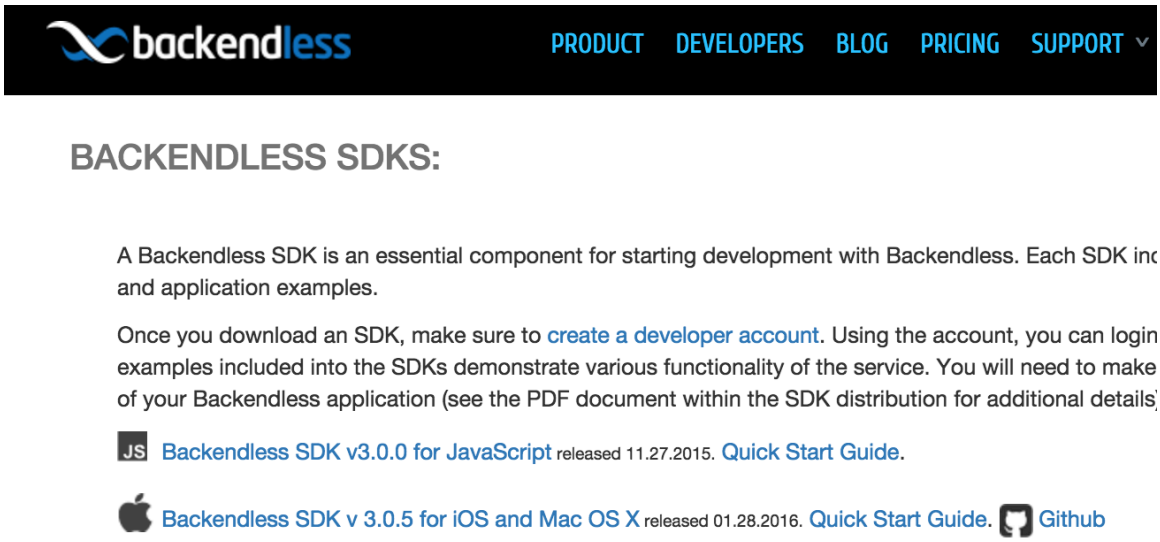
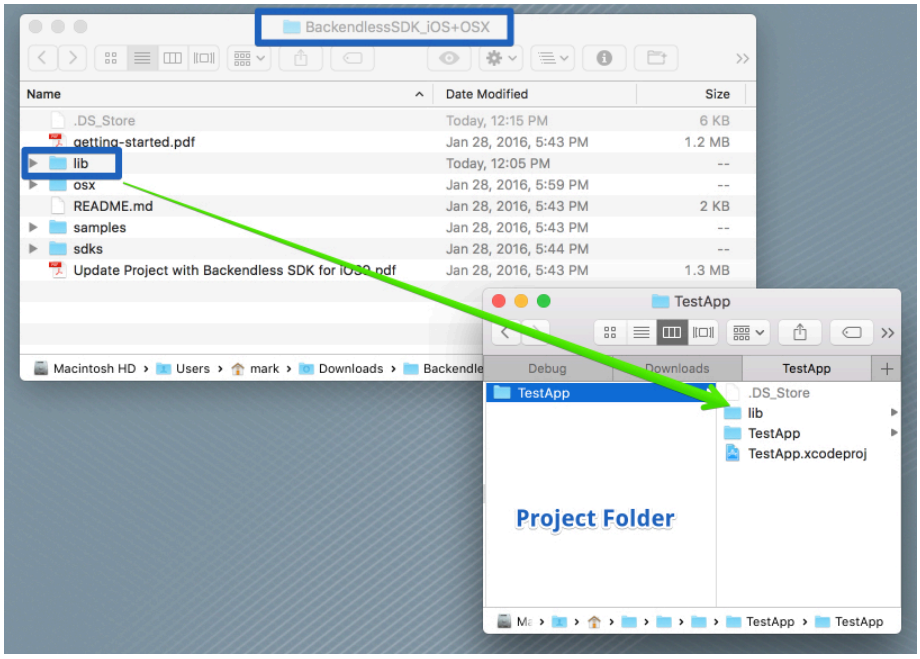
Getting Backendless SDK

1. Login to your Backendless account or register to create a new one.
2. Download Backendless SDK for iOS/OSX from the Backendless SDK Downloads page and unzip it into a directory on your computer.



