

Swiftで QuickLookとSpotlightの 実装

mindtools@mac.com

本当のテーマ

- C/Objective-CからSwift関数を呼び出すには？
- Objective-Cのクラス名とSwiftのクラス名の相互運用
- プラグインの中でのswiftコードの挙動の確認

結論

- C/Objective-CからSwift関数を呼び出すには？
- Objective-Cのクラス名とSwiftのクラス名の相互運用
- プラグインの中でのswiftコードの挙動の確認

C/Objective-CからSwift関 数を呼び出すには？

- `@_cdecl`(“関数名(シンボル名)”)を関数の1行前に書く

[illegible]

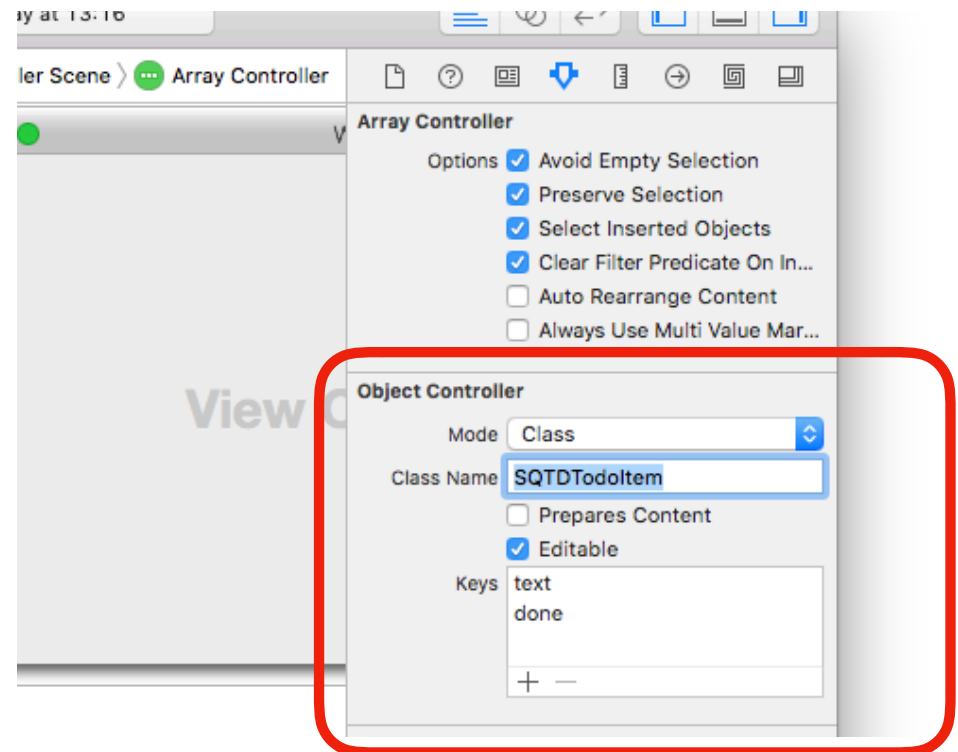
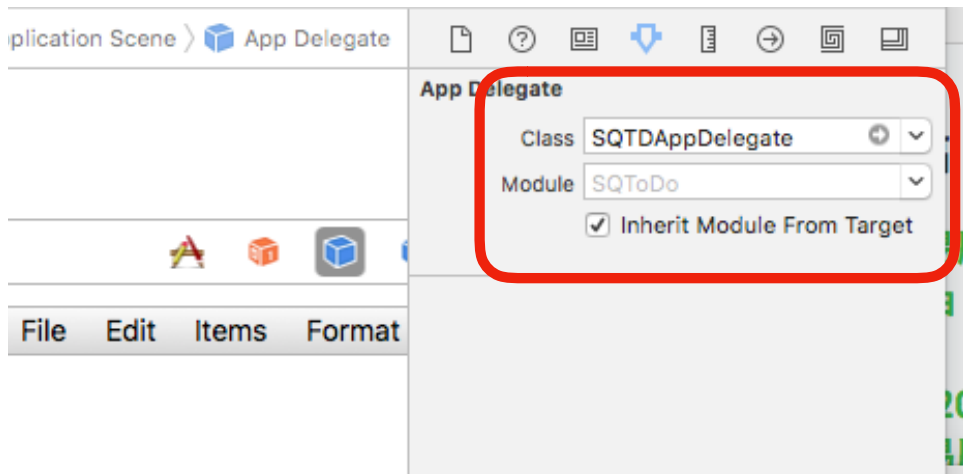
Objective-Cのクラス名と Swiftのクラス名の相互運用

- @objc(クラス名)をクラスの1行前に書く

```
@objc(SQTDModel)
class SQTDModel: NSObject, NSCoding
{
    // MARK: - class methods
```

Objective-Cのクラス名と Swiftのクラス名の相互運用

- nibファイル(とcoreData)の中のクラス名に注意を払う



プラグインの中でのswiftコードの挙動の確認

- autoreleasepoolで囲むこと、メモリ管理は自動じゃない

```
autoreleasepool  
{  
  
}
```

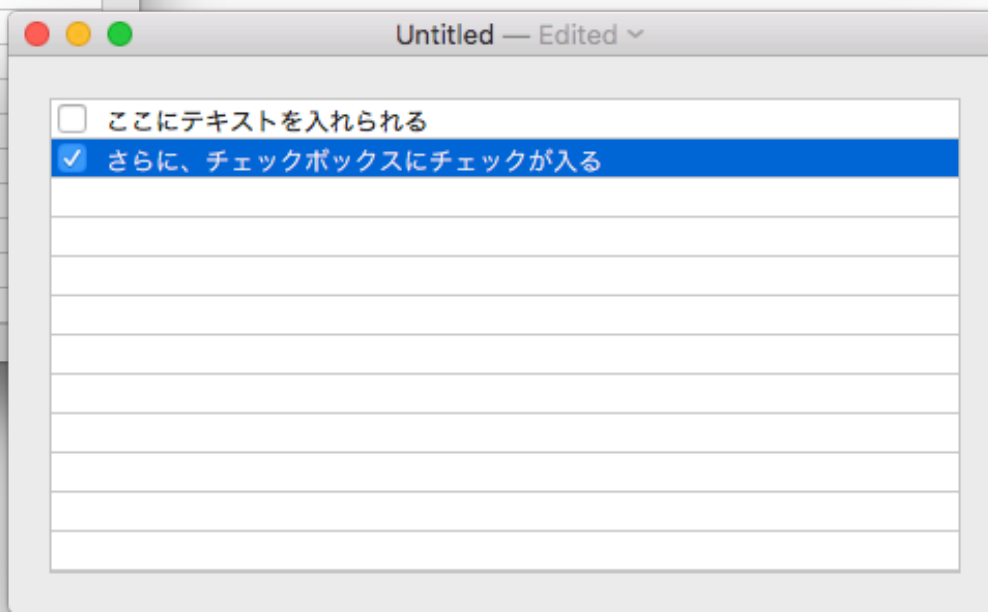
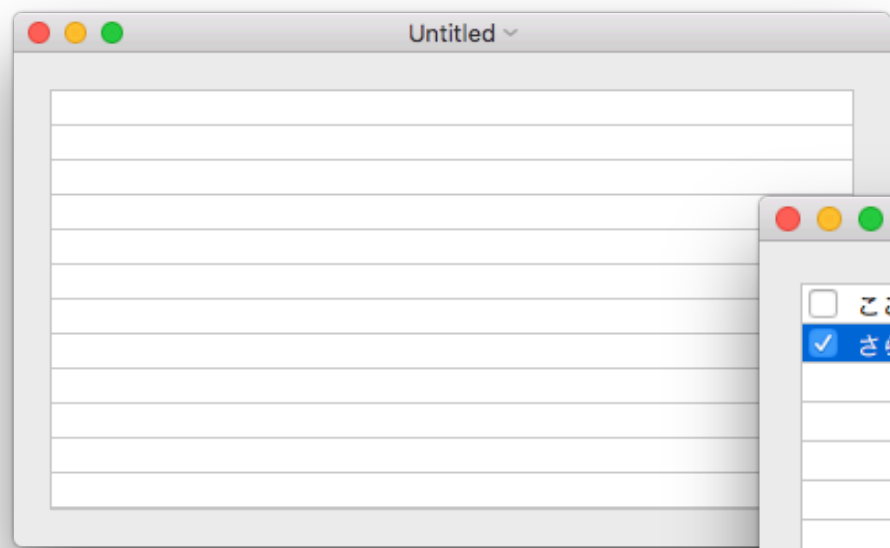
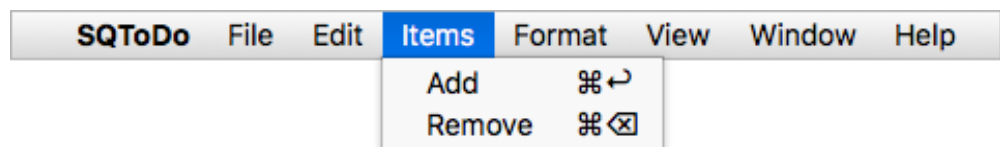
プラグインの中でのswiftコードの挙動の確認

