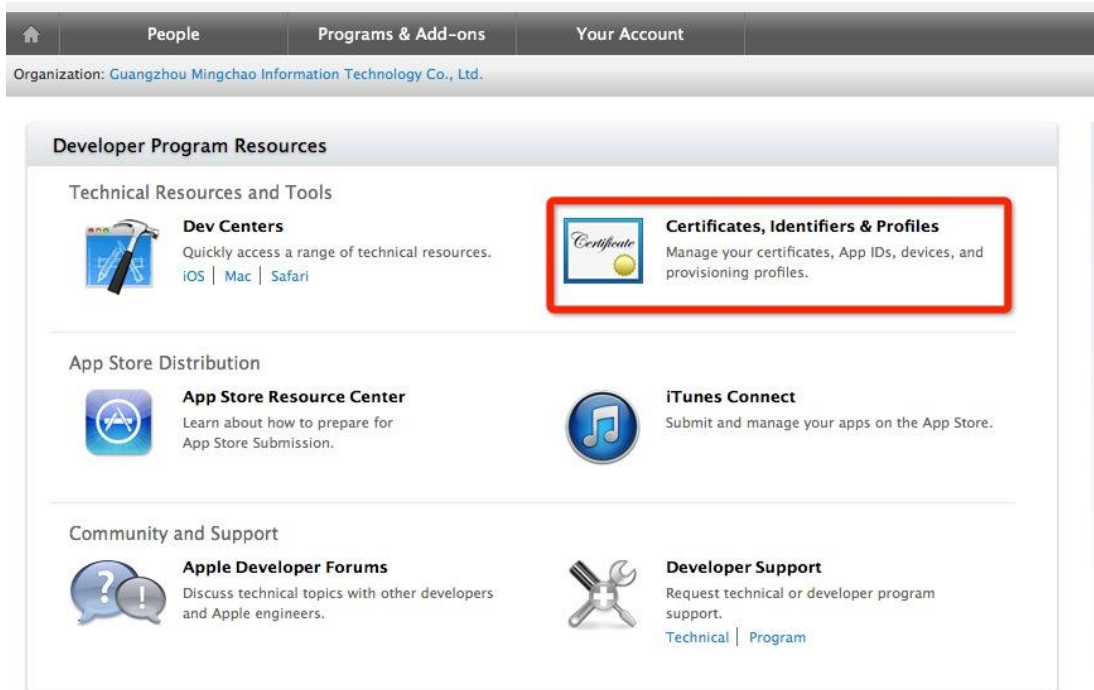
根据上面的介绍, 可以知道进行 Development 主要有以下几个步骤:

- 添加 APP ID
- 申请调试证书
- 加入设备
- 生成调试 Provisioning Profile
- 设置 Xcode Code Sign Identifier, 设置 debug 选项为调试证书



## 一、添加 APP ID

1、打开 [iOS Dev Center](#)，选择 Sign in，登陆（至少 99 美元账号），登陆之后在网页右边找到 iOS Developer Program,选择 Certificates, Identifiers & Profiles。



2、点击 App IDs, 点击右上角的“+”



入以下界面

进



## Registering an App ID

The App ID string contains two parts separated by a period (.)—an App ID Prefix that is defined as your Team ID by default and an App ID Suffix that is defined as a Bundle ID search string. Each part of an App ID has different and important uses for your app. [Learn More](#)

### App ID Description

添一个自己能分辨的名称即可

Name:

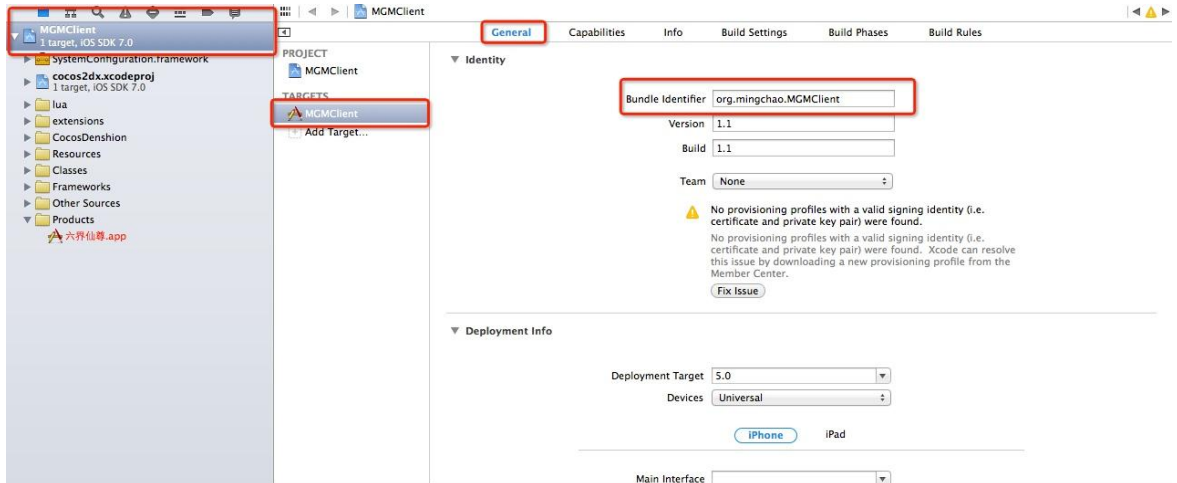
You cannot use special characters such as @, &, %, ', ", "

### App ID Prefix

自动生成

Value: 7T9FVP7WY3 (Team ID)

在 xcode 的左边项目导航区点击 MGMClient.xcodeproj 查看项目的 Bundle ID。



填写我们项目的 Bundle Identifier

### App ID Suffix

#### Explicit App ID

If you plan to incorporate app services such as Game Center, In-App Purchase, Data Protection, and iCloud, or want a provisioning profile unique to a single app, you must register an explicit App ID for your app.

