

Mac と実機を USB で接続してインストールする

<http://tande.jp/lab/2012/01/1621>

iOS Developer Program に登録

お金払って登録。

Xcode のインストール

AppStore から Xcode をダウンロードして、インストール。

開発証明書の取得と登録

主な流れ

「証明書署名要求ファイル (CSR ファイル)」の生成

- ・開発に使う Mac で、アプリ「キーチェーンアクセス」を起動
- ・キーチェーンアクセスのメニューから「証明書アシスタント > 認証局に証明書を要求 ...」を選択
- ・保存ダイアログが表示されるので、デフォルトの名前のまま CSR ファイル (CertificateSigningRequest.certSigningRequest) を保存
- ・[鍵ペア情報] が表示されるので、[鍵のサイズ] は [2048 ビット] [アルゴリズム] は [RSA] を選択して、[続ける] ボタンをクリック
- ・CSR ファイル (CertificateSigningRequest.certSigningRequest) がローカルに保存される

ローカルに保存した CSR ファイルをアップロード

- ・iOS Developer Program にアクセス
- ・「Member Center」->「iOS Provisioning Portal」を選択
- ・「Certificates」を選択し、「Request Certificate」ボタンをクリック
- ・「参照」ボタンで開いたダイアログで、先程の CSR ファイルを選択し、アップロードする

証明書ファイルをダウンロード

- ・「Certificates」ページの「Action」列のところに「Download」ボタンが表示される
- ・「Download」ボタンをクリックして、証明書ファイル (developer_identifier.cer) をローカルにダウンロード

証明書を Mac にインストール

- ・ダウンロードした証明書ファイルをダブルクリックすると、キーチェーンアクセスが起動して証明書がインストールされる

自分の iPhone,iPad を開発デバイスとして登録

端末の UDID を確認

- ・iTunes を起動します。
- ・iPhone/iPod touch を接続します。
- ・デバイスから iPod を選択します。
- ・シリアル番号が表示されます。「シリアル番号」の文字列をクリックします。
- ・表示が識別子に変わり、40 桁の文字列が表示されます。これが UDID となる

App ID の取得

- ・ iOS Developer Program にアクセスし、App ID を作成
- ・ テストアプリの制作にはワイルドカードの「*」を使った App ID を利用すると

hoge.test.*

など。

一度作った App ID は消す事ができない

Provisioning Profile の作成

- ・ iOS Developer Program にアクセスし「Member Center」の「iOS Provisioning Portal」を開く
- ・ 「Provisioning」を選択し、「Add Profile」ボタンをクリックして Provisioning Profile を登録する

--	--
Profile Name	任意の名前を入力します。
Certificates	チェックを入れます。
App ID	上記で作成したものを選択します。
Device	利用するデバイスにチェックを入れます。

Provisioning Profile を Xcode に登録

- ・ 作成した Provisioning Profile をダウンロード
- ・ Xcode の「Window」から「Organizer」を選択
- ・ 「Organizer」ウィンドウを立ち上げ、「Deveices」タブを選択
- ・ 左サイドバーの「Provisioning Profiles」を選択し、「Automatic Device Provisioning」がチェックされていることを確認
- ・ Provisioning Profile が自動的にダウンロードされて登録されているはず
- ・ 何も登録されていない場合は、「Refresh」ボタンを押す

iPhone 実機へサンプルアプリをインストール

- ・ Simulator の種類を選ぶプルダウンリストに、USB 接続されたデバイス名がある
- ・ ない場合は、対象の OS のバージョンを変えたりしてみる

Web サーバからアプリをインストールする

<http://kaigian.co.jp/taka/2013/ota.html>

<http://plus.vc/iphone/3355/>

OTA (Over The Air) または in house と呼ばれる方法で実機にアプリをインストールする。

Provisioning Profiles の作成

iOS Provisioning Portal で Provisioning Profiles を作成する。
このとき、Distribution Method を「in House」にしておく。