- プラグインはメインバンドルではない。
 - メインバンドルは、プラグインを呼び出しているバイナリ
 - 専用のバンドル取得関数を利用すること
 - 例外をプラグインの外に投げるな。

```
QLThumbnailRequestGetGeneratorBundle(thumbnail).takeRetainedValue()
```


サンプルアプリケーションの 内部構造

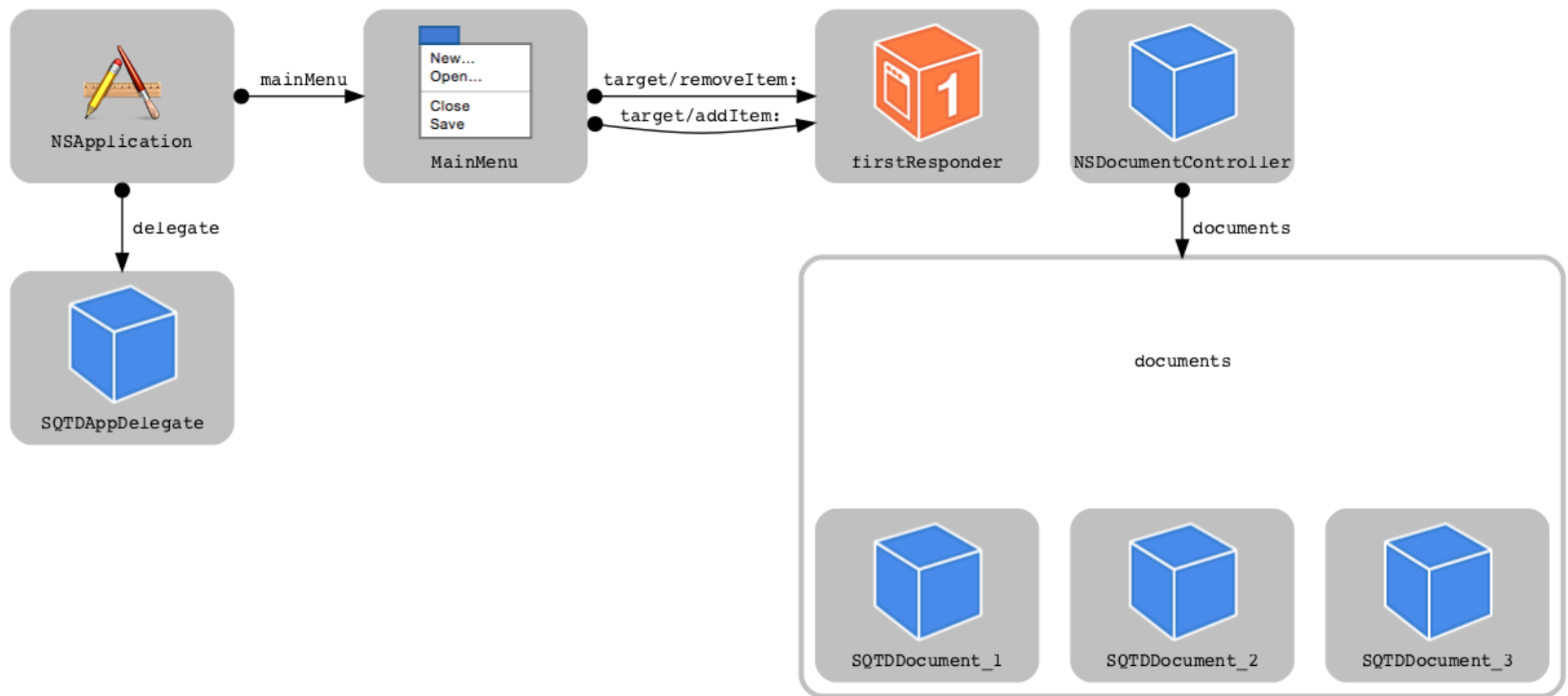
- ToDo Listを作るアプリケーション
- ドキュメントベースで、ファイルにドキュメントの内容を独自フォーマットで読み書きできる。
- 特有の機能は2つ。ToDoアイテムの作成/削除



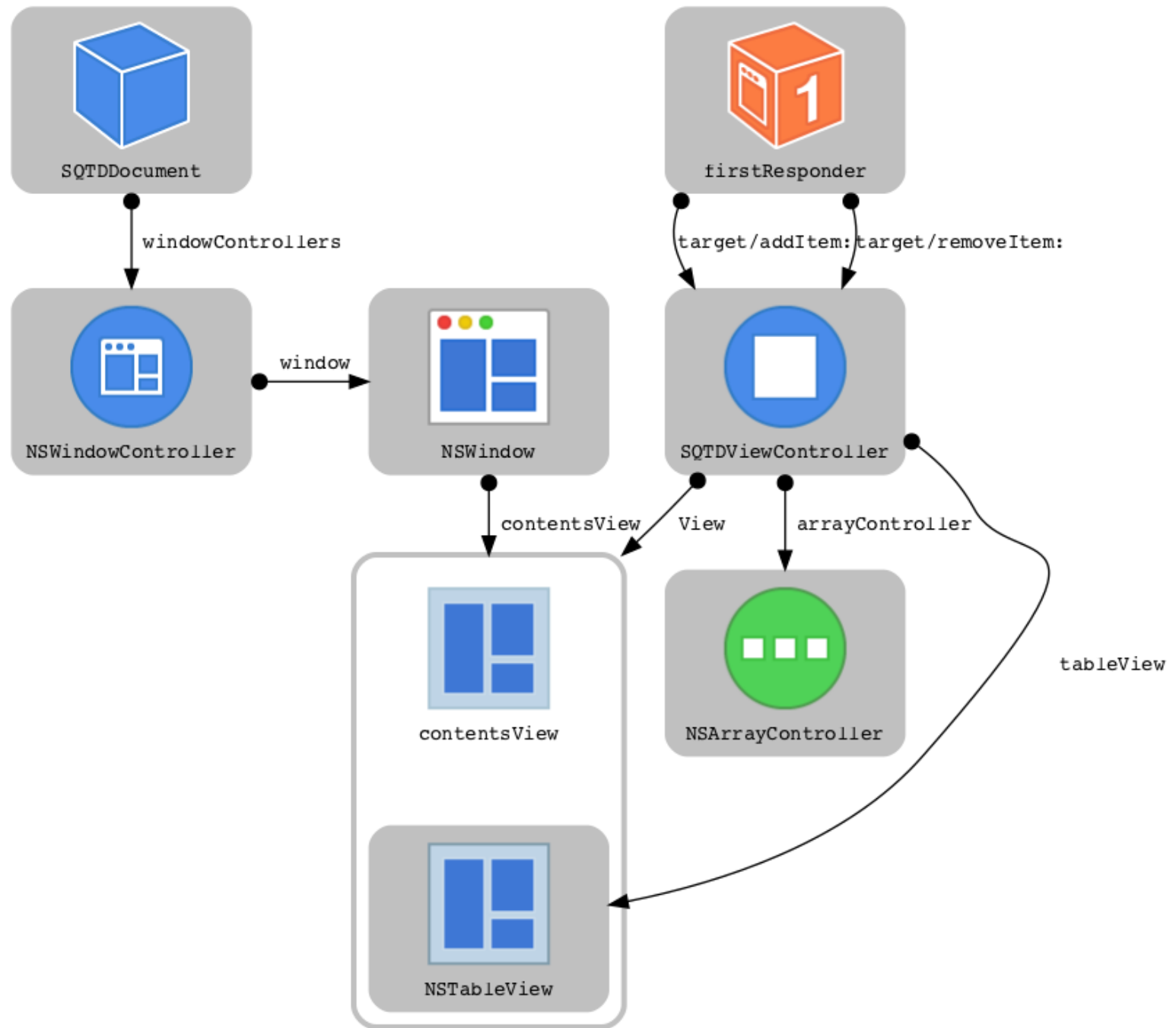
クラス一覧

クラス名	目的
SQTDAppDelegate	アプリケーションデリゲート。何もしてません。
SQTDDocument	ファイルからモデルの保存と再生とモデルとWindowとの結びつけを担当しています。
SQTDModel	モデルクラスです。ToDoItemの配列を持ち。NSDataへの保存と再生が出来ます。
SQDToDoItem	モデルクラスです。1個のToDoItemをあらわします。
SQTDViewController	コントローラーです。UI上での追加と削除コマンドを受け持ちます。