To create an explicit App ID, enter a unique string in the Bundle ID field. This string should match the Bundle ID of your app.

Bundle ID:

We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (\*).

#### Wildcard App ID

This allows you to use a single App ID to match multiple apps. To create a wildcard App ID, enter an asterisk (\*) as the last digit in the Bundle ID field.

Bundle ID:

Example: com.domainname.\*

3、App Services: 应用程序提供的服务 (App Services: 选择你应用中将会使用的服务, 在 App ID 注册成功之后也可再次编辑你的选择)

在自己应用中所使用到的选项上打√, 因为我选择的是精确型 (Explicit), 所以 Game Center, In-App Purchase, Push Notifications 都是可选的, 勾选我们需要的 APP Services, 点击 Continue。

**App Services**  
Select the services you would like to enable in your app. You can edit your choices after this App ID has been registered.

Enable Services:

- Data Protection
  - Complete Protection
  - Protected Unless Open
  - Protected Until First User Authentication
- Game Center
- iCloud
- In-App Purchase
- Inter-App Audio
- Passbook
- Push Notifications

Cancel Continue

4、确认信息, 点击 Submit 即可

**ID Confirm your App ID.**

To complete the registration of this App ID, make sure your App ID information is correct, and click the submit button.

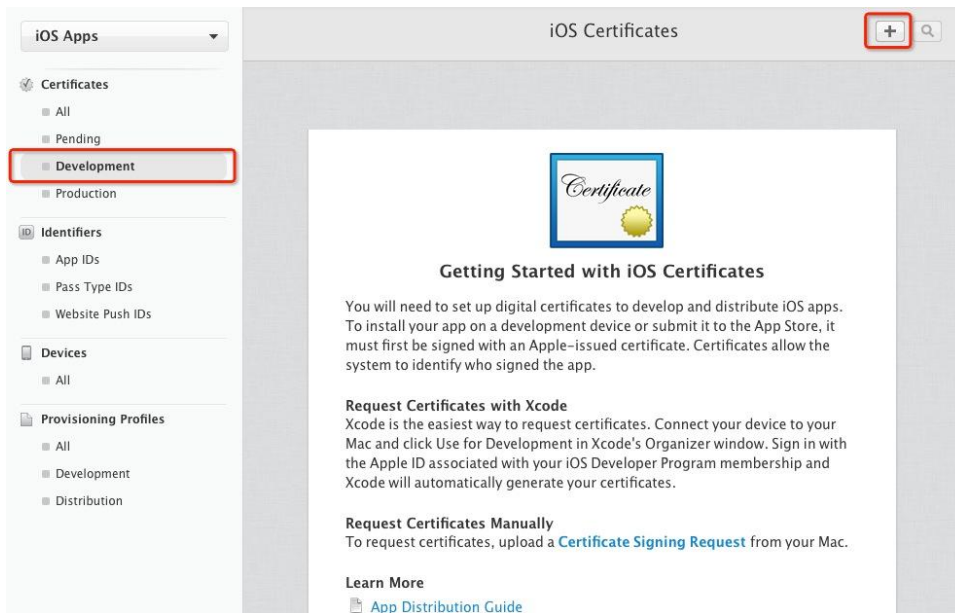
App ID Description: **MGMCClient**  
Identifier: **7T9FVP7WY3.org.mingchao.MGMCClient1**

Data Protection:  Disabled  
Game Center:  **Enabled**  
iCloud:  Disabled  
In-App Purchase:  **Enabled**  
Inter-App Audio:  Disabled  
Passbook:  Disabled  
Push Notifications:  **Enabled**

