Getting Started with EPUB UI Component

AnFengDe 

Introduction

EPub UI Component is a development kit with which EPub 3.0 applications can be developed easily. Referenced in your project, the component helps your parsing and displaying the book.

Installation

The **EPub UI Component** is distributed as a .zip archive. The file has the following naming convention: AnFengde-EPUB_UI_Component-<version>.zip, where

- <version> is the version number, e.g. 20120704
- for example: AnFengde-EPUB_UI_Component-20120704.zip

The distribution

After downloading and extracting the archive to the desired location, you will find the following contents:

- **android** directory containing the EPub SDK resources for the Android platform
  - **EPUB_SDK_Example** directory containing the EPub SDK example for android
  - **UI_Example** directory containing the UI component project for android
  - **lib** directory containing the ***EPUB_UI*** library project
- **ios** directory containing the EPub_SDK resources for the iOS platform
  - **EPUB_SDK_Example** directory containing the EPub SDK example for iOS
  - **UI_Example** directory containing the UI component project for iOS
  - **lib** directory containing the framework ***AnFengDe_EPUB_SDK.framework*** and ***AnFengDe_EPUB_UI.embeddedframework***
- **README.md**
- **GettingStartedwithEPUBSDK.pdf**
- **GettingStartedWithEPUBUIComponent.pdf**

Setting Up Development Environment

For the Android platform, this guide assumes that you are using Eclipse as your Android IDE, and that you have installed and configured the Android SDK and ADT plug-in for Eclipse.
For iOS platform, this guide assumes that you are using Xcode as your IDE.
If you are using other IDEs, please consult the documentation of your IDE.

Using EPUB UI Component in Your Application Project

Android Platform

EPub UI Component is a Library Project in android platform, and getting started with it is very simple. You just add the library to your project. Now look at the sample and it will tell you how to reference the **EPub UI Component** in your android project.

Importing EPUB_UI Project

To import **EPUB_UI** library project in you **Eclipse(Version: Juno Release)**.

- **File**-> **Import**->**Existing Android Code Into Workspace**->Browse our **EPUB_UI** project

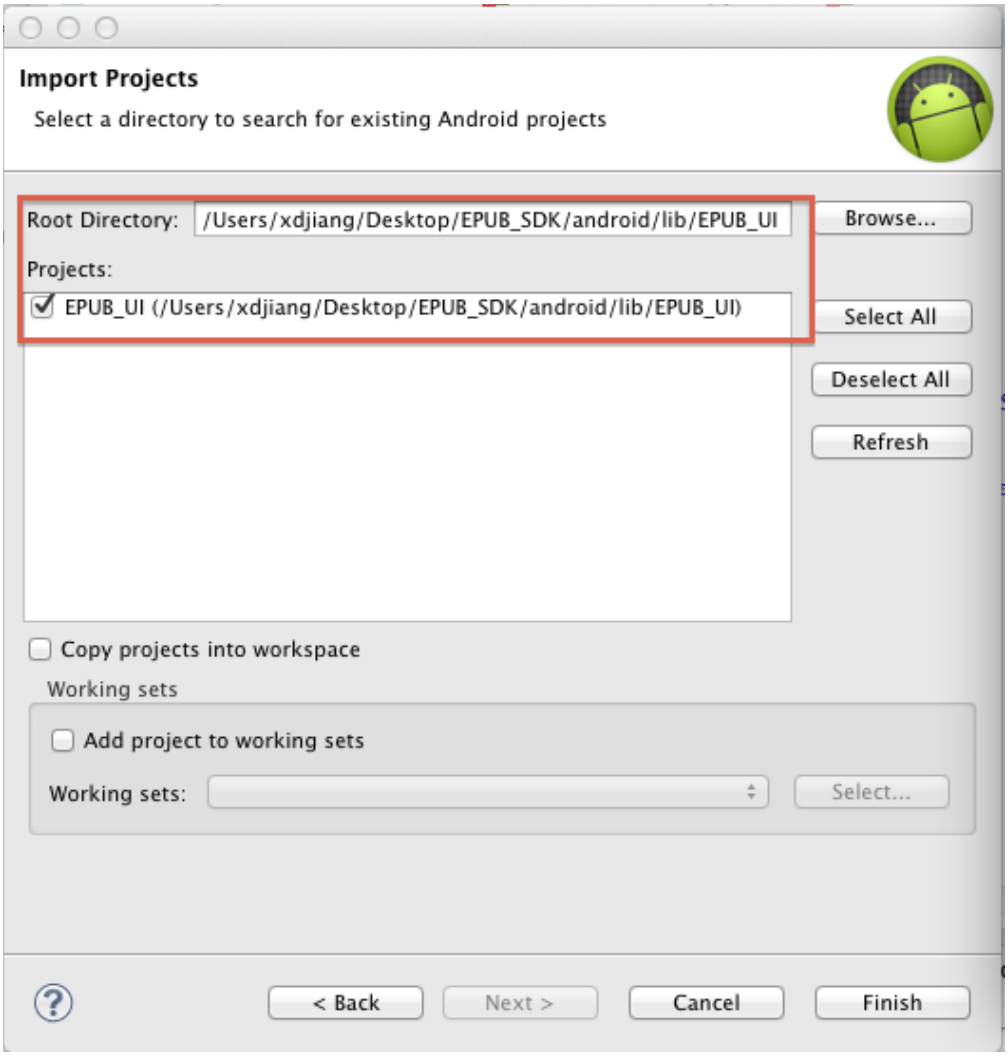


Figure 1-import the EPUB_UI project

Referencing EPUB_UI Project

To create an android project:

- **File->New->Android Project**, name the project as **CallEPubUI**
- Right click on the **CallEPubUI** project->**Properties->Android->Add->EPUB_UI->OK->OK**

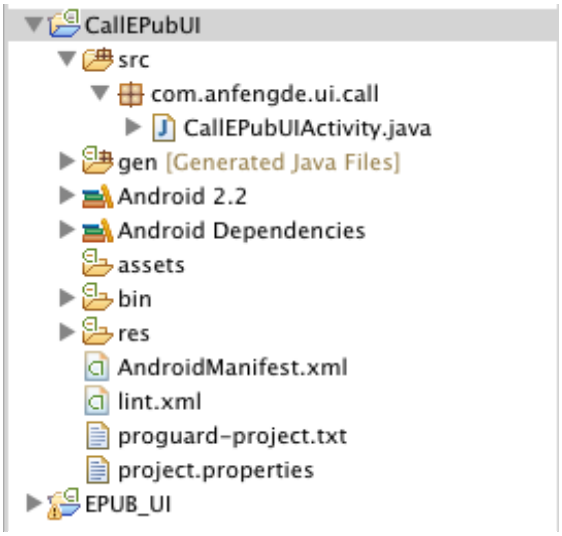


Figure 2-CallEPubUI project

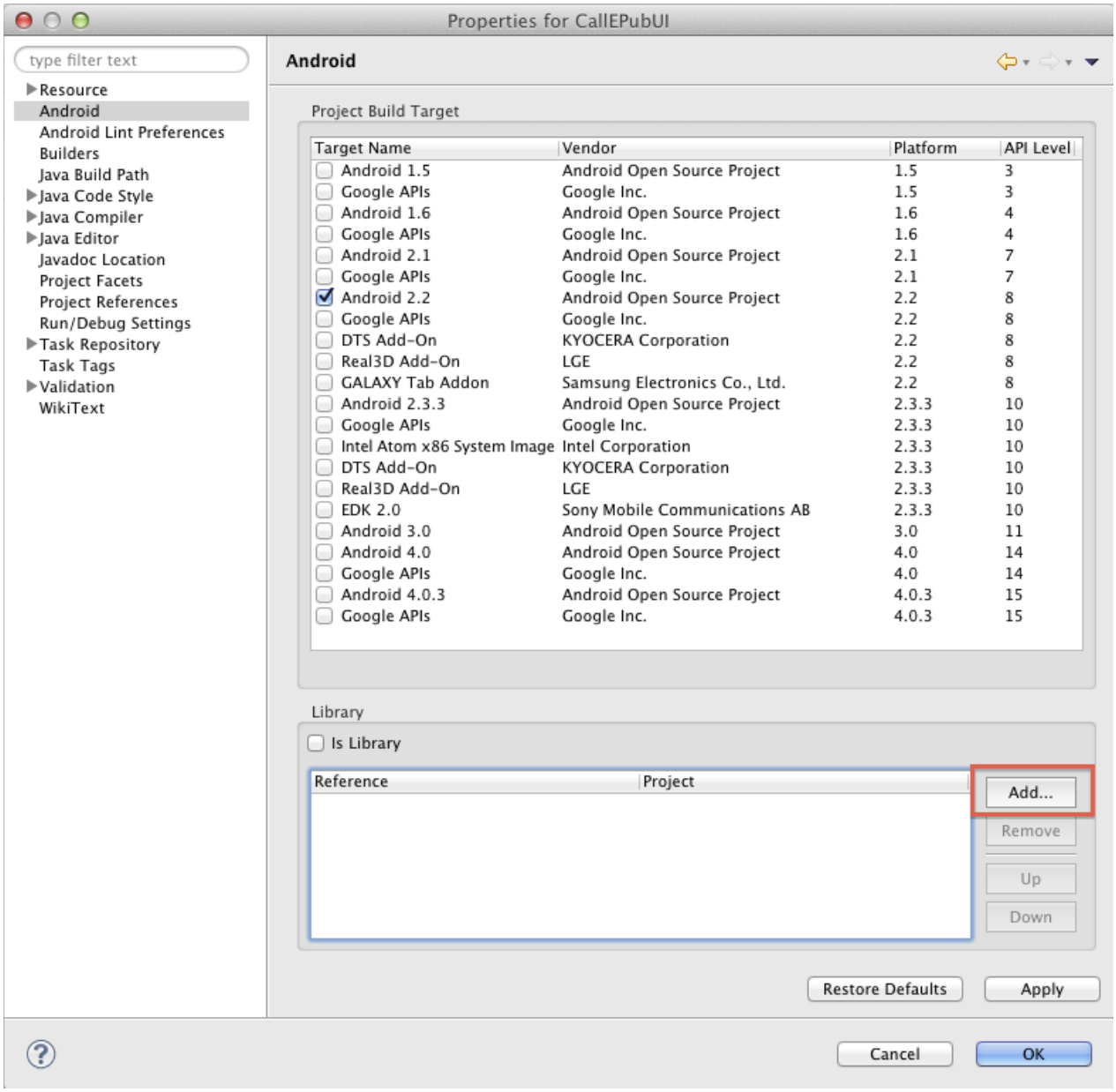


Figure 3-reference the EPUB_UI project(1)