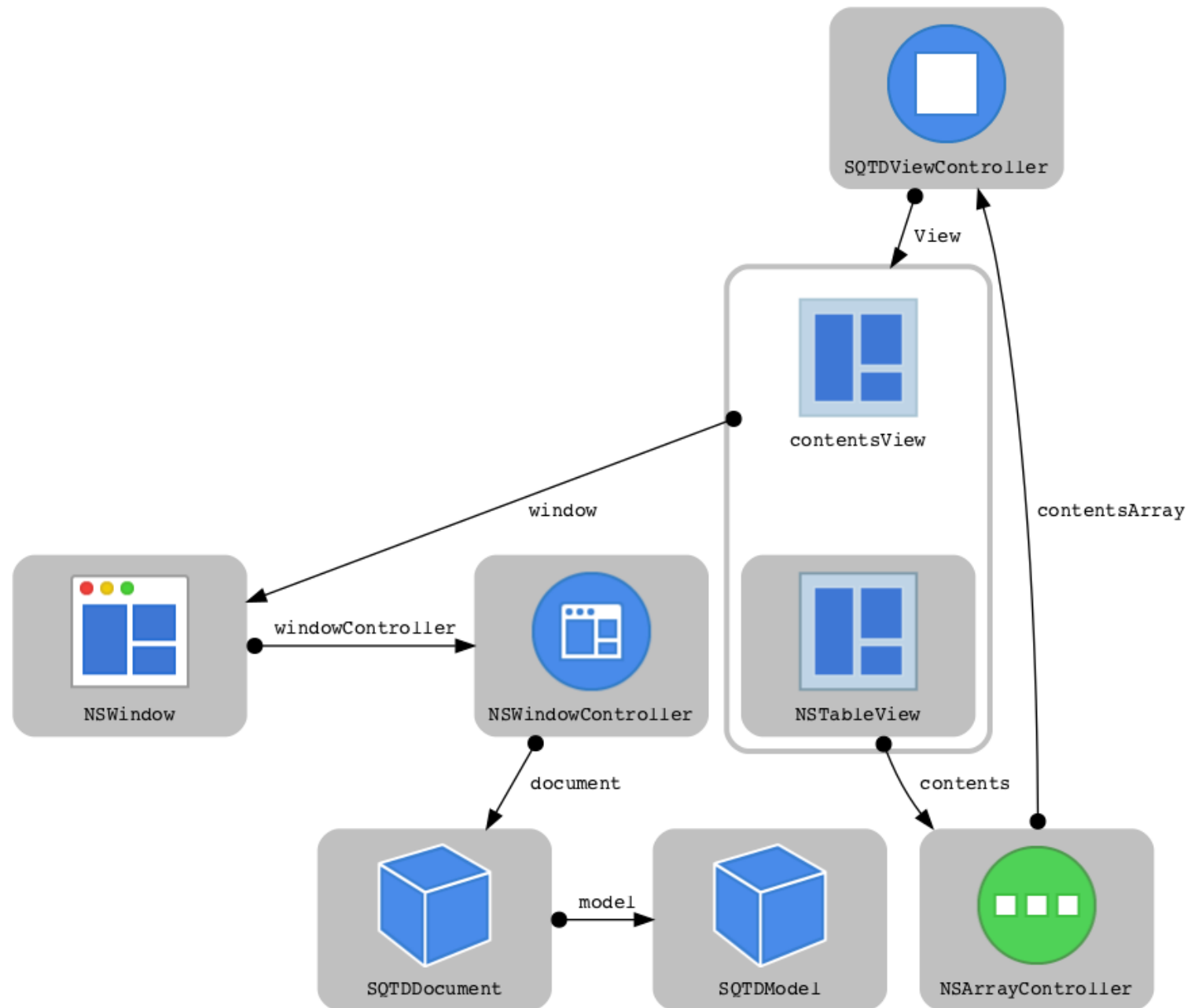
1. SQTDAppDelegateを中心にシングルトンを含めた図



2. SQTDDocumentを中心にViewを含めた図



3. Binding ☒



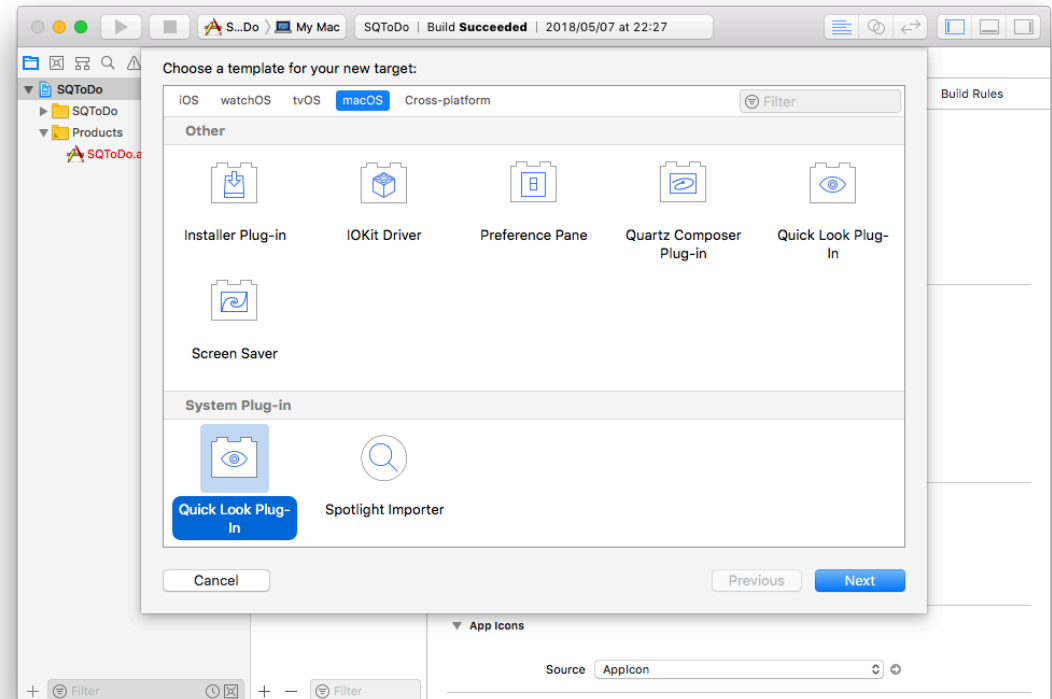
QucikLookとSpotlight のテンプレート

- ややこしい部分は、全部入り。
- Swiftのコンパイルとリンク
- Info.plistの雛形
- 関数の雛形

インストール方法

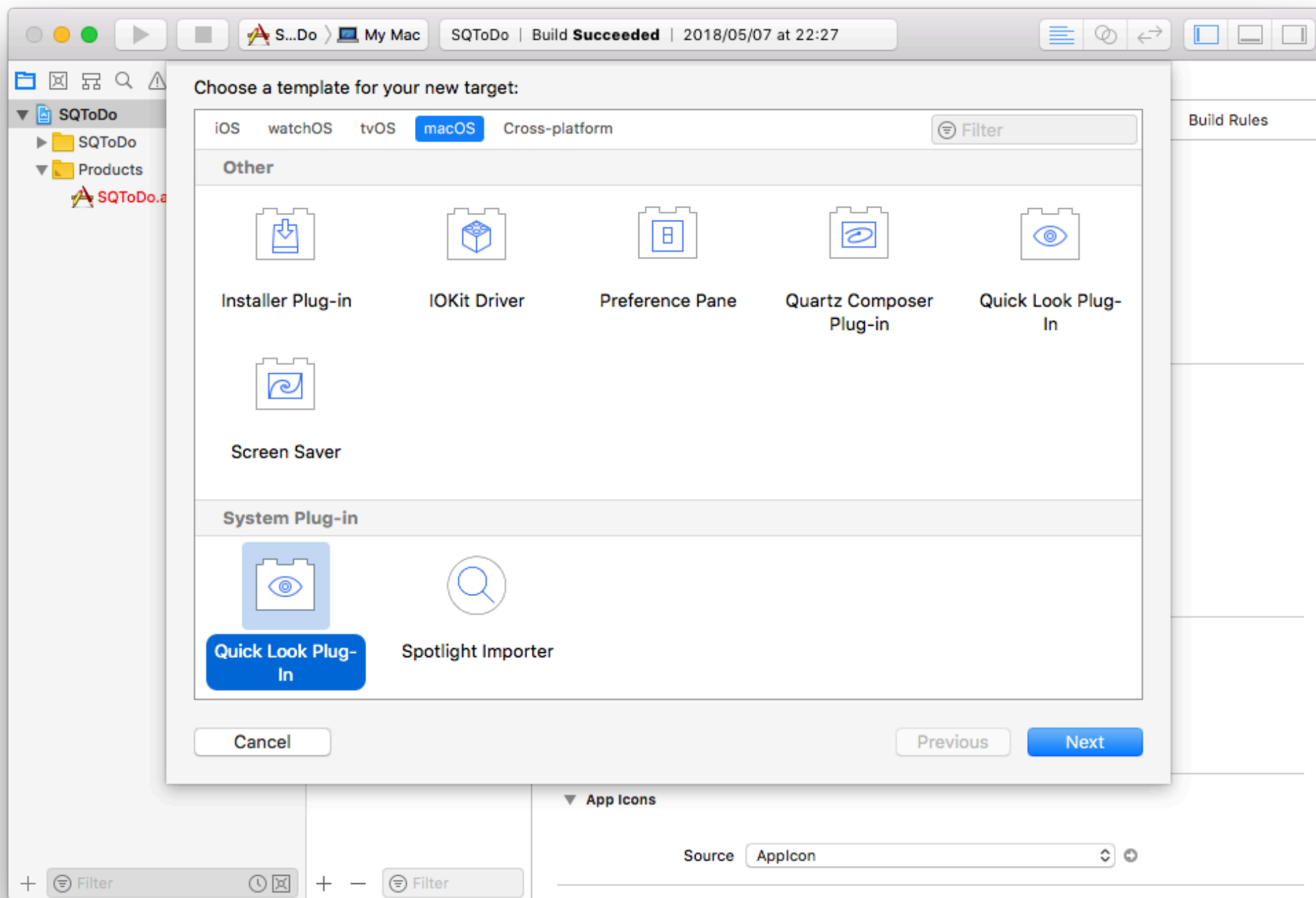
- Mac.zipを展開する
- “~/Library/Developer/Xcode/Templates/”にコピーする

```
$ tree ~/Library/Developer/Xcode/Templates/  
~/Library/Developer/Xcode/Templates/  
├── File\ Templates  
├── Framework\ &\ Library  
├── Project\ Templates  
├── Mac  
│   ├── System\ Plug-in  
│   │   └── Quick\ Look\ Plug-In.xctemplate  
│   │       ├── GeneratePreviewForURL.swift  
│   │       ├── GenerateThumbnailForURL.swift  
│   │       ├── TemplateIcon.png  
│   │       ├── TemplateIcon@2x.png  
│   │       ├── TemplateInfo.plist  
│   │       └── main.c  
│   └── Spotlight\ Importer.xctemplate  
│       ├── GetMetadataForFile.swift  
│       ├── TemplateIcon@2x.png  
│       ├── TemplateInfo.plist  
│       └── main.c
```