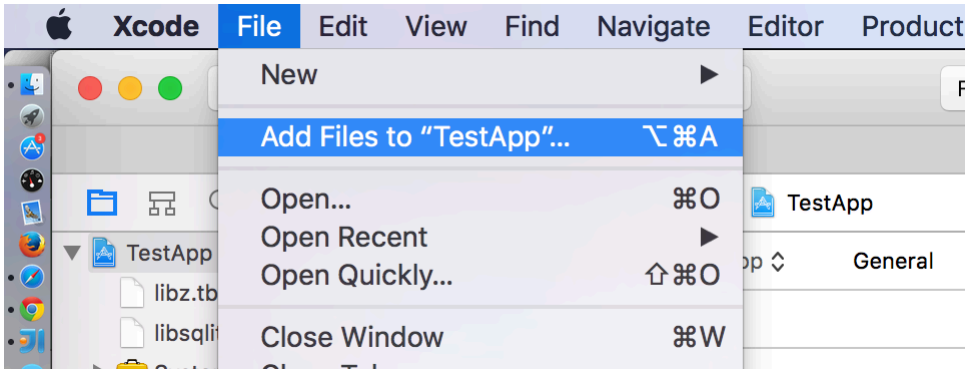
Getting Started

1. Start up Xcode and go to **File** > **New** > **Project**.
2. Select **iOS** > **Application** > **Single View Application**, and click **Next**.
3. Enter a name for the **Product Name**, set the Language to Objective-C or Swift, and click **Next**.
4. Choose a directory to save your project, and click Create.
5. Return to Finder and locate the directory where you expanded Backendless SDK zip. Copy the **lib** folder from the SDK distribution into the root folder of your project.