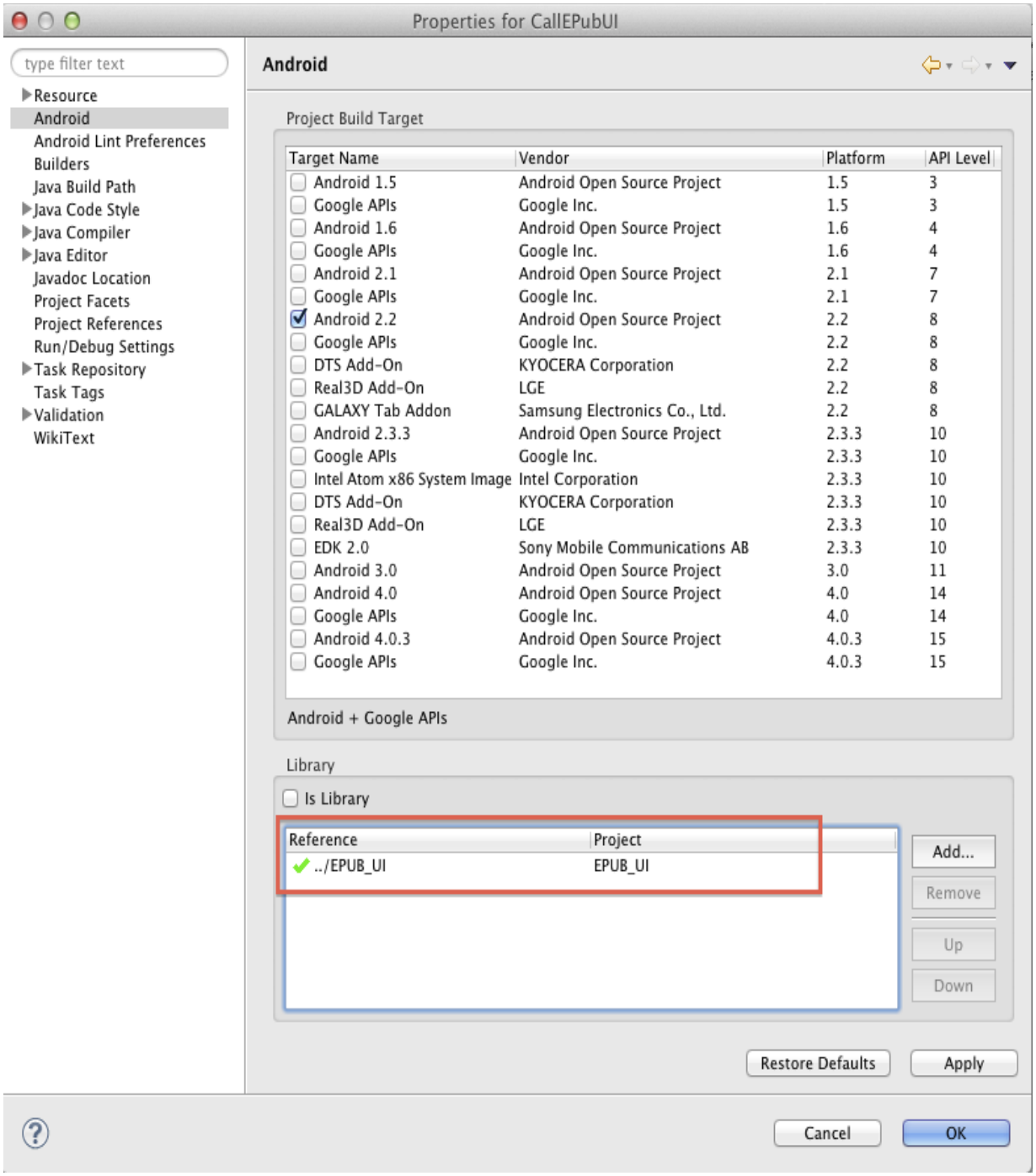


Figure 3-reference the EPUB_UI project(2)

Opening the EPub Book

If everything is OK, now add the code in the main.xml of the **CallEPubUI** project.

```
<com.anfengde.epub.ui.BookView
 android:id="@+id/bookView1"
```

```
 android:layout_width="match_parent"
 android:layout_height="fill_parent" >
</com.anfengde.epub.ui.BookView>
```

Add some codes in the AndroidManifest.xml of the **CallEPubUI** project.
in the **manifest** tag add the code:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_SETTINGS"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"/>
```

in the **application** tag add the code:

```
<activity android:name="com.google.ads.AdActivity"
android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|uiMode|screenSize|
smallestScreenSize"/>
```

Pay attention to **space symbol**. If some problems occur, please set project target in ***project.properties*** to ***android-13*** or above and then clean the project.

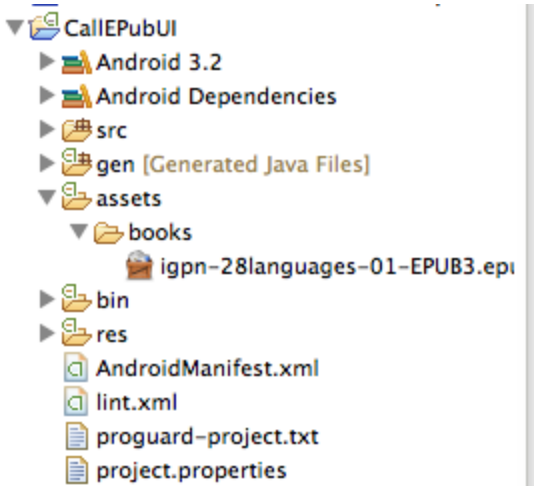
As a last step, add some codes in the **onCreate** method of the CallEPubUIActivity.