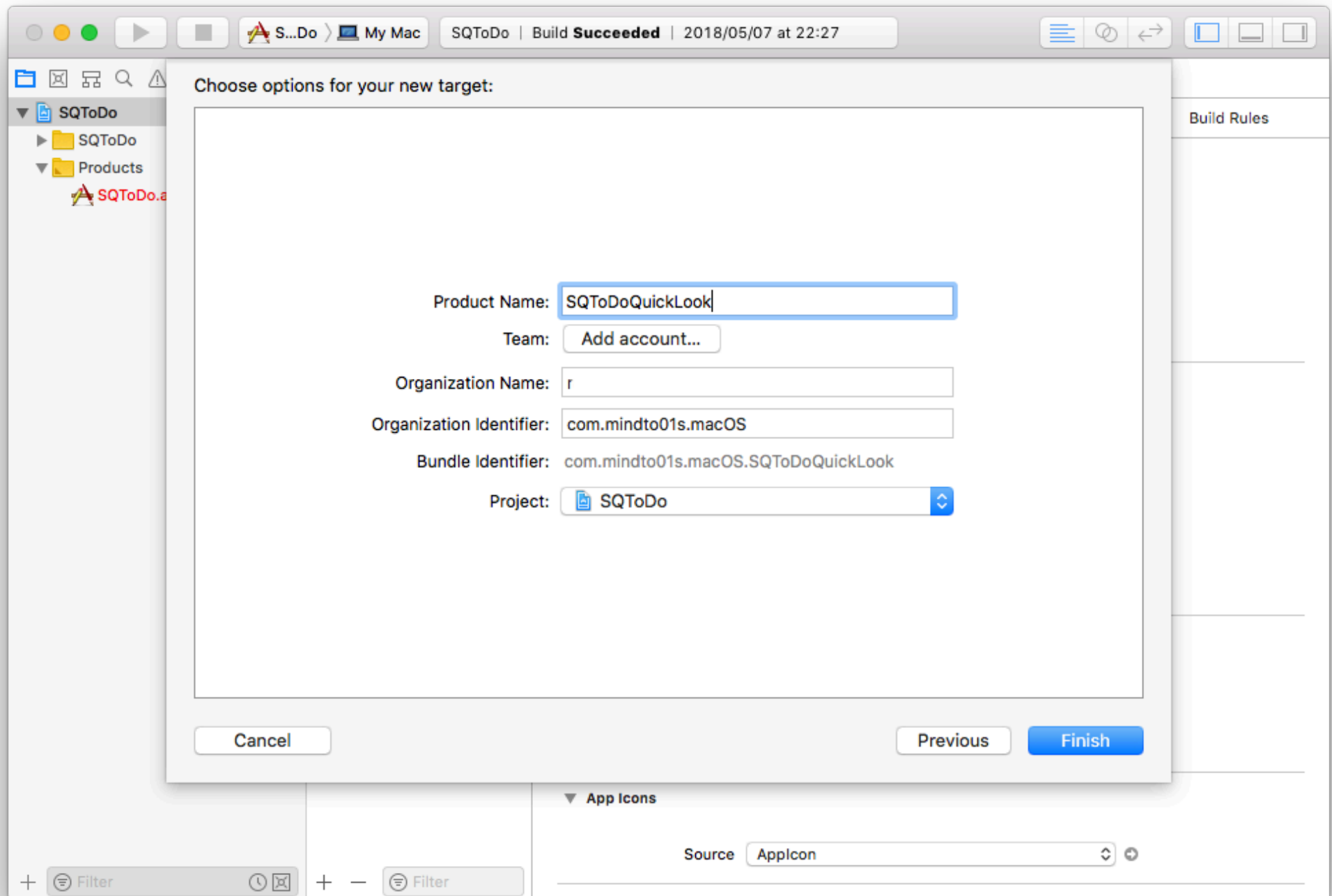


QucikLookの作成

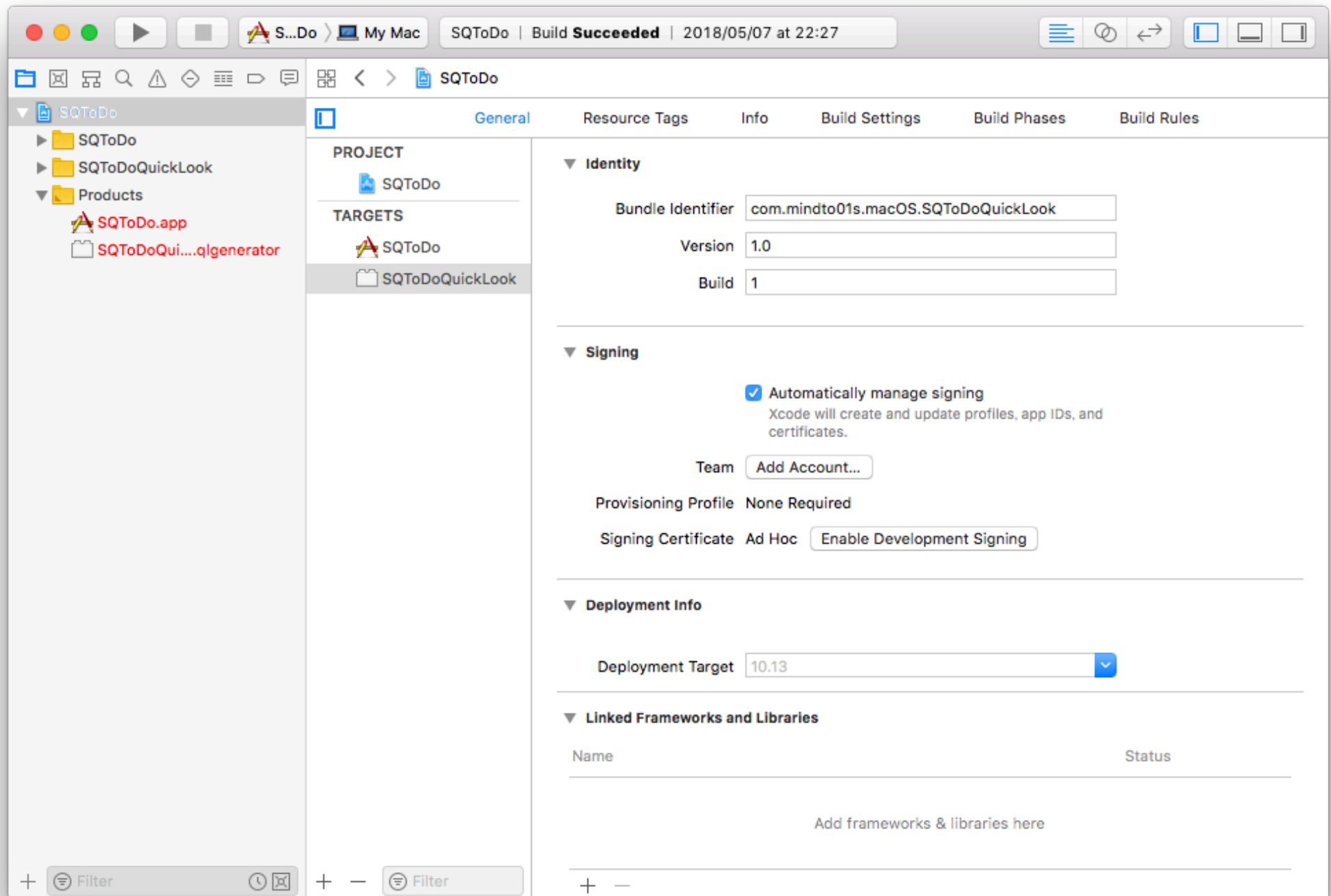
選んで



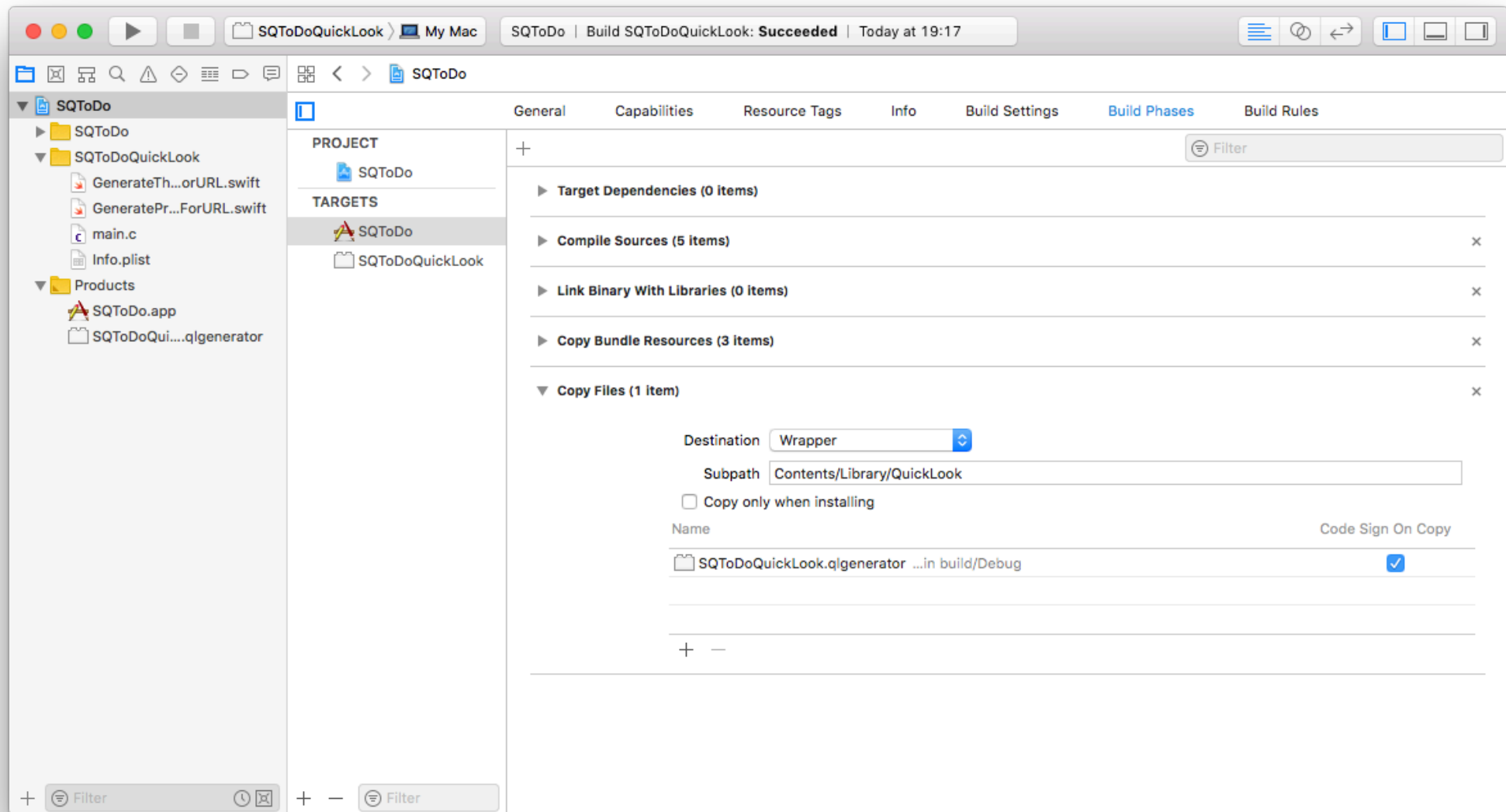
名付けて



こうなる



ToDoアプリに埋め込んで



テンプレートで作成されたファイルの役割

ファイル名	役割
GenerateThumbnailForURL.swift	サムネイル画像を生成させるための関数のプレースフォルダ。ここにコードを追加する。
GeneratePreviewForURL.swift	プレビュー画像を生成させるための関数のプレースフォルダ。ここにコードを追加する。
main.c	変更するな、と書いているので従うこと。pluginとして動作させるためのコードが入っている。
Info.plist	コード以外の各種設定を、実行時にmainBundleに与える。いつものInfo.plist。

info.plistの設定

```
<key>CFBundleDocumentTypes</key>  
  <array>  
    <dict>  
      <key>CFBundleTypeRole</key>  
      <string>QLGenerator</string>  