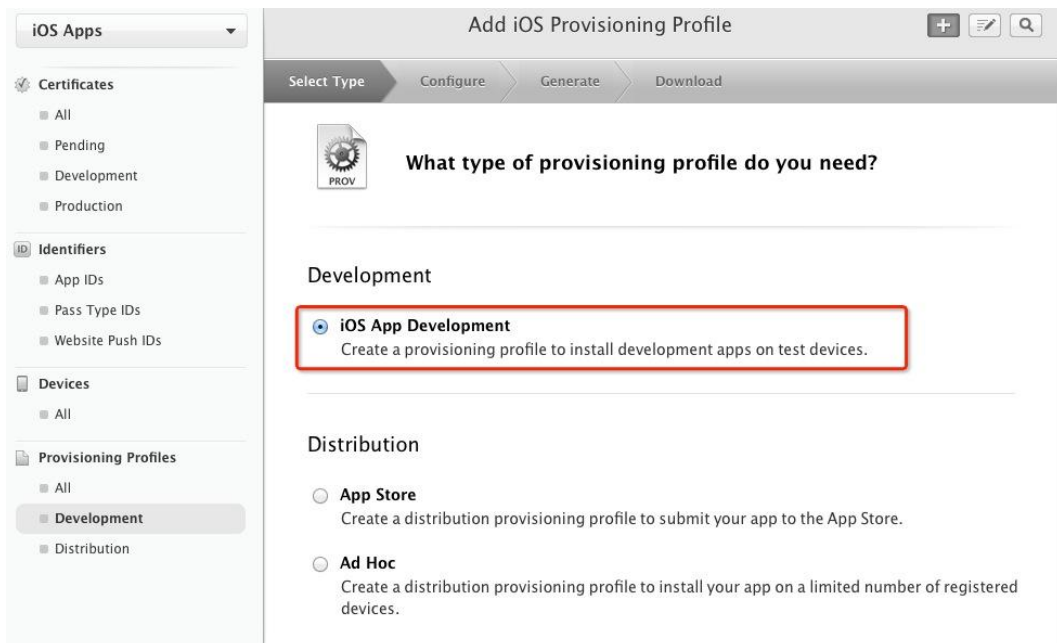
Cancel Back Submit

## 二、申请调试证书

1、点击 Development，点击右上角的“+”号



选择 iOS App Development



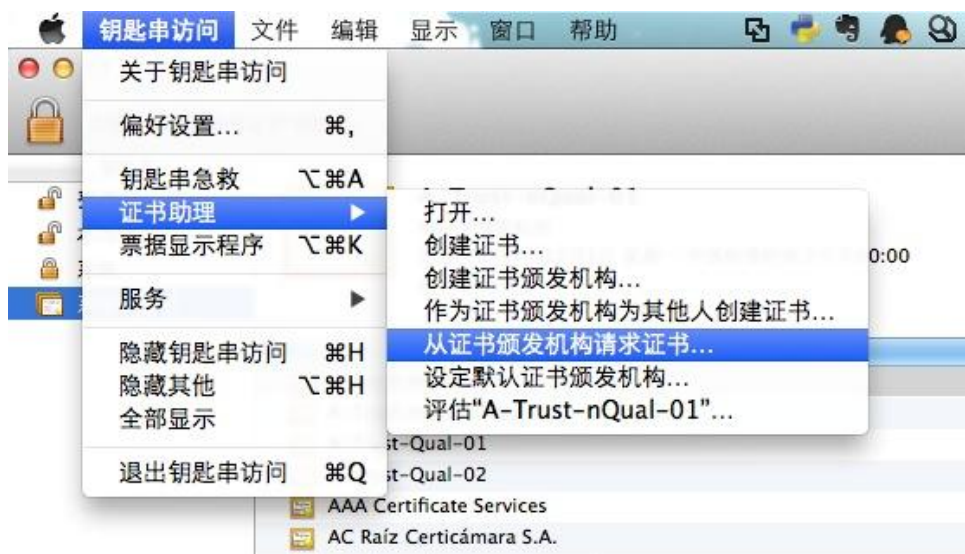
提示需要手动创建 CSR 文件



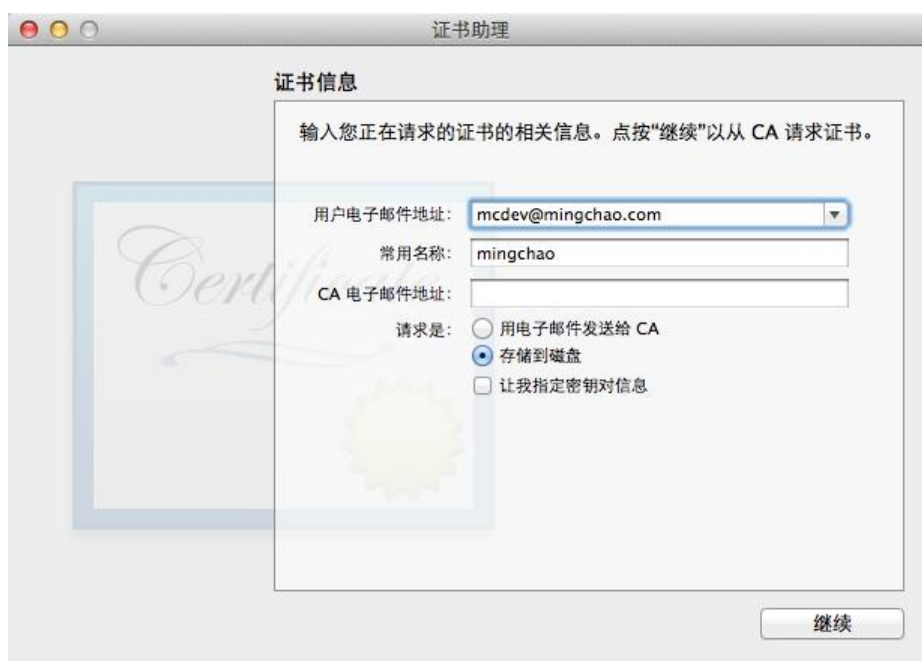
2、打开 LaunchPad,在“其他”文件夹可以看到钥匙串访问， 点击打开