```
BookView bookView = (BookView) findViewById(R.id.bookView1);
bookView.setPath(Constants.CACHE_PAHT);
bookView.initBook();
bookView.openShelf();
```

“bookView1” is the EPub UI Component ID, you can change it.
“Constants.CACHE_PAHT” is the cache path.

Notice:
When you want to rebuild your app, you must follow the steps below:
1, Cleanse the /sdcard/epub/ fold in the device;
2, Delete the program that has been compiled in the device;
3, Rebuild.

Now you can add the books in assets/books folder like this:



Run the **CallEPubUI** project.

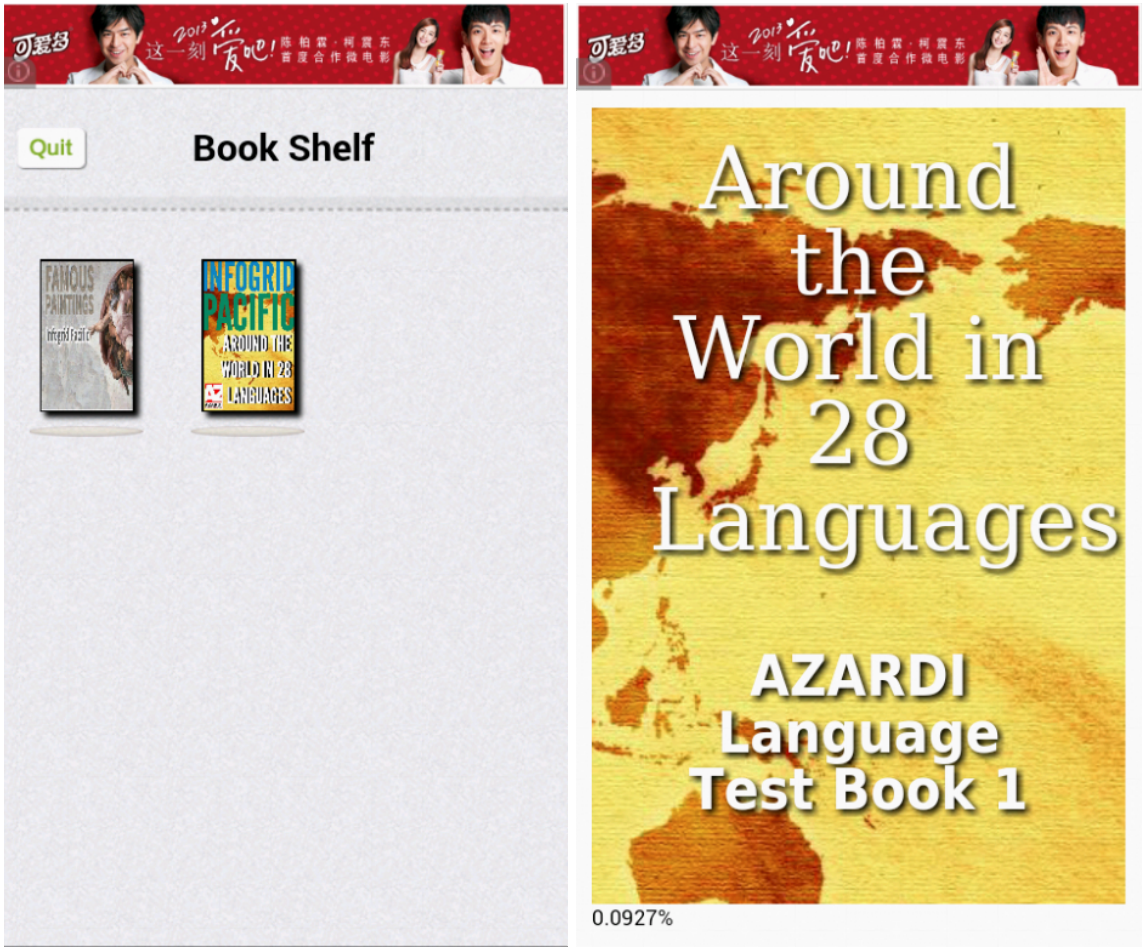


Figure 4-the book showing up

IOS Platform

