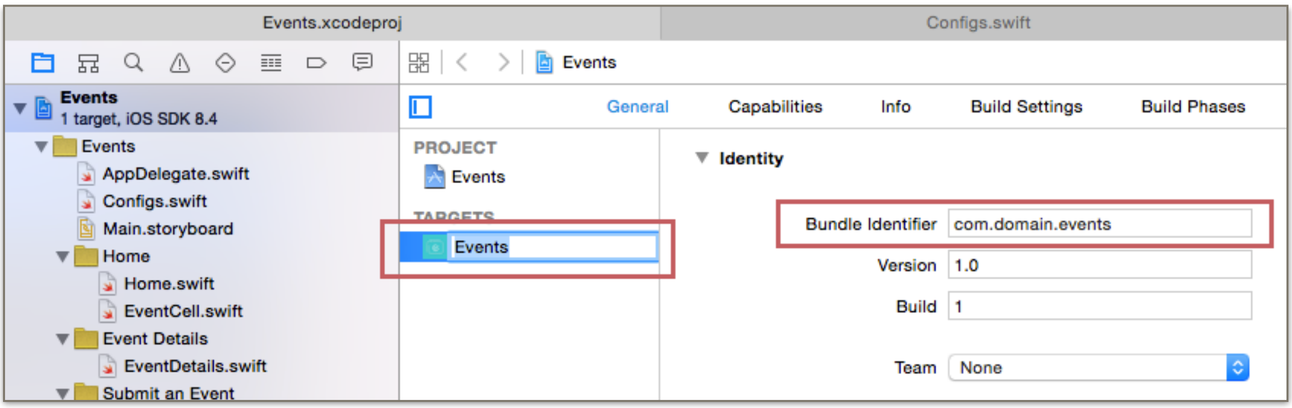
Events

- User Guide -

Thanks for purchasing **Events**, Full Swift iOS Universal Event Sharing App Template from CodeCanyon, we really appreciate your support and we're sure you'll have so much fun with this template!

- First Setup -

- **Bundle Identifier & App name**: If you want to use the demo project included into the package, you can do so. Once you have generated a new **App ID** and **Distribution Provisioning** file from the **Apple Developer portal** (in order for you to be able to upload the binary of your app), you must change the **Bundle Identifier** accordingly to the one you generated in your provisioning profile. Also, you must change the App's name under **TARGETS** by double clicking on **Events** and renaming it as you want, so you'll see your own app's name underneath the app's icon (see below):