      <key>LSItemContentTypes</key>  
      <array>  
        <string>com.mindTo01s.macOS.sqtdDoc</string>  
      </array>  
    </dict>  
  </array>
```

他の値は基本は無視してOK

最小限の動作確認

プレビューもサムネイルも作成されないが、実際にコードが動くかを確認することが出来る。
qlmanageコマンドを使う。

デバッグに必要なqlmanageの引数オプション

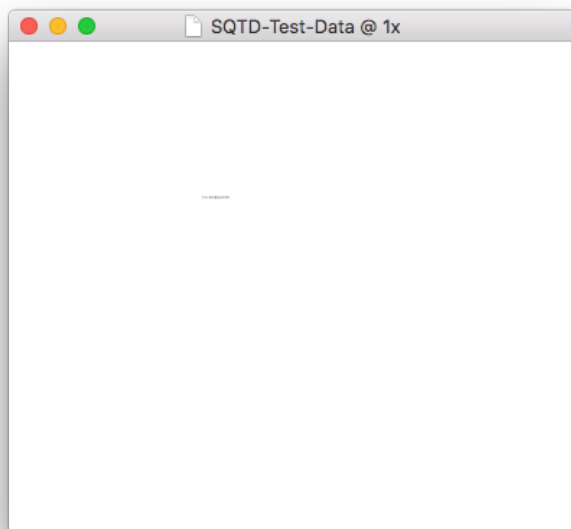
キー	役割
-t	サムネイル画像を生成させる時に与える。元データのファイルパスを取る
-p	プレビュー画像を生成させるときに与える。元データのファイルパスを取る
-c	UTI形式のタイプを指定する。ここでは、上の方で設定した'com.mindto01s.ToDoList.XXXXX'を指定
-g	ビルドしたプラグインのパスを指定します。

例えば、以下のように使います。

```
qlmanage -g path/to/SQToDoQuickLook.qlgenerator -c com.mindTo01s.macOS.sqtdDoc -t アプリで作ったファイル
```