EPub UI Component is the framework in iOS platform, and getting started with it is very simple. You just add the framework to your project. Now look at the sample and you know how to reference it in your iOS project.

Creating a Project

To create a new project and name it as **CallEPubUI** with **Xcode(Version 4.2.1)**

- **Create a new Xcode project** -> **Application**->**Single View Application**

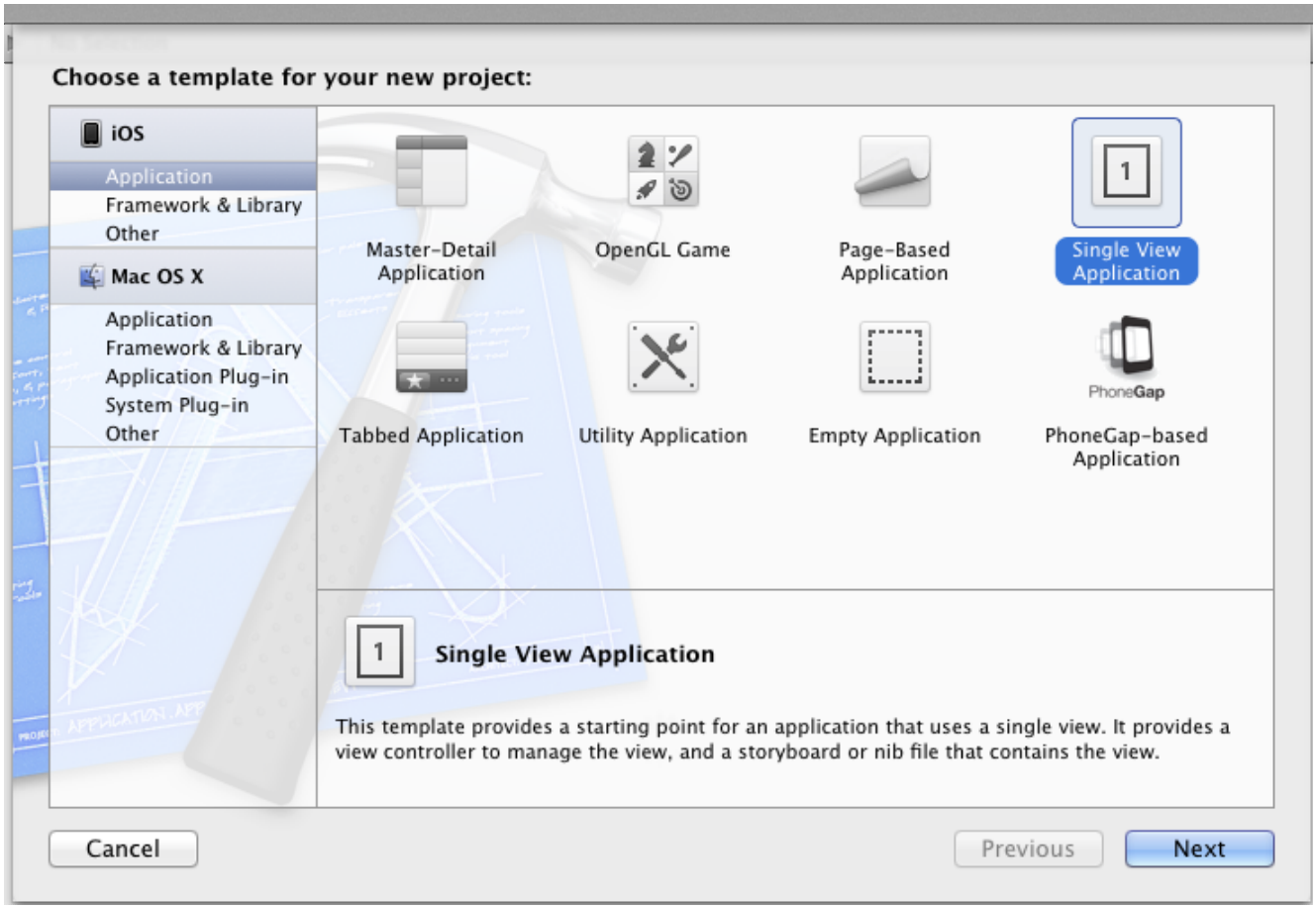


Figure 5-create a single view application

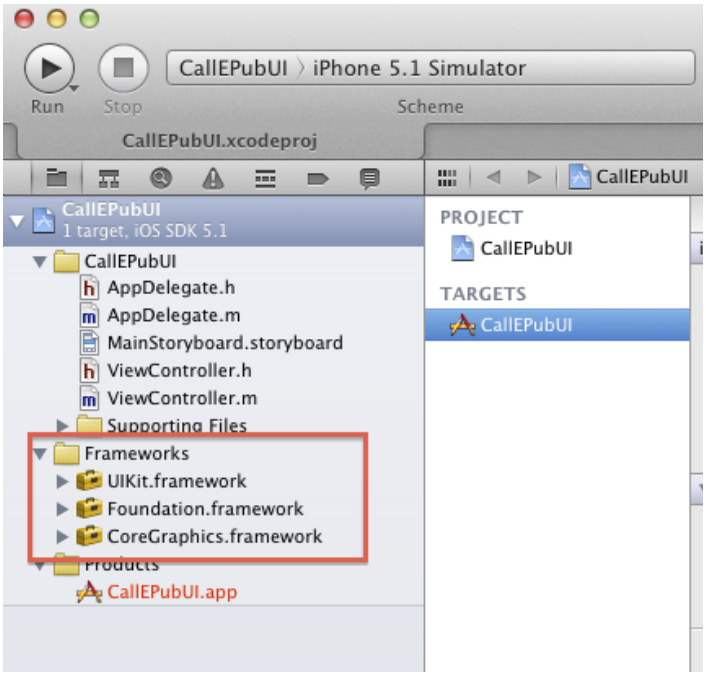


Figure 6-CallEPubUI project

Adding the Frameworks and Resources

To add the frameworks and resources to **CallEPubUI** project in Xcode:

- Right click on the Frameworks in the **CallEPubUI** project