3、选择钥匙串访问->证书助理->从证书颁发机构请求证书

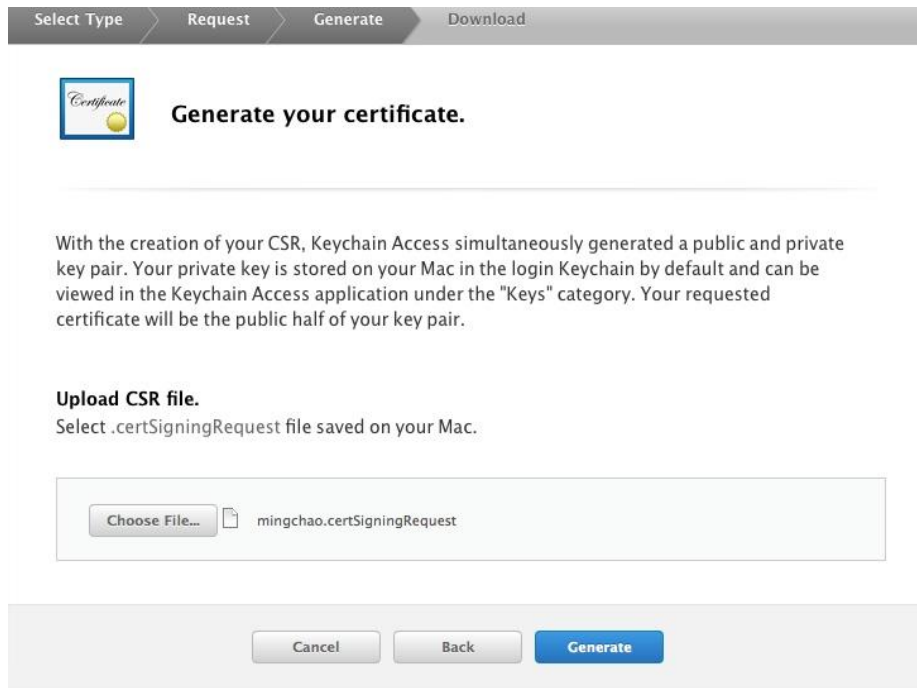


4、输入开发者账号邮箱和随便写一个常用名称，勾选存储到本地，点击继续，最后选择存储位置(一般存储在桌面)，CSR 文件完成。

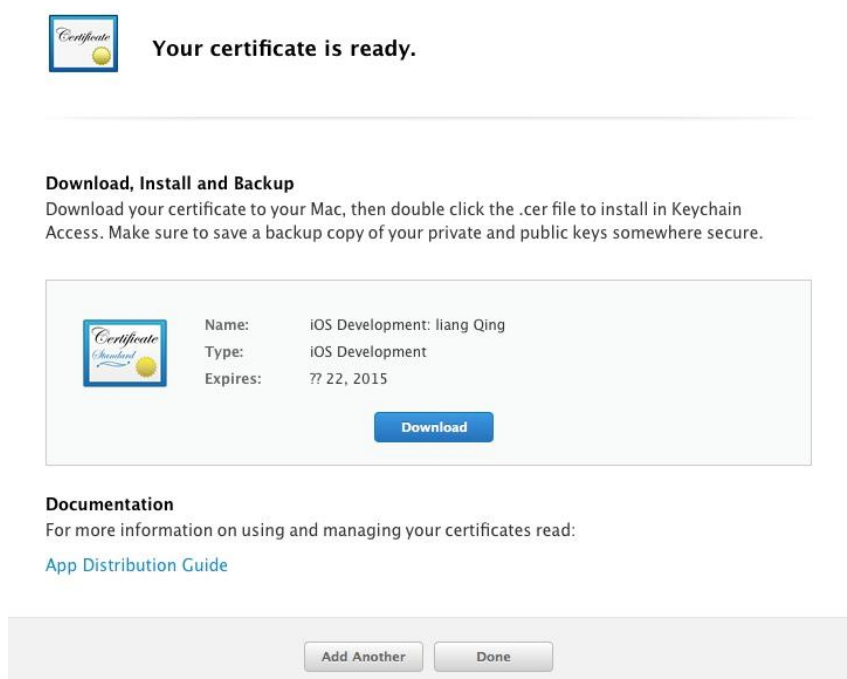


5、回到 developer 的界面，选择 Continue，出现上传证书界面，选择 Choose File 上传刚才存储的证书，然后选择 Generate.