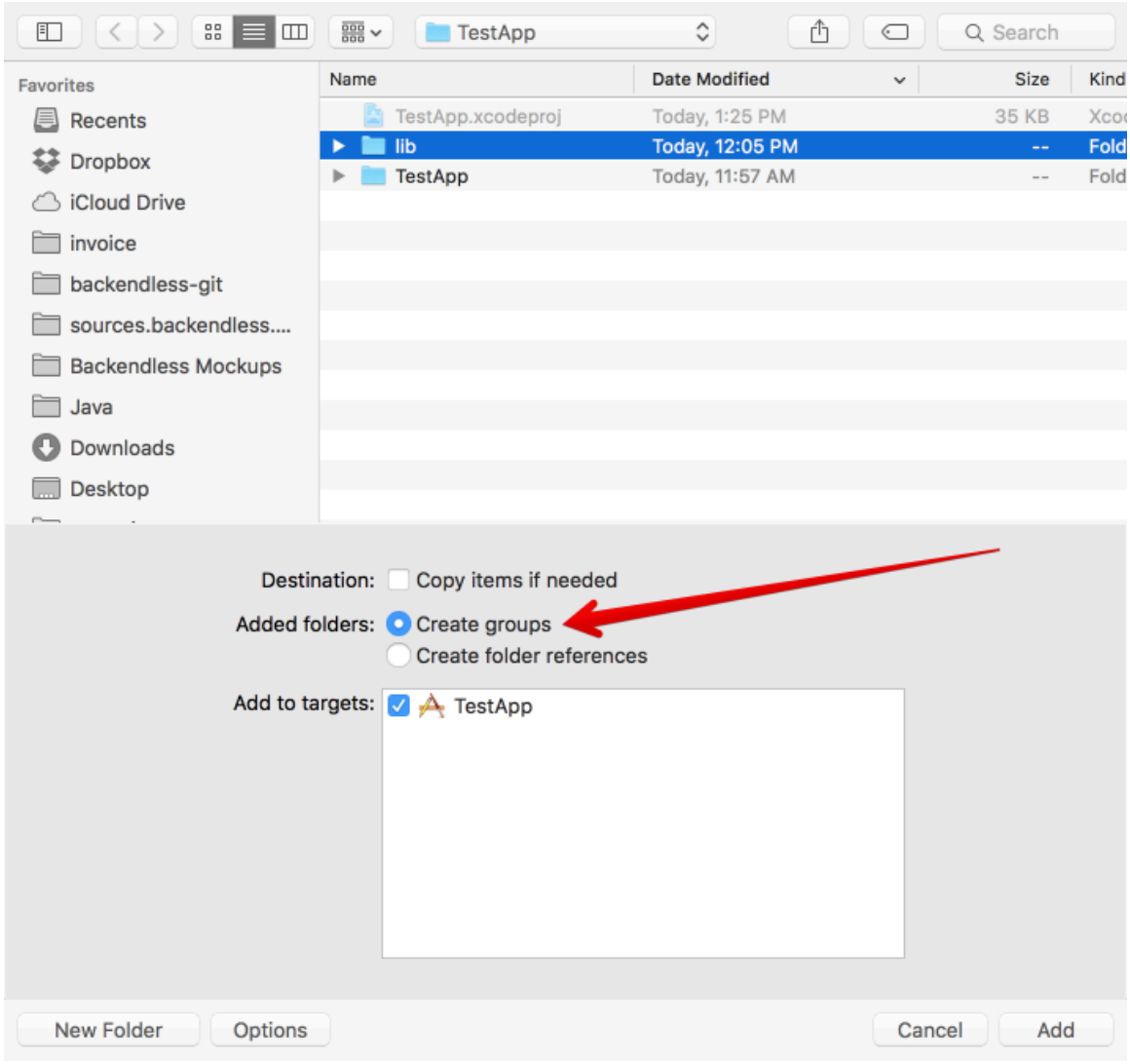


Add the Libraries and Frameworks to the project

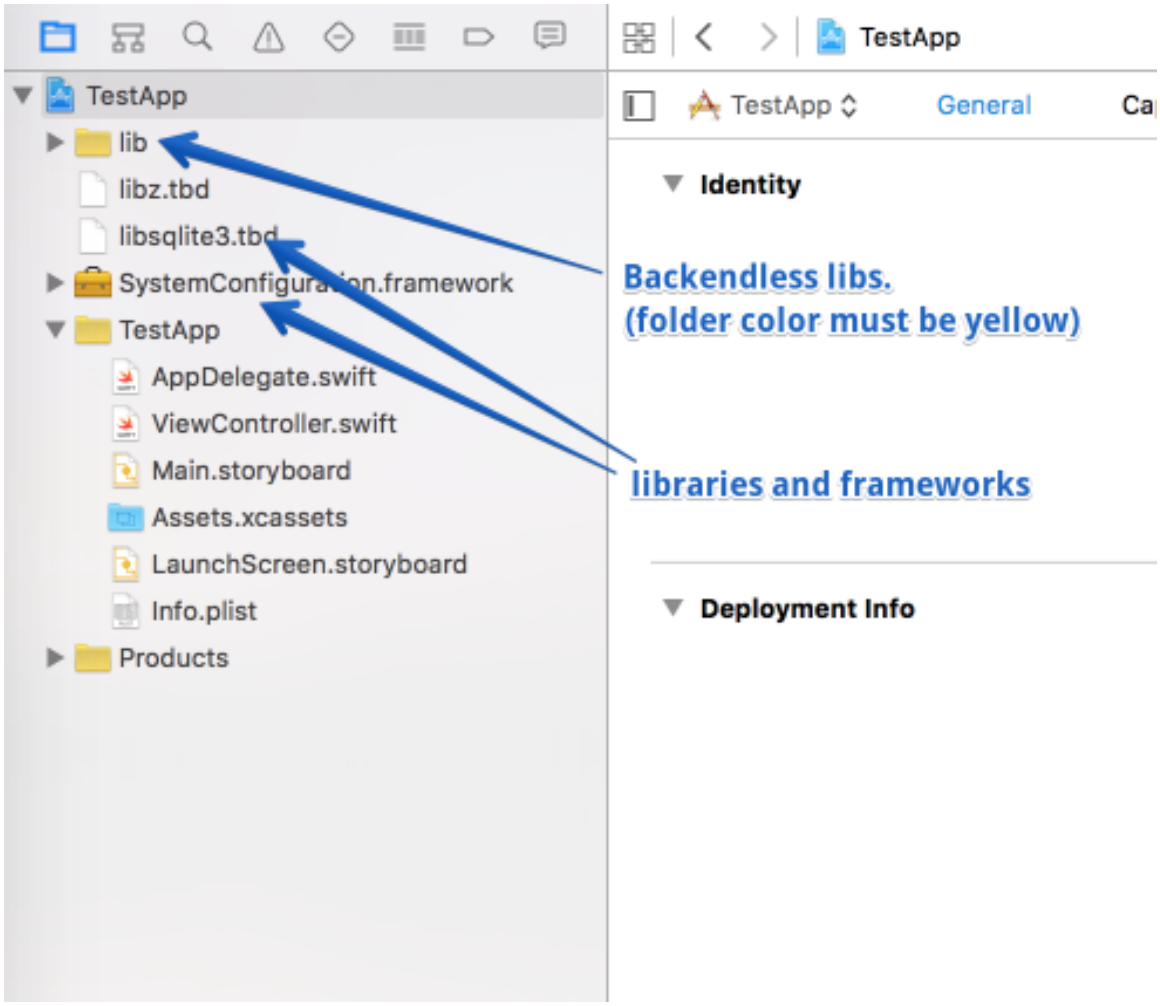
1. Return to XCode and select Project Structure. Select the root node and navigate to **Build Phases** > **Link Binary With Libraries**. Click the + icon and add the following framework and libraries:
   - `SystemConfiguration.framework`
   - `libsqlite3.tbd`
   - `libz.tbd`
2. Select the root node in Project Structure and from the main menu choose **File** > **Add Files to …**:



3. Locate the **lib** folder in the project folder. This is the lib directory you copied from the Backendless distribution. Click the Option button at the bottom of the window and make sure the **Create groups** radio button is selected. Also, make sure that your target in the **Add to targets** list is checked:



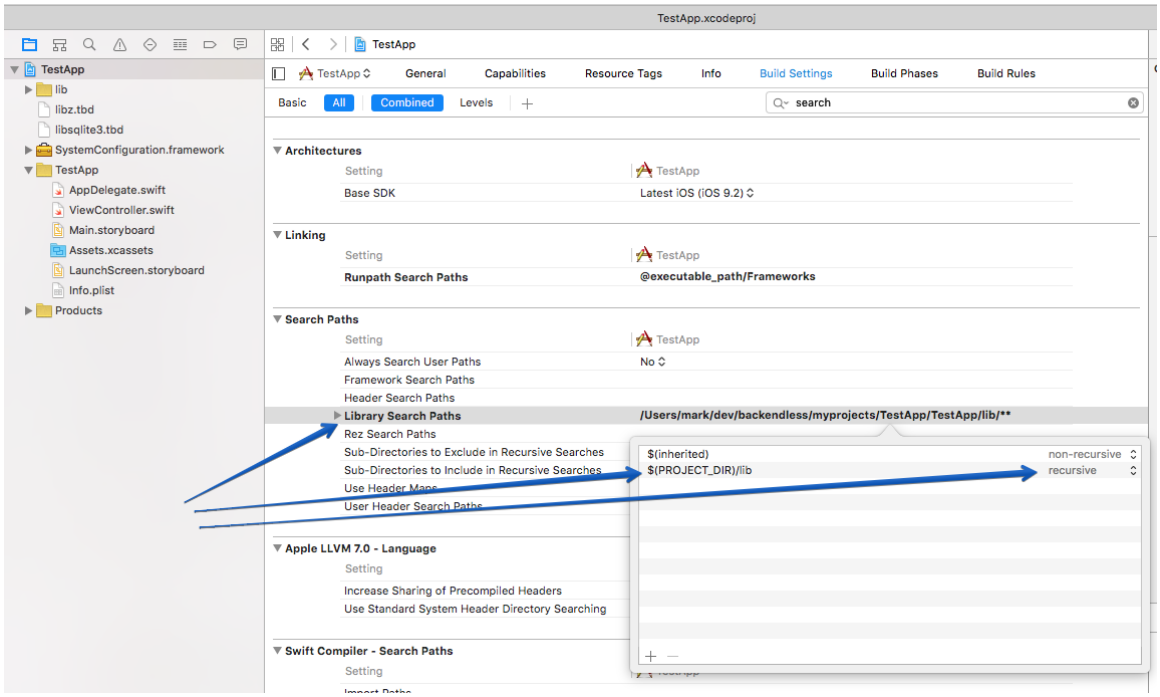
4. Your project structure at this point should look as shown below:



5. Select the root node in the Project Structure. Navigate to **Build Settings** > **Search Paths** -> **Library Search Paths**. Add the following paths:

   ```
   $(inherited)
   $(PROJECT_DIR)/lib
   ```

   Make sure the latter is marked as **recursive**.



6. If the project uses Objective-C, add the following import statement to the application's delegate `AppDelegate.m` file:

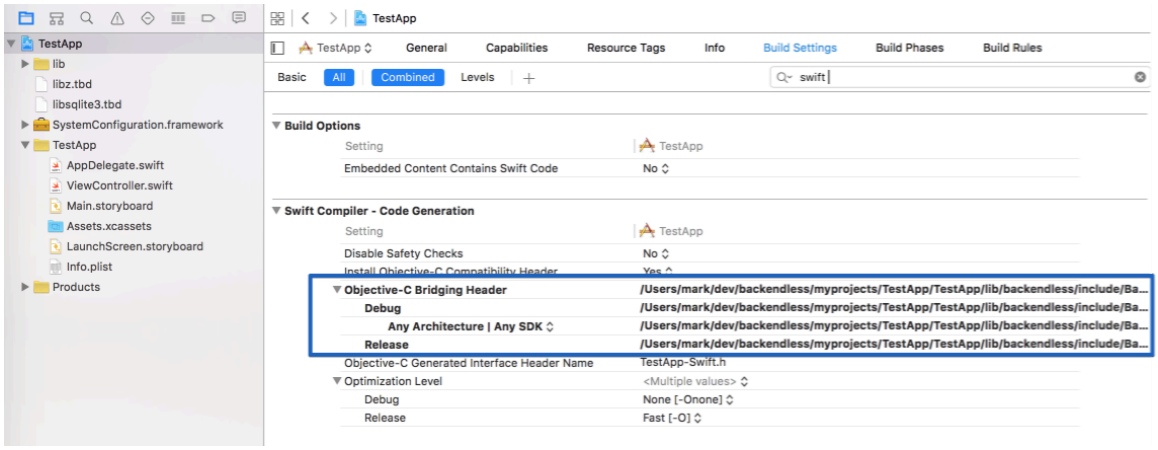
   ```
   #import "Backendless.h"
   ```

   If you are going to use Backendless MediaService, also add:

   ```
   #import "MediaService.h"
   ```

7. If the project uses Swift, add the following value into the **Objective-C Bridging Header** field in the project's Build Settings:

   ```
   $(PROJECT_DIR)/lib/backendless/include/Backendless-Bridging-Header.h
   ```



If you are going to use Backendless MediaService, use the following value for **Objective-C Bridging Header** field:
`$(PROJECT_DIR)/lib/backendless/include/Backendless-With-Media-Bridging-Header.h`

Project Configuration using CocoaPods

Another way to add frameworks and libraries to your Backendless application is by using the CocoaPods service. CocoaPods manages library dependencies for your Xcode projects.

The dependencies for your projects are specified in a single text file called Podfile. CocoaPods resolves dependencies between libraries, fetch the resulting source code, then link it together in an Xcode workspace to build your project.

Make sure you have the Cocoapods ruby gem installed your system. If you don't please follow the directions at CocoaPods Getting Started , or just fire up a Terminal window and run:

```
$ sudo gem install cocoapods .
```

Creating a new Xcode project with CocoaPods

To create a new project with CocoaPods, follow these simple steps:

1. Create a new project in Xcode as you would normally, then close this project.
2. Open a Terminal window, and $ cd into your project directory.
3. Create a Podfile. This can be done by running $ touch Podfile.
4. Open your Podfile using your favorite text editor (or Xcode), and add a text

that looks like this:

```
platform :ios, '8.0'
pod 'Backendless-ios-SDK', '~>3.0.0'
```

The first line specifies the platform and its minimum supported version, the second line specifies the pod name and the minimum its version which you need.