- Select "**Add Files to "CallEPubUI"**"
- Select **AnFengDe_EPUB_SDK.framework** and **AnFengDe_EPUB_UI.embeddedframework,** and then add them to **CallEPubUI** project
- In **Build Phases** option drag **.js files** from **Compile Sources** to **CopyBundle Resources**

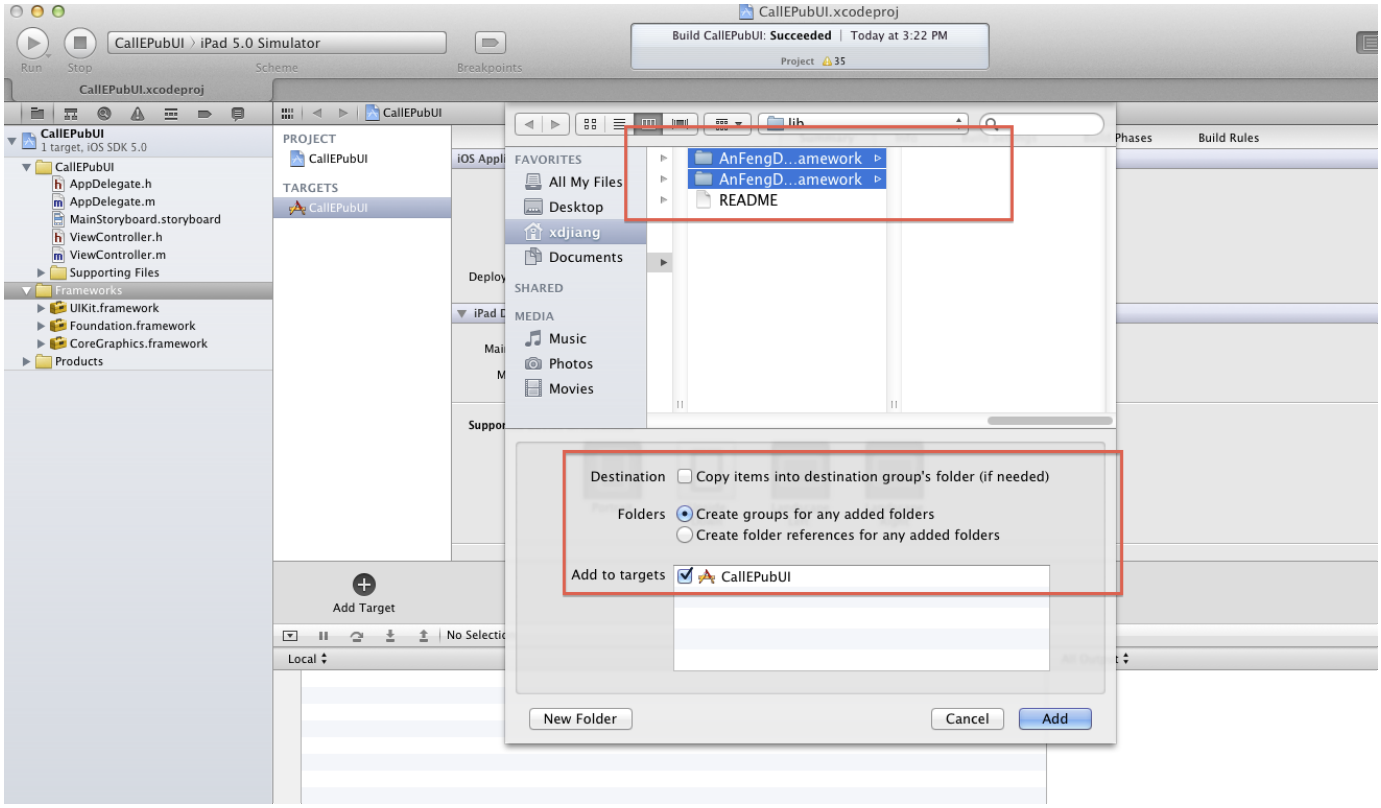


Figure 7-add the frameworks to CallEPubUI project(1)

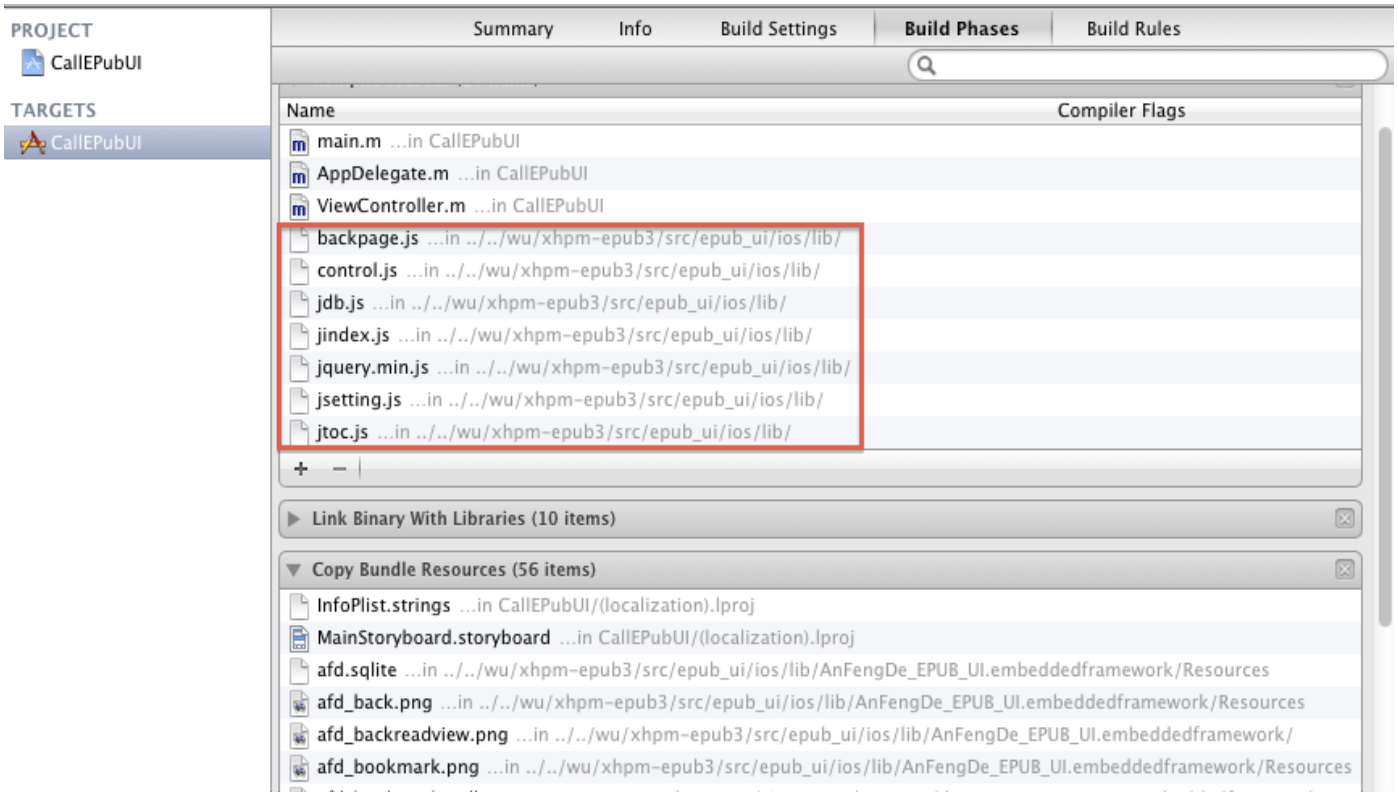


Figure 7-add the frameworks to CallEPubUI project(2)

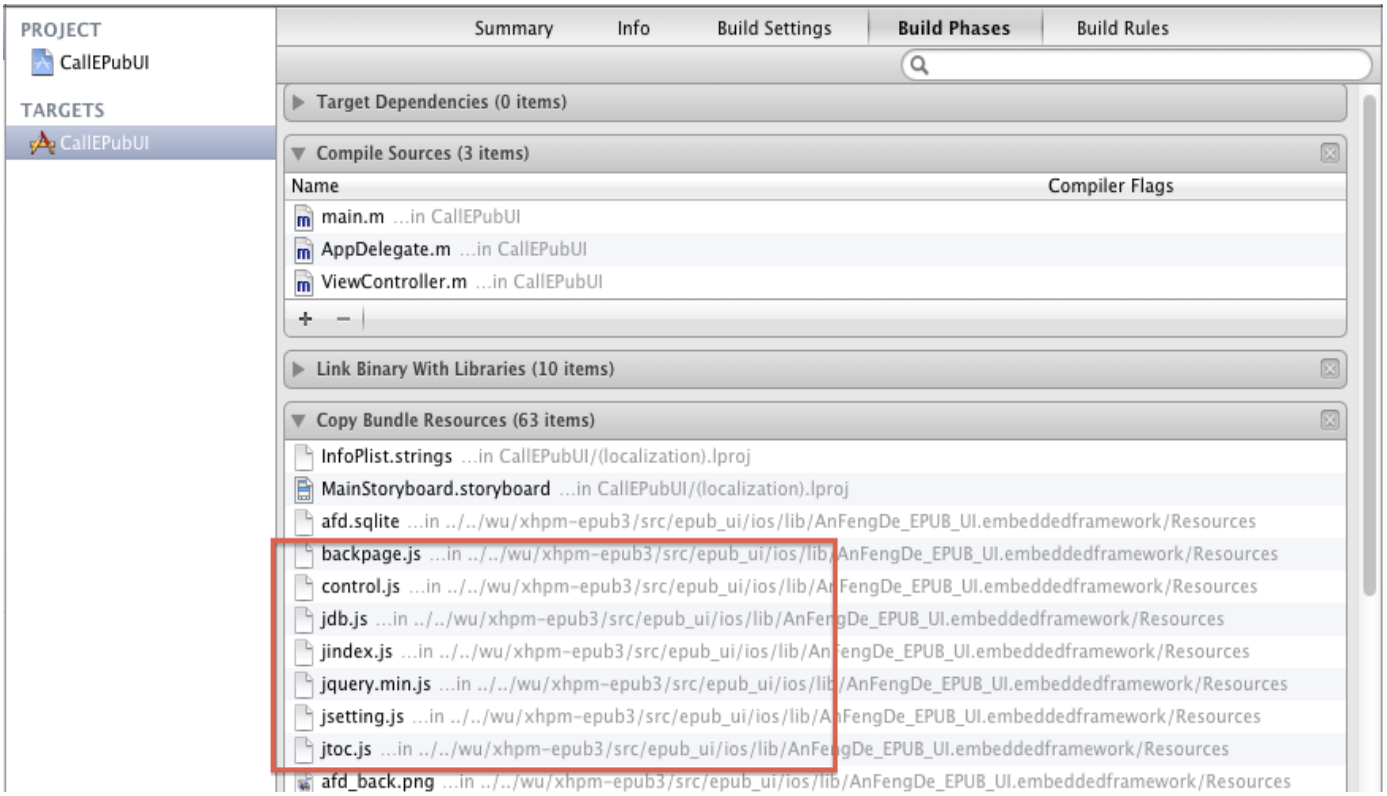


Figure 7-add the frameworks to CallEPubUI project(3)

Add iOS framework

Add the following frameworks to your project:**CallEPubUI** target ->**Build Phases** ->**Link Binary With Libraries** ->"+"