6、最后出现以下界面，提示证书生成成功，然后单击 **Download** 下载到本地，双击安装即可



### 三、添加设备

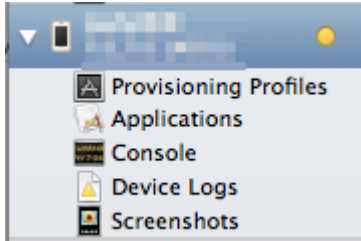
1、将设备连上电脑，打开在 **XCode->Window->Organizer->Devices** 中(快捷键 shift+command+2)

设备连接上 XCode 之后 Devices 中的信号灯有灰色，黄色，绿色三种情况

### 1.1、灰色灯

如果设备是之前未作为开发机，在 **Devices** 中设备上的灯应该为灰色，在右侧有选项将设备作为开发机，点击即可，由于没有新机，无法截图演示

### 1.2、黄灯，有两种情况



设备未连接好，请将连接线重新插一次，并检查接口是否插紧，如果还是黄灯，则未下一种情况

设备版本高于 XCode 所能支持的版本，如 XCode4.6.3，能支持的最高版本未 iOS6.1，如果你的设备未 iOS7 及以上，此时需要更新至 XCode5

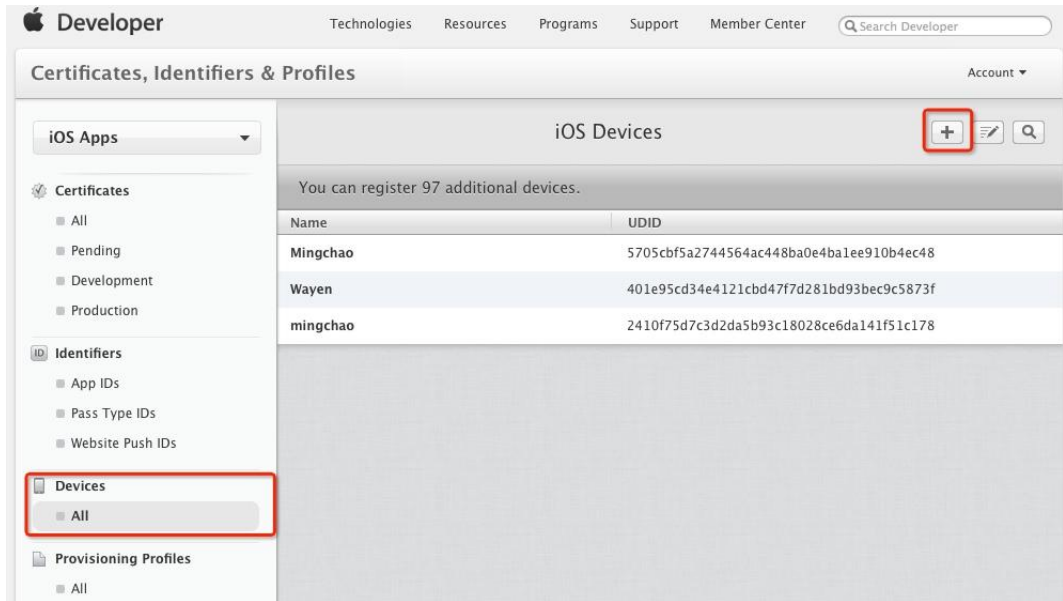
### 1.3、绿灯:说明连接成功，可进行调试

## 2、查看设备的 UDID



## 3、回到 Apple Developer，点击 Devices，再点击右上角的“+”号





4、输入 Name 和 UDID，点击 continue，最后点击 Register



#### Registering a New Device or Multiple Devices

##### Pre-Release Software Reminder

You may only share Apple pre-release software with employees, contractors, and members of your organization who are registered as Apple developers and have a demonstrable need to know or use Apple software to develop and test applications on your behalf.

Unauthorized distribution of Apple confidential information (including pre-release software) is prohibited and may result in the termination of your Apple Developer Program. It may also subject you to civil and criminal liability.

##### Register Device

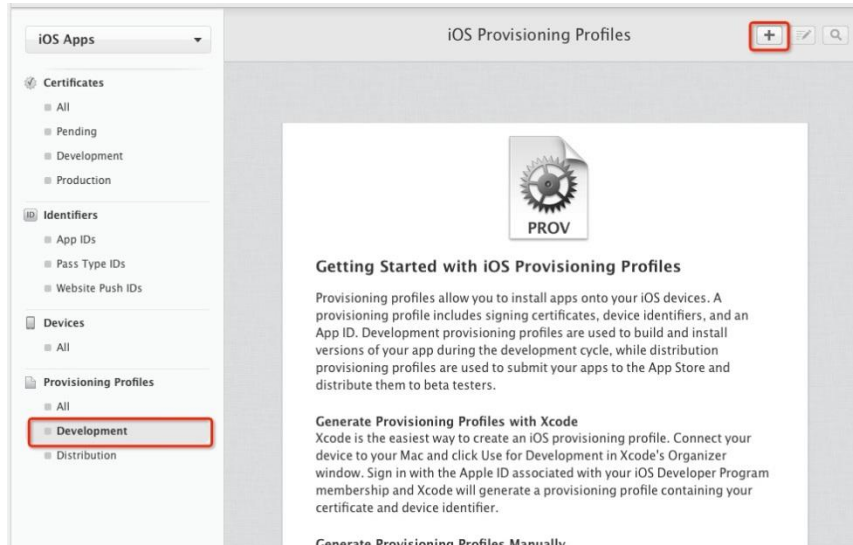
Name your device and enter its Unique Device Identifier (UDID).