実際に実行してみると、以下のような動作になる。サムネイル関数の実行。

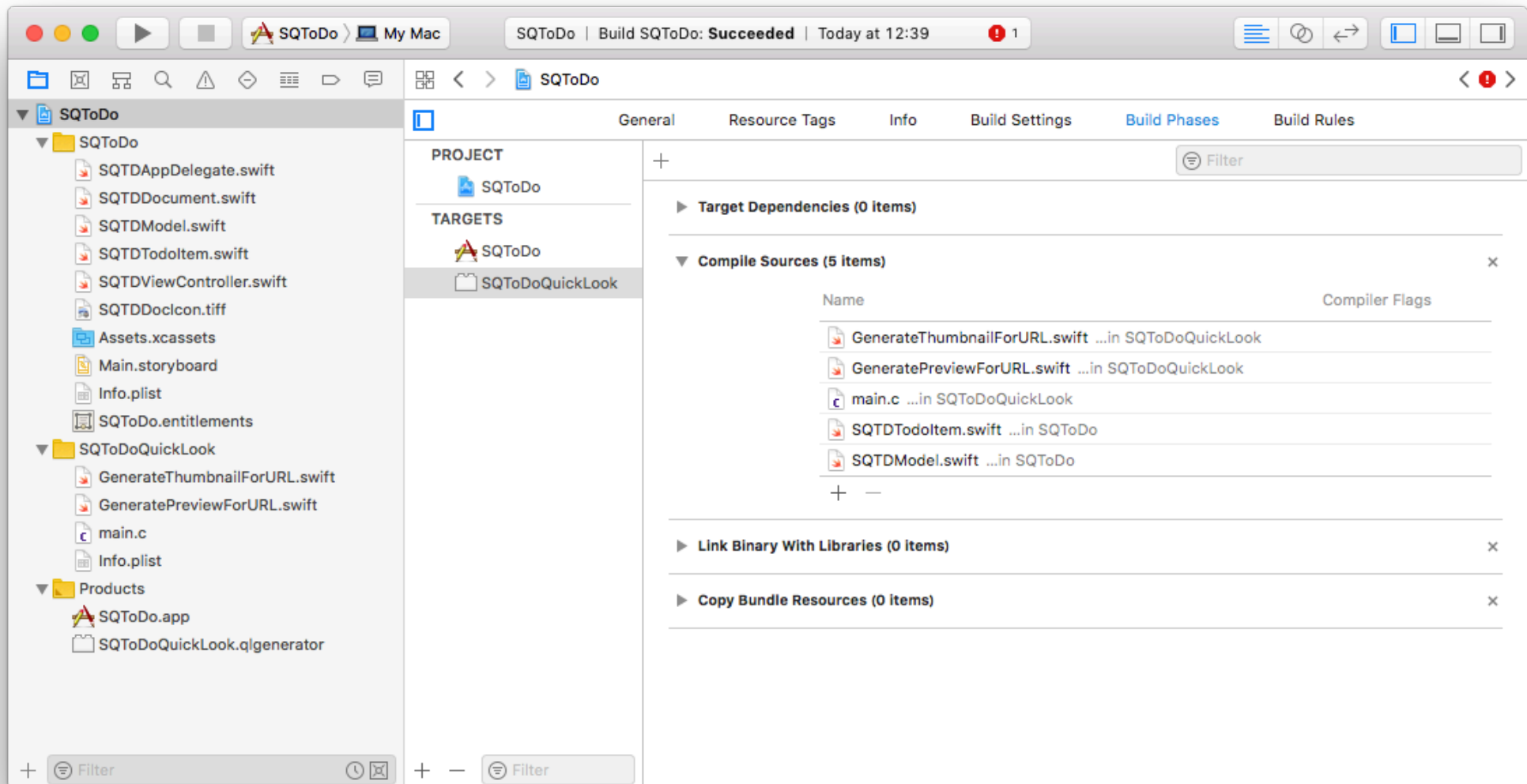
```
$qlmanage -g path/to/SQToDoQuickLook.qlgenerator -c com.mindTo01s.macOS.sqtdDoc -t path/to/アプリで作ったファイル.sqtd
Testing Quick Look thumbnails with files:
  path/to/アプリで作ったファイル.sqtd
  - force using content type UTI: com.mindTo01s.macOS.sqtdDoc
  - force using generator at path: path/to/SQToDoQuickLook.qlgenerator
Thumbnail in Swift <--- ソースコード上に書いたログ出力が出てればOK
* path/to/SQTD-Test-Data.sqtd produced one thumbnail
Done producing thumbnails
```



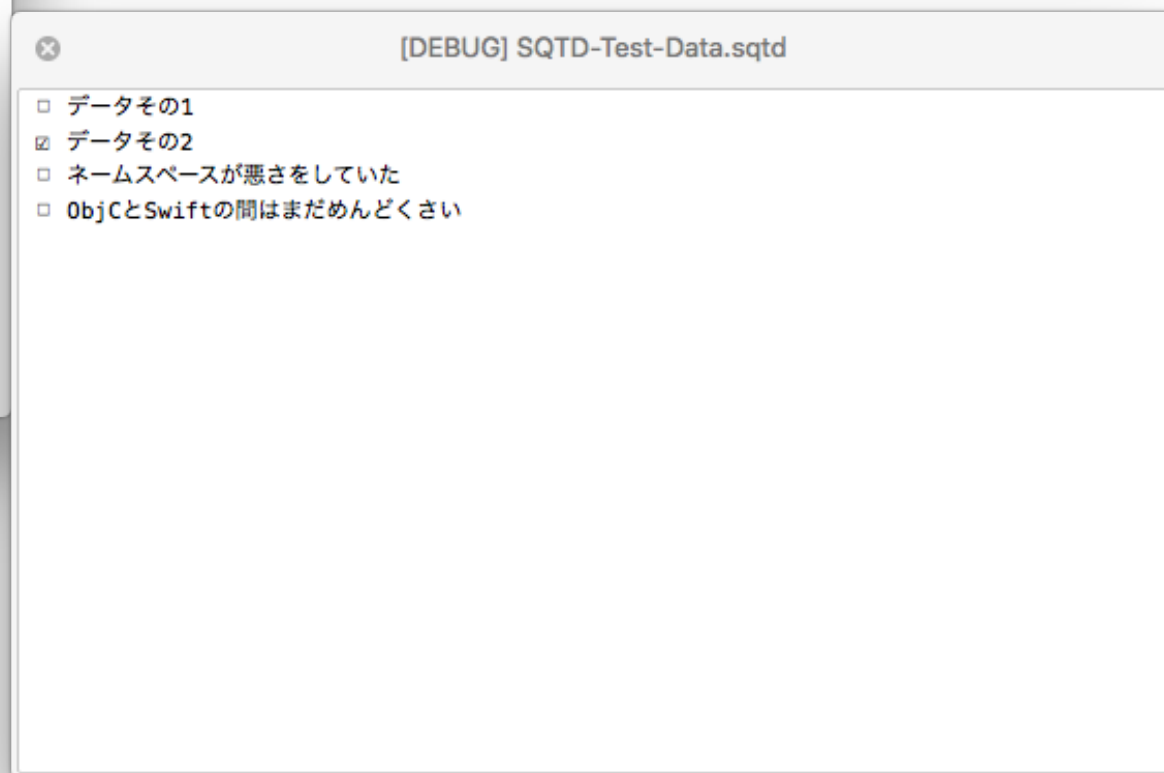
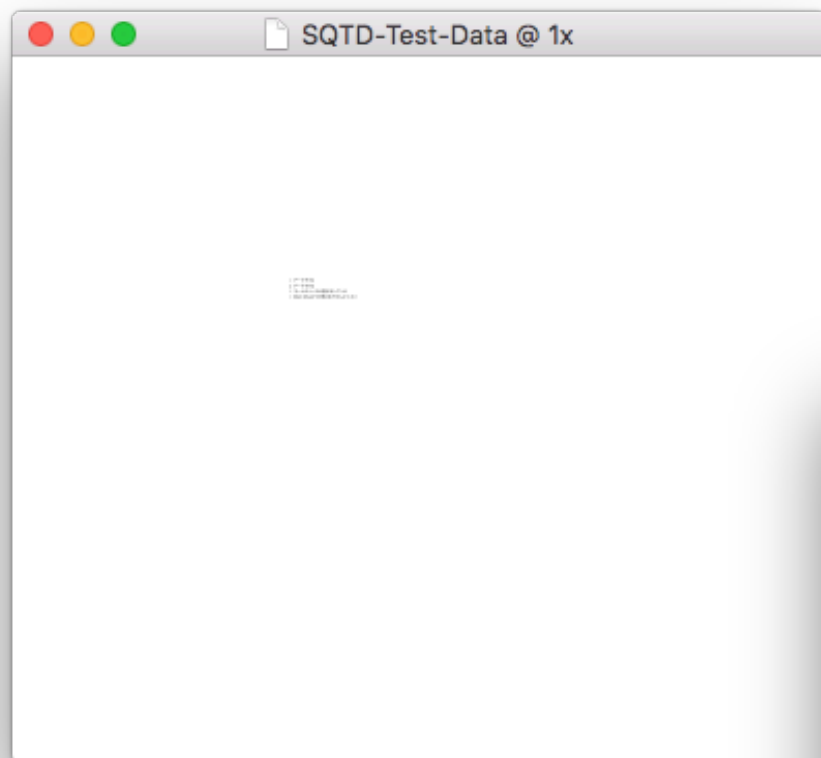
ソースコードの変更して、実際に動作できるようにする。