5. Save Podfile, return to Terminal window and run:

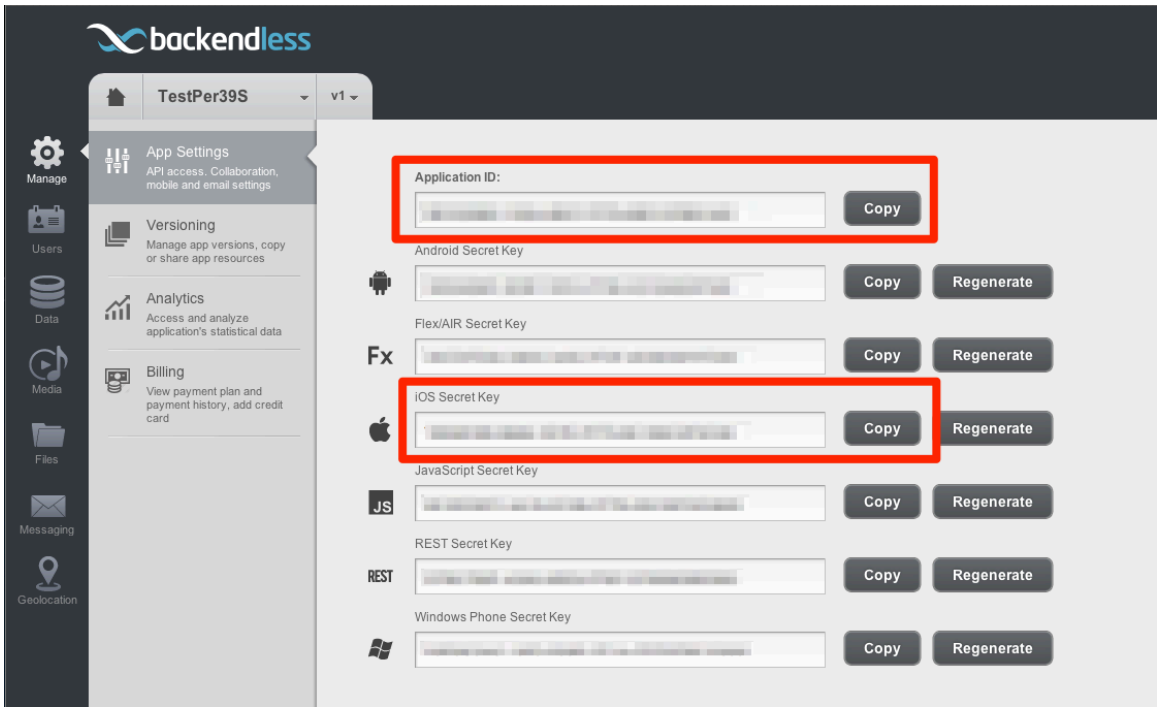
```
$ pod install
```

Once all of the pod data is downloaded, Xcode project workspace file will be created. This should be the file you use everyday to create your app.

6. Open *.xcworkspace* file to launch your project, and build it using scheme for iOS device.

Add Backendless Application Id & Secret Key

1. Get your Backendless application and secret keys for iOS from the Backendless Console. The keys can be found on the Manage -> App Settings section:



2. Add Backendless application initialization code block in `AppDelegate.m`:

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {

    [backendless initApp:APP_ID secret:SECRET_KEY version:VERSION_NUM];
    // if you don't need the MediaService - you must remove the next line
    backendless.mediaService = [MediaService new];

    return YES;
}
```

or in AppDelegate.swift:

```
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

       let APP_ID = "-YOUR-APPLICATION-ID-"
       let SECRET_KEY = "-YOUR-APPLICATION-IOS-SECRET-KEY-"
       let VERSION_NUM = "v1"

       var backendless = Backendless.sharedInstance()

       var window: UIWindow?

func application(application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [NSObject: AnyObject]?) -> Bool {

       backendless.initApp(APP_ID, secret:SECRET_KEY, version:VERSION_NUM)
       // if you don't need the MediaService - you must remove the next line
       backendless.mediaService = MediaService()

       return true
       }
. . .
}
```