Name:

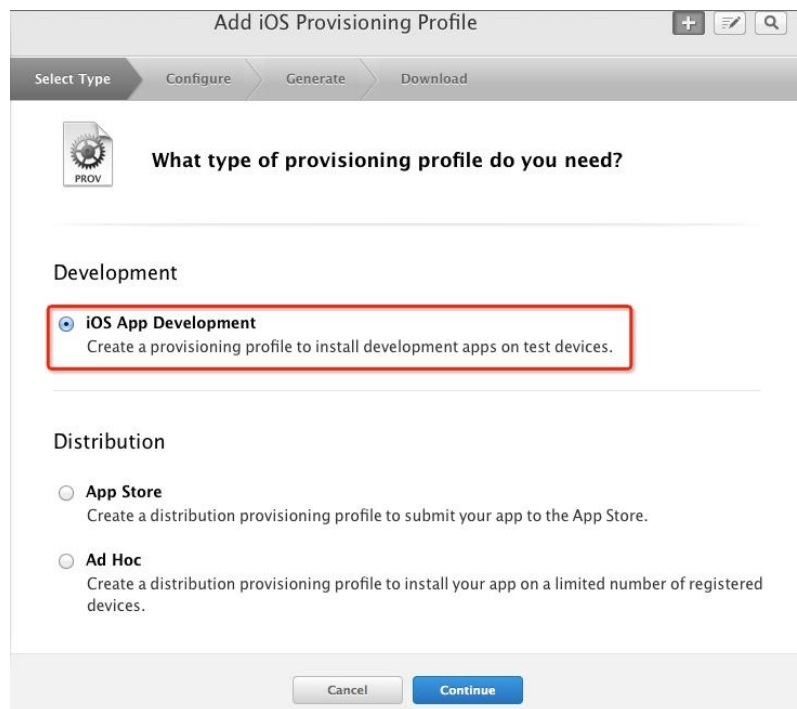
UDID:

## 四、生成 Provisioning Profile

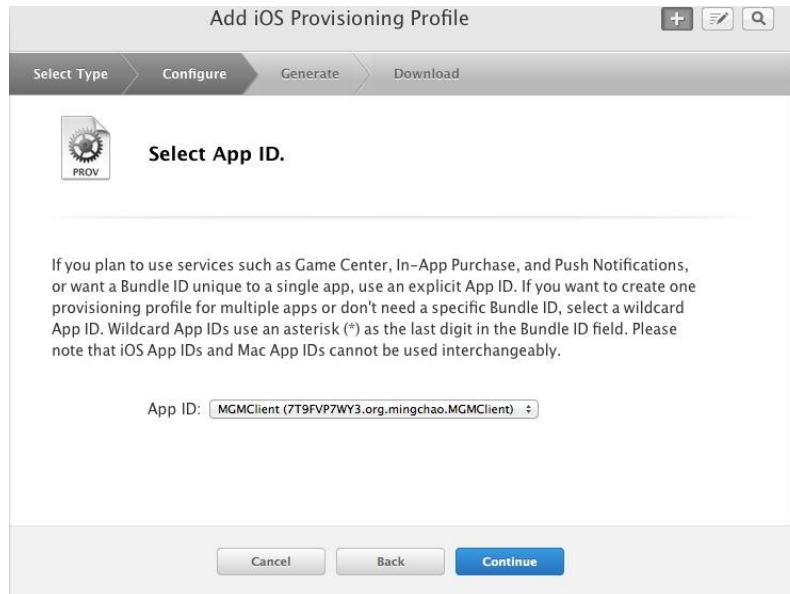
1、点击 Development，点击右上角的“+”