ビルド設定

Xcode のビルド設定を変更する。

Code Signing を Provisioning Portal で作った Distribution に設定。

ビルド

GUI でビルドする

Xcode の「Product」->「Archive」でアプリをアーカイブする。

この時、実行先をシミュレータではなく iOS Device を選択していないと、Archive を選択できない。

ビルド実行後、Organizer が起動する。

Distribute を選択しアプリのファイルを作る。

このとき「Save for Enterprise Distribution」を選択すると plist ファイルを生成できる。

URL は、「<http://localhost/test/hoge.ipa>」のようなアプリの URL を指定する。

コマンドラインでビルドする

http://d.hatena.ne.jp/ryo_abe/20120226/1330242142

http://atkonn.blogspot.jp/2012_06_03_archive.html

コマンドでビルドするには、以下のようなシェルを作っておくと良い。

SDK の一覧を取得する（プロジェクトのルートディレクトリで実行）

```
xcodebuild -showsdk
```

ターゲットの一覧を取得する（プロジェクトのルートディレクトリで実行）

```
xcodebuild -list
```

User interaction is not allowed って出る場合

```
security unlock-keychain /Users/funa/Library/Keychains/login.keychain
```

でキーチェーンのロックを解除しておく必要がある。

```
#!/bin/sh

#SDK
SDK="iphoneos5.0"

# コンフィグレーション（「Debug」、「Release」、「Ad hoc」）
CONFIGURATION="Release"

# Xcode のプロジェクト名
PROJ_FILE_PATH="hoge.xcodeproj"

# ターゲット名
TARGET_NAME="hogeTarget"

# 「Build Settings」にある、プロダクト名
PRODUCT_NAME="hogeProduct"

# app 出力先ディレクトリ名
OUT_APP_DIR="out_app"

# 出力先 ipa ディレクトリ名
OUT_IPA_DIR="out_ipa"
```

```

# 出力される ipa ファイル名
IPA_FILE_NAME="hogeipa"

# ライセンス取得時の開発者名
DEVELOPPER_NAME="iPhone Distribution: hoge Developer"

# アプリのプロビジョニングファイルのパス
PROVISIONING_PATH="${HOME}/Library/MobileDevice/Provisioning Profiles/hoge.mobileprovision"

# 出力先 ipa ディレクトリ作成
# -----
if [ ! -d ${OUT_IPA_DIR} ]; then
    mkdir "${OUT_IPA_DIR}"
fi

# クリーン
# -----
xcodebuild clean -project "${PROJ_FILE_PATH}"

# ビルド
# -----
xcodebuild -project "${PROJ_FILE_PATH}" -sdk "${SDK}" -configuration "${CONFIGURATION}" -target
"${TARGET_NAME}" install DSTROOT="${OUT_APP_DIR}"

# Create ipa File
# -----
xcrun -sdk "${SDK}" PackageApplication "${PWD}/${OUT_APP_DIR}/Applications/${PRODUCT_NAME}.app" -o
"${PWD}/${OUT_IPA_DIR}/${IPA_FILE_NAME}.ipa" -embed "${PROVISIONING_PATH}"

```

plist

自動生成していない場合は以下の様な XML を plist ファイルとして作成する

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0
.dtd">
<plist version="1.0">
<dict>
<key>items</key>
<array>
<dict>
<key>assets</key>
<array>
<dict>
<key>kind</key>
<string>software-package</string>
<key>url</key>
<string>https://dl.dropboxusercontent.com/u/1234567/MyApp.ipa</string>
</dict>
</array>
<key>metadata</key>
<dict>
<key>bundle-identifier</key>
<string>com.exsample.MyApp</string>
<key>bundle-version</key>
<string>1.0</string>
<key>kind</key>
<string>software</string>
<key>title</key>
<string>Exsample</string>
</dict>
</dict>
</array>
</dict>
</plist>

```

Web サーバに設置する

配布用 html の作成

ファイルの設置

- index.html
- アプリファイル (MyApp.ipa)
- plist ファイル (MyApp.plist)

を Web サーバに設置する。

インストール

実機で index.html にアクセスすればインストールされるはず。