- `MediaPlayer.framework`
- `MessageUI.framework`
- `SystemConfiguration.framework`
- `AudioToolbox.framework`
- `CoreGraphics.framework`
- `libsqlite3.dylib`

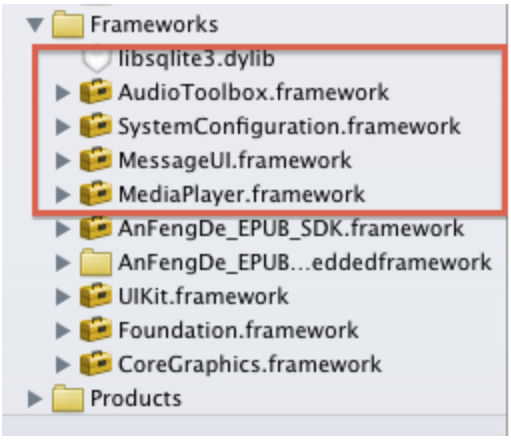


Figure 8-add iOS framework

Add `-all_load` under Other Linker Flags in the project build info: **CallEPubUI** target ->**Build Settings** ->**Linking** ->**Other Linker Flags**->Add "`-all_load`".

Opening the EPUB Book

If everything is OK, the **CallEPubUI** project looks like this:

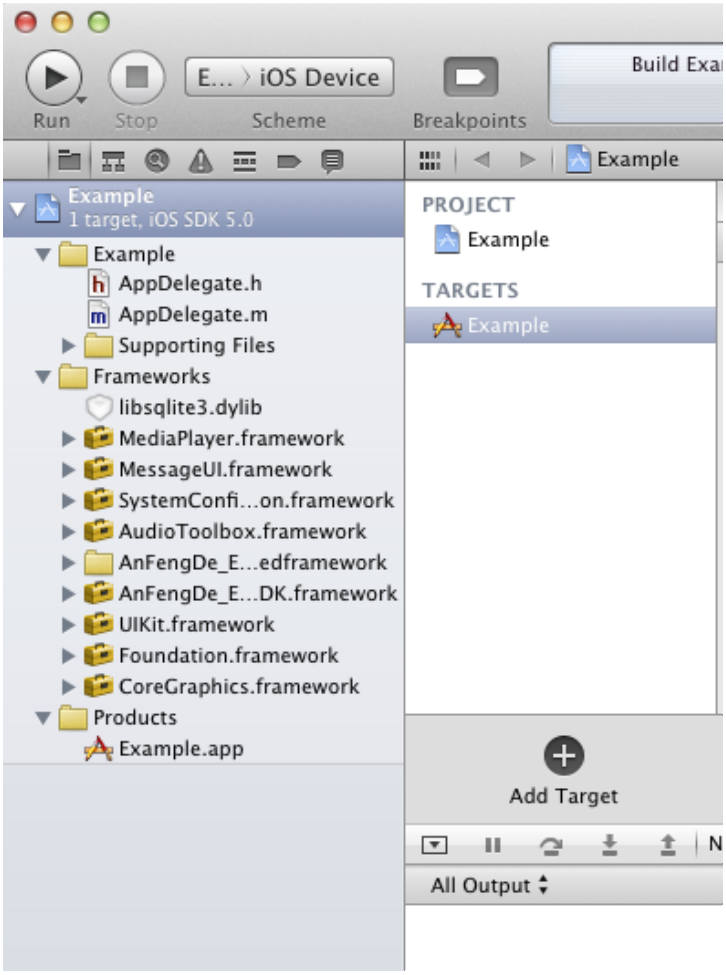


Figure 8-the resources list of CallEPubUI project

Now add some codes in the AppDelegate.h (not ViewController.h).