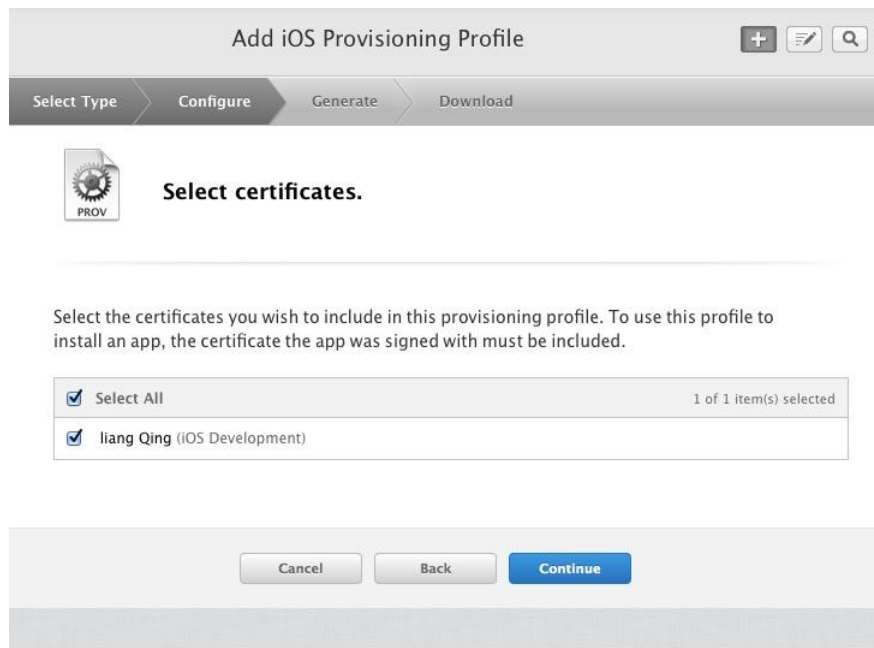
选择 iOS App Development， 点击 Continue



选择我们一开始创建的 APP ID， 点击 Continue



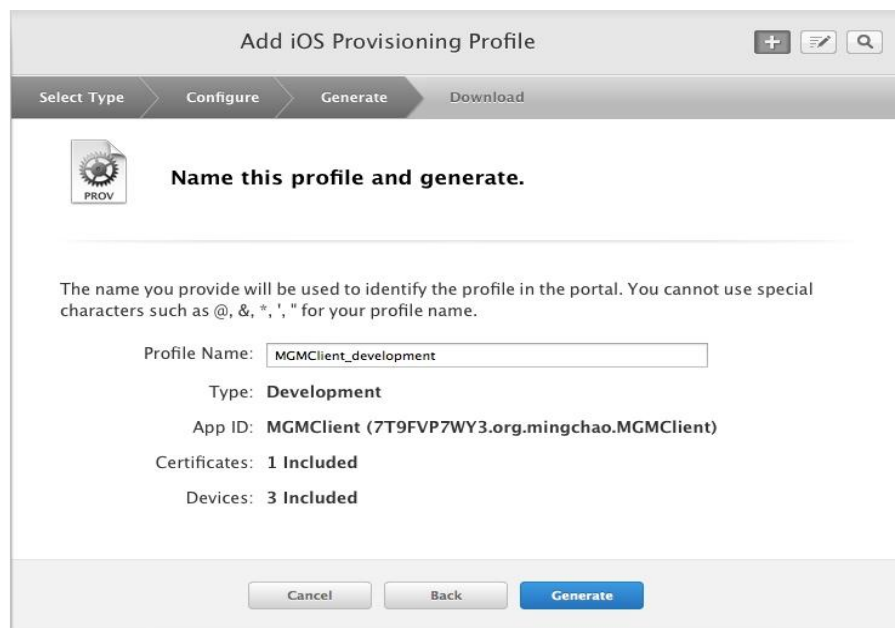
选择刚才创建的调试证书，点击 Continue



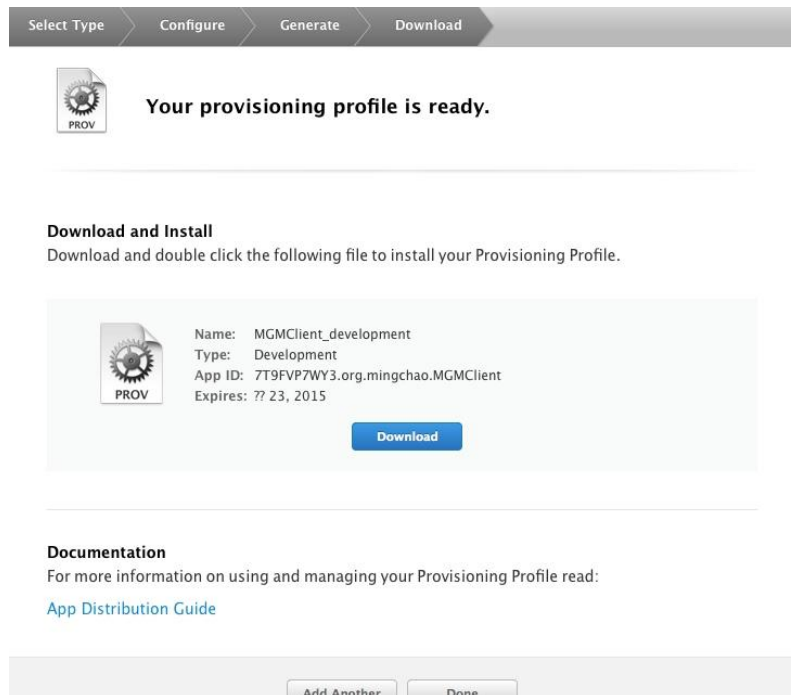
选择添加过的设备，点击 Continue



给这个描述文件起个名字，点击 Generate

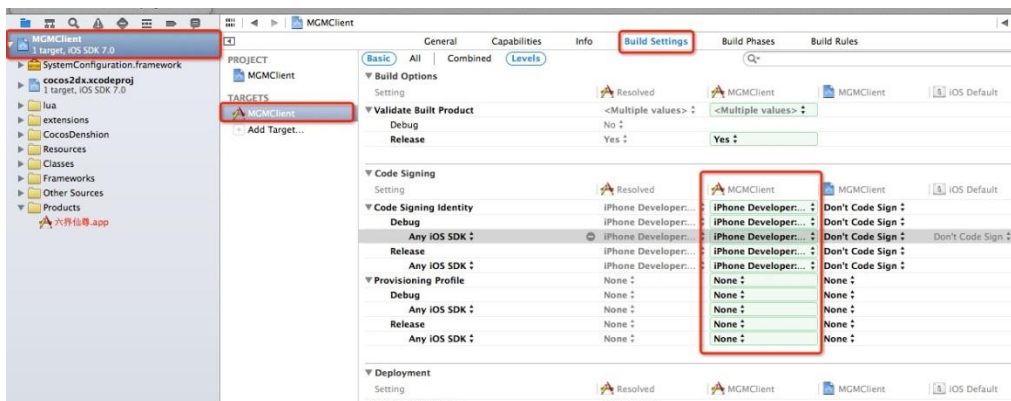


成功提示，点击 Download 下载到本地，双击安装



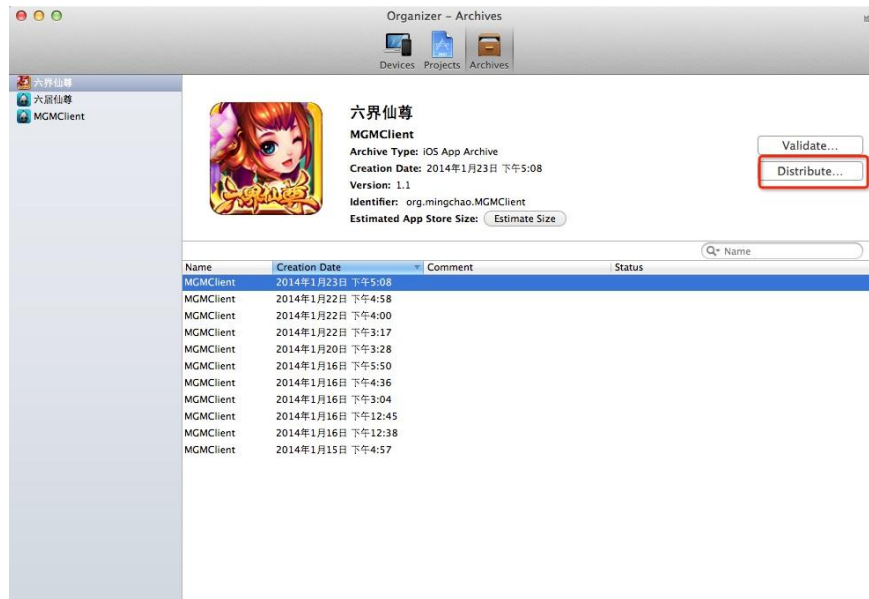
## 五、设置 Xcode Code Sign Identifier

1、xcode 的左边项目导航区点击 MGMCClient.xcodeproj，点击 Targets 的 MGMCClient，点击 Build Settings，选择刚才下载下来的调试证书即可，这样就可以真机调试了