ファイルからモデルデータに戻す為に、以下の2つのファイルをQuickLookターゲットに追加します。

- SQTModel.swift
- SQTDoItem.swift

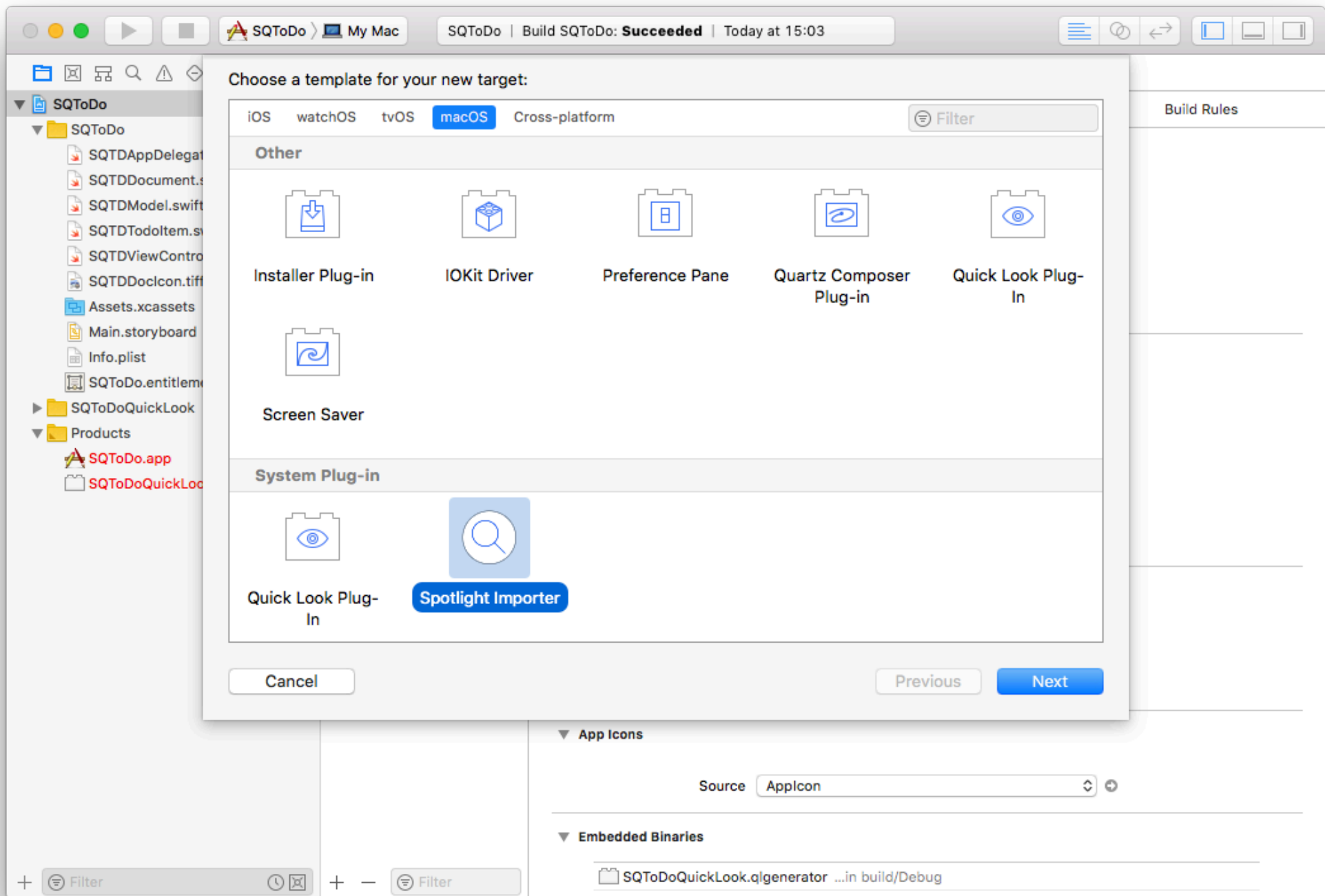


ソースコードの解説します。添付のコードを見てね

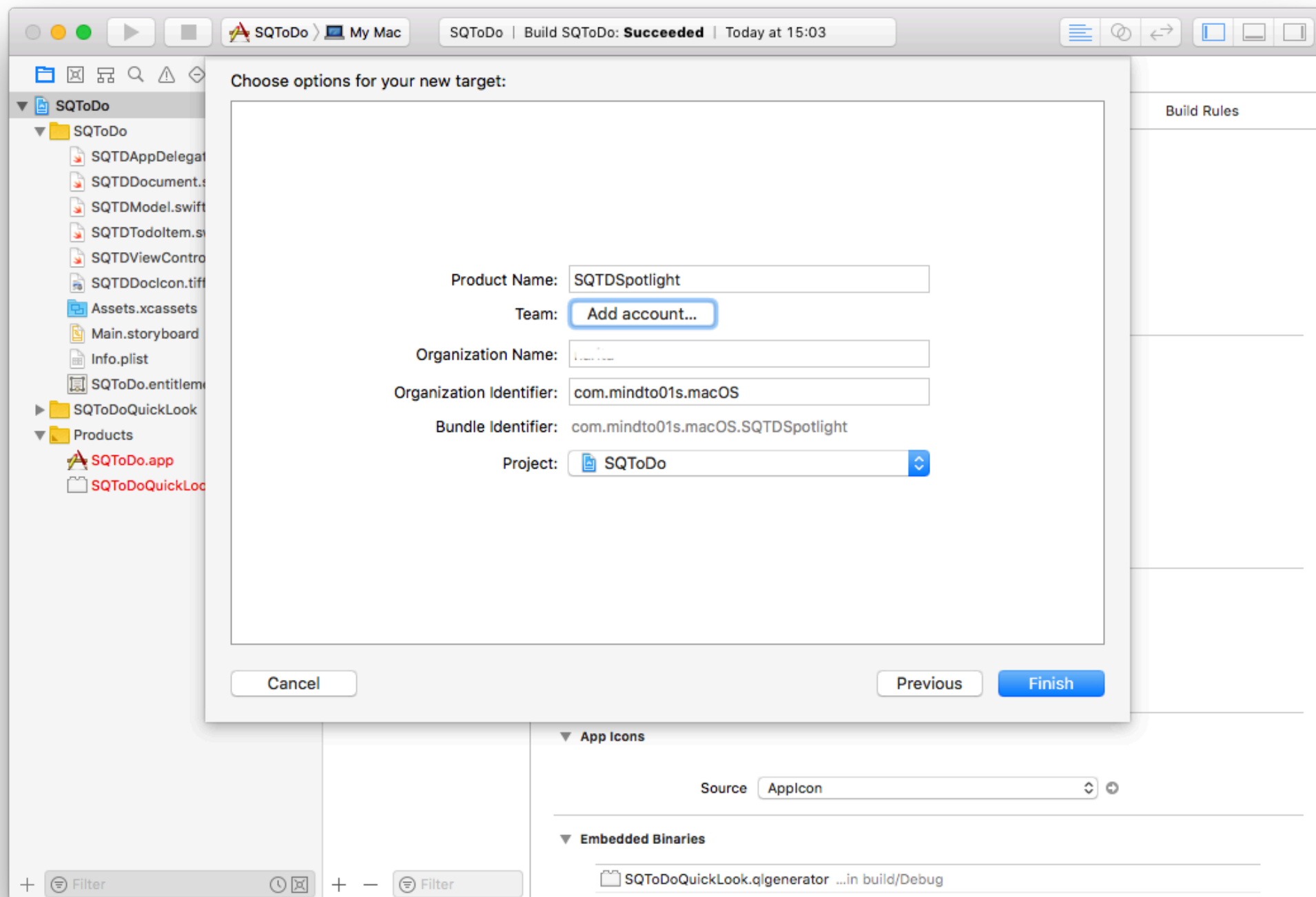


Spotlightの作成

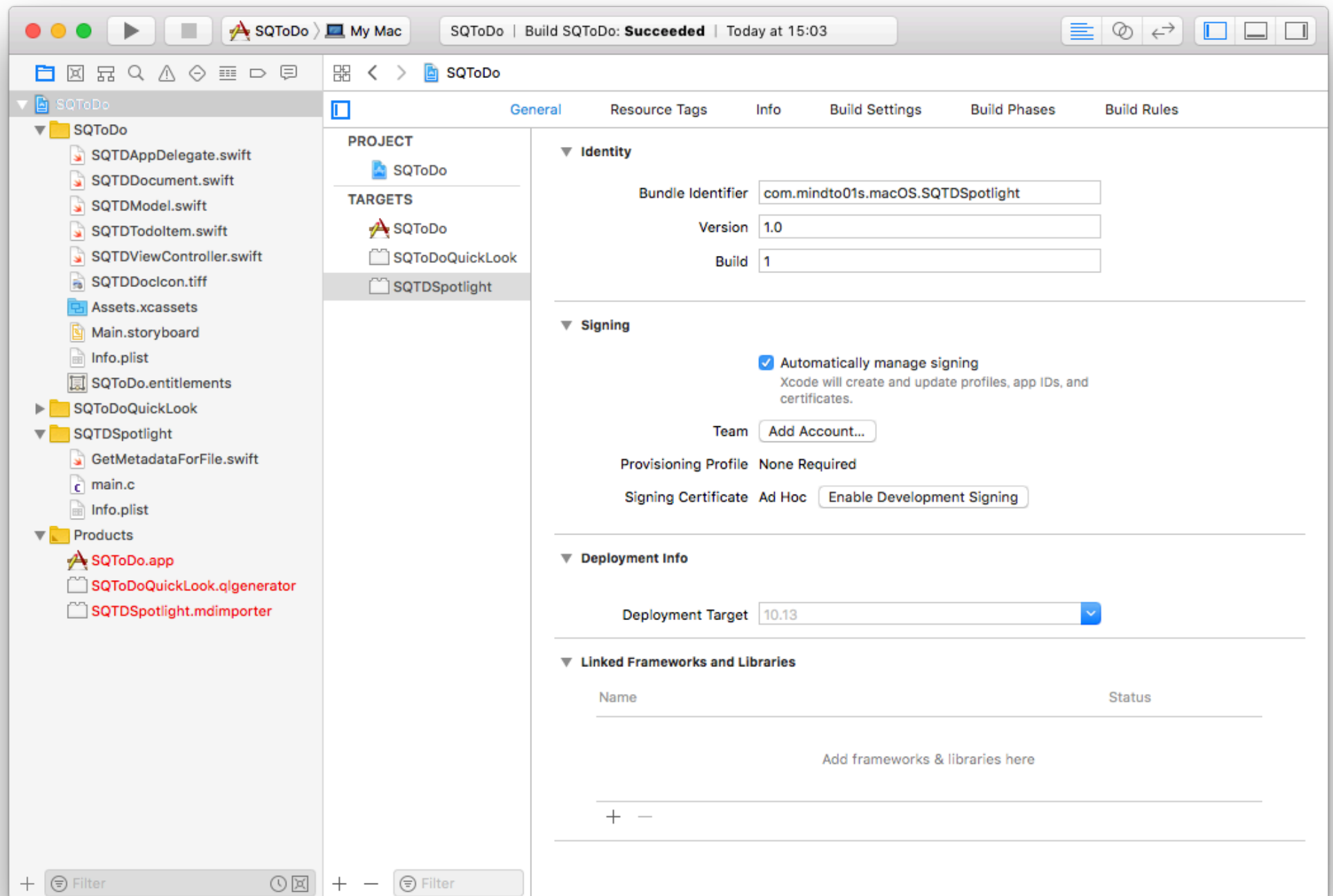
選んで



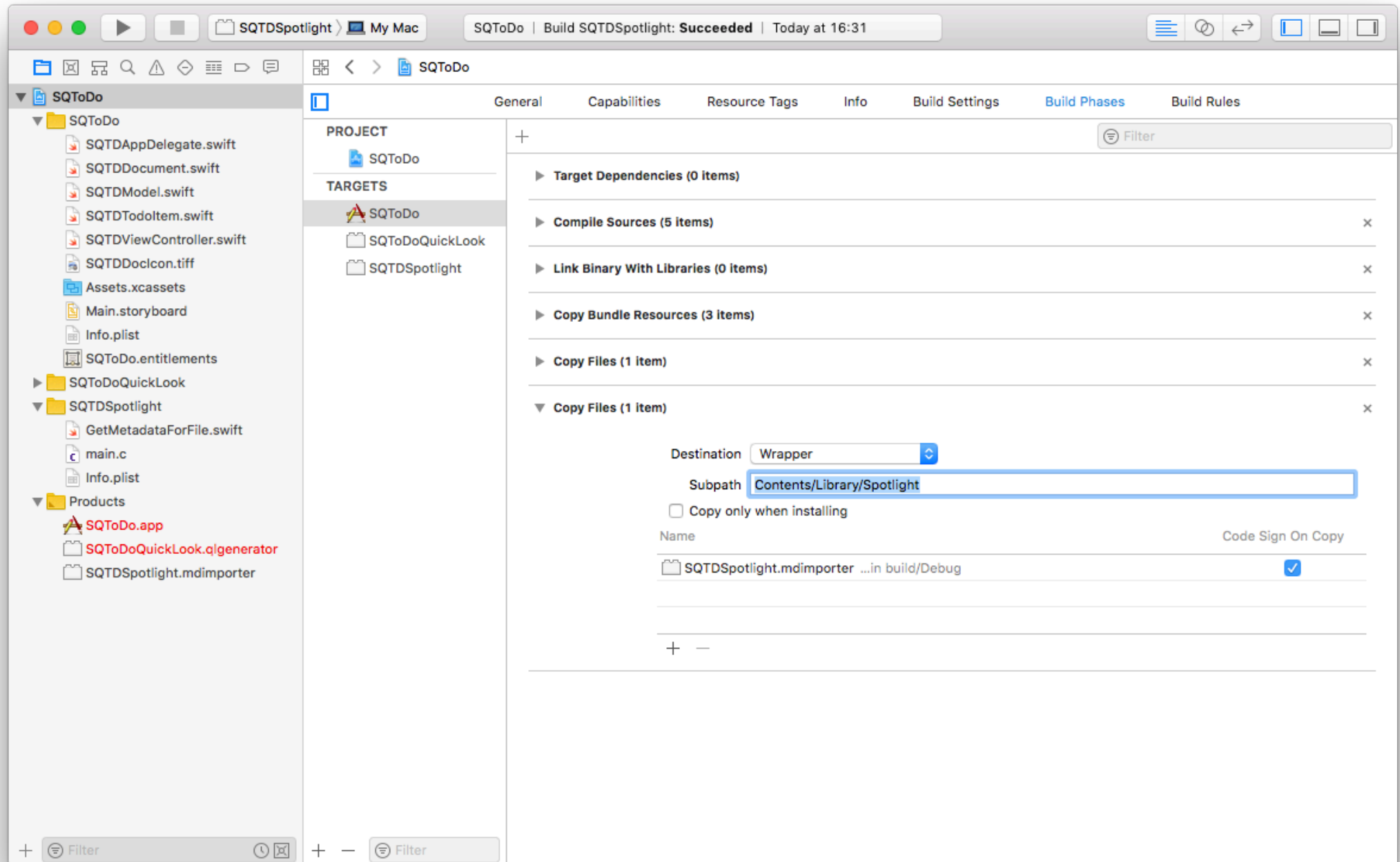
名付けて



こうなる



ToDoアプリに埋め込んで



テンプレートで作成されたファイルの役割

ファイル名	役割
GetMetadataForFile.swift	ファイルからメタデータを作り出す場所。ここにコードを追加する。
main.c	変更するな、と書いているので従うこと。pluginとして動作させるためのコードが入っている。
Info.plist	コード以外の各種設定を、実行時にmainBundleに与える。いつものInfo.plist。

info.plistの設定

```
<key>CFBundleDocumentTypes</key>
  <array>
    <dict>
      <key>CFBundleTypeRole</key>
      <string>QLGenerator</string>
      <key>LSItemContentTypes</key>
      <array>
        <string>com.mindTo01s.macOS.sqtdDoc</string>
      </array>
    </dict>
  </array>
```

アプリケーション本体がある場合は、他の値は基本は無視してOK。

最小限の動作確認

mdimportを使う。

```
mdimport -g プラグインへのパス -d2 メタデータを抜き出すファイルへのパス
```

スイッチgは、デバッグ中のプラグインへのパス

スイッチdは、デバッグレベル。1から4まで指定できる。2より大きくなないと、Attributesが表示されない。

上記のコードで使って見る。”print文”を仕込むと動いているかどうか分かる。適当な場所にprint文を仕込んで実行する。 return文の前にprint(“適当なログ”)を挿入すると、以下のような出力が得られる。

```
$ mdimport -g /path/to/SQTDSpotlight.mdimporter -d2 /path/to/SQTD-Test-Data.sqtd
```

適当なログ

```
2018-05-10 23:02:14.397 mdimport[43202:7551652] Imported '/path/to/SQTD-Test-Data.sqtd' of type 'com.mindto01s.macos.sqtdoc' with plugin /path/to/SQTDSpotlight.mdimporter.
```

```
2018-05-10 23:02:14.398 mdimport[43202:7551652] Attributes: {
```

```
    ":EA:_kMDItemUserTags" = (
    );