```
#import <UIKit/UIKit.h>
#import <AnFengDe_EPUB_UI/EPubUIHeader.h>

@interface AppDelegate : UIResponder <UIApplicationDelegate>
@property (strong, nonatomic) UIWindow *window;
@property (strong, nonatomic) EPubRootViewController *rootEpubView;
@end
```

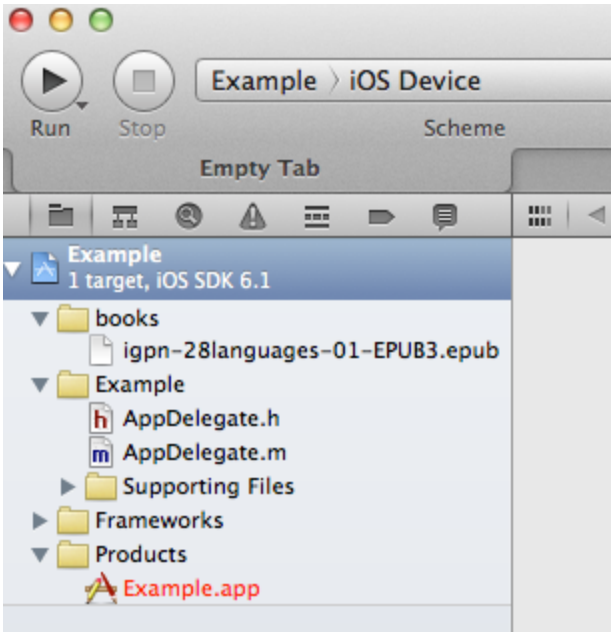
And in the AppDelegate.m (not ViewController.m), some codes are added to the method:

```
@implementation AppDelegate

@synthesize window = _window;
@synthesize rootEpubView;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary
*)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    // Override point for customization after application launch.
    self.rootEpubView = [[EPubRootViewController alloc]
initWithNibName:@"EPubRootViewController" bundle:nil];
    self.window.rootViewController = self.rootEpubView;
    [self.window makeKeyAndVisible];
    return YES;
}
```

Now you can add the books in books folder like this:



The red part is added to show the book.
Run the **CallEPubUI** project.