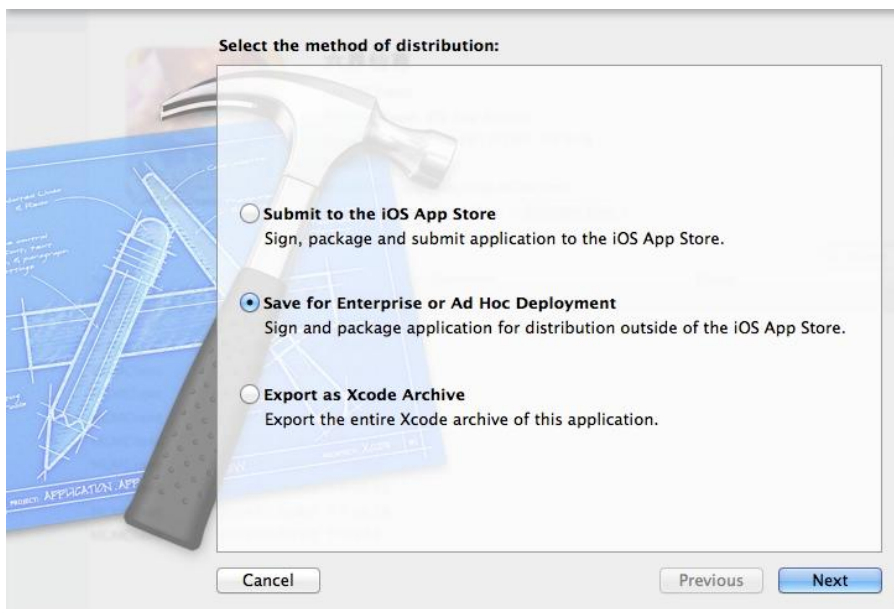


2、导出 ipa 文件

2.1、选择 xcode->Produce->archive,如果没有错误编译完成会见到以下界面，点击 Distribute



然后选择第二项, 点击 next, 选择刚才安装的描述文件, 最后点击 Export 导出 ipa 文件。



## 发布流程

Distribution 主要有以下几个步骤:

- 申请发布证书
- 加入设备
- 生成发布 Provisioning Profile
- 设置 Xcode Code Sign Identifier, 设置 release 选项为发布证书

- 通过 Xcode 来 archive 功能，选择 AdHoc 选项以及发布 Provisioning Profile，最后导出 ipa 安装包文件

发布流程的操作与前面的调试流程基本一样，照着操作即可