    ":EA:kMDItemLastUsedDate" = "2018-05-10 04:25:05 +0000";
    ":MD:DeviceId" = 16777220;
    ":MD:kMDExtendedImportDictionary" = {
        "PLUGIN-ERROR" = "RETURNS-ZERO";
    };
    ":MD:kMDItemPath" = "/path/to/SQTD-Test-Data.sqtd";
    "_kMDItemContentChangeDate" = "2018-05-10 04:25:06 +0000";
    "_kMDItemCreationDate" = "2018-05-10 04:25:05 +0000";
    "_kMDItemCreatorCode" = 0;
    "_kMDItemDisplayNameWithExtensions" = {
        "" = "SQTD-Test-Data.sqtd";
    };
    "_kMDItemFileName" = "SQTD-Test-Data.sqtd";
    "_kMDItemFinderFlags" = 16;
    "_kMDItemFinderLabel" = 0;
    "_kMDItemIsExtensionHidden" = 1;
    "_kMDItemLocked" = 1;
    "_kMDItemOwnerGroupID" = 501;
    "_kMDItemOwnerUserID" = 501;
    "_kMDItemPrescanCandidate" = 1;
    "_kMDItemStaticInterestScore" = 1;
    "_kMDItemTypeCode" = 0;
    "com_apple_metadata_modtime" = "547619106.173092";
    kMDItemAlternateNames = (
        "SQTD-Test-Data.sqtd"
    );
    kMDItemContentCreationDate = "2018-05-10 04:25:05 +0000";
    kMDItemContentModificationDate = "2018-05-10 04:25:06 +0000";
    kMDItemContentType = "com.mindto01s.macos.sqtdoc";
    kMDItemContentTypeTree = (
        "public.item",
        "com.mindto01s.macos.sqtdoc",
        "public.data"
    );
    kMDItemDateAdded = "2018-05-10 04:25:05 +0000";
    kMDItemDisplayName = {
        "" = "SQTD-Test-Data";
    };
    kMDItemKind = {
        "" = "Todo list for Spotlight and QuickLook ";
    };
    kMDItemLogicalSize = 584;
    kMDItemPhysicalSize = 4096;
}
```

ソースコードの解説します。添付のコードを見てね

終わり