Bundle Identifier: com.domain.events

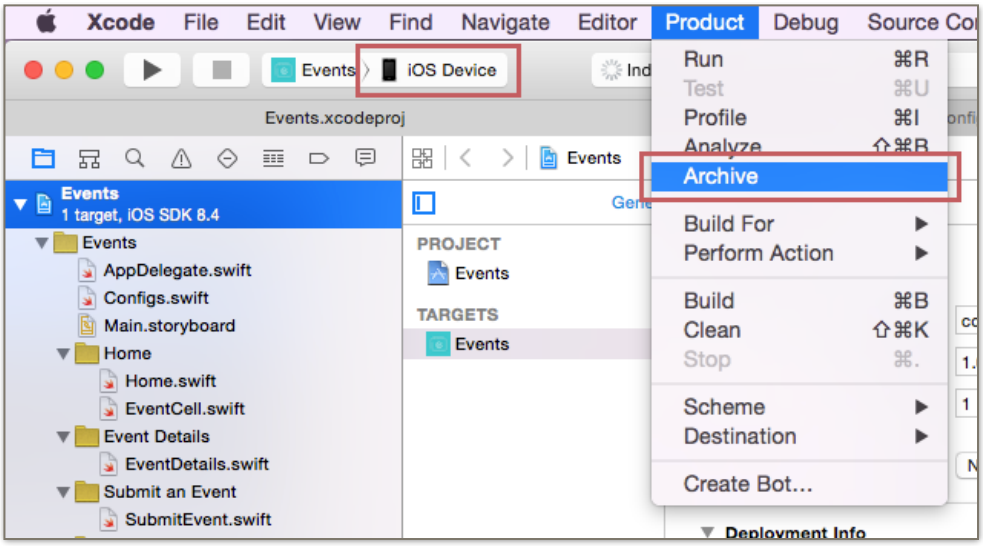
Version: 1.0

Build: 1

Team: None

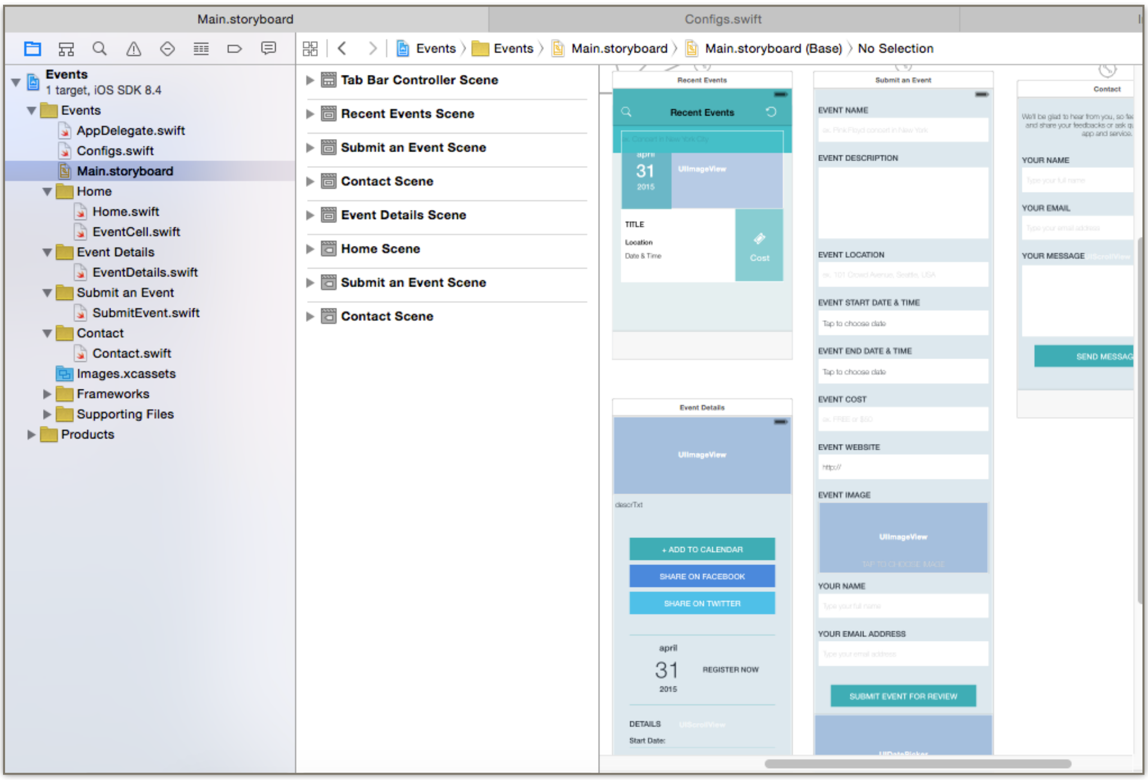
NOTE: *do not change the* **Events** *folder's name or Project's name, or else XCode won't recognize the app and you won't be able to test it on the iOS Simulator nor Archive it for AppStore publishing.*

- **Archive the App:** To archive the app and be able to upload it to the **iTunes AppStore**, select **Generic iOS Device** on the device list and then click on **Product -> Archive** to launch the **Organizer** and archive the app (see below):



- Customization -

- **User Interface:** If you want to edit the User Interface of **Events**, you can do so by the Storyboards file called **Main.storyboard.** You can edit buttons, labels and image views via Inspector panel on the right side of the XCode window, and select them by the the Document Outline panel (see below)



- **Images:** You can easily change App icons, logo, buttons and other images by accessing **Images.xcassets** folder and editing the *.png* images in there, just do not rename them so XCode will still recognize them. Make your reskin and save .png files with the same names as the existing ones in the project.

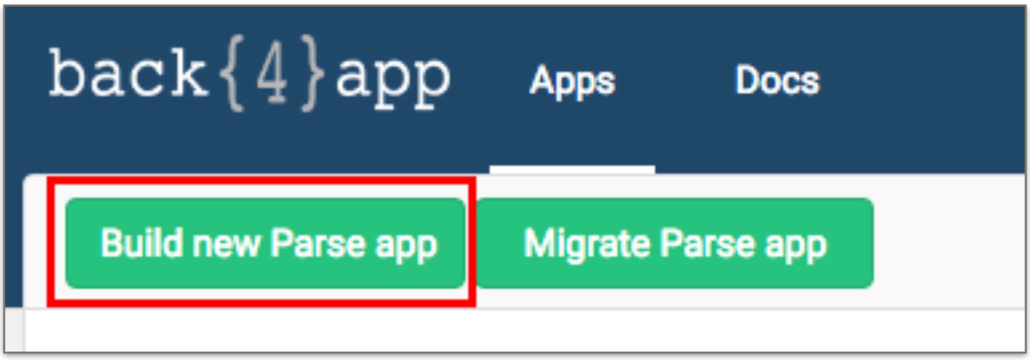
- Configuration -

Head over **Configs.swift** file and check it out. It contains most of the keys/strings you have to set in order to customize your app and get connected to your own Parse App's database. Check out the comments in the swift file which will help you editing them.

- Parse database setup on back{4}app -

Before using the app, you must create your own App on **https://dashboard.back4app.com/apps** and a Bundle Identifier as mentioned in the beginning of this guide. The link we've provided above points already to the back4app Dashboard, but in case you don't have an account on back4app.com yet you'll be redirected to the sign up page.

Once you'll be signed in in back4app, simply click on the **Build new Parse app** button, give your app a name and a brief description (optional), click **Next** and you'll be redirected to the setting page where you'll be able to copy the App ID and Client Key to replace the existing ones into **Configs.swift**



That's it, you can start using your app, back4app provides also the classic Parse dashboard where you can edit records manually.

TIP: We suggest you to insert at least a first even using the app -> Submit screen, so after that the app will create all necessary columns and Class in your Parse Dashboard. Then you'll be able to add rows in there and manually add events if you don't want to use the app all the times.

- **Approve Pending Events:** Once a user submits an event, it gets stored into your own Pasre Dashboard on back4app, and you also get an email with all the details of such event (set a specific email address into **Configs.swift** file, as explained by comments)
All you'll have to do to approve that event is to login into your Parse Dashboard and change the ***isPending*** value relative to that event into **False**. In that way the Event will appear on the home screen and also can be found by search (see below):

If you will not want to approve it for some reason (it may contain inappropriate contents or whatever your policy will be about), you can reply to the user via Mail app. You may still reply to the user who submitted the event to just notify him that his Event has been published on the app. These are just suggestions, unleash your creativity and reskin this app as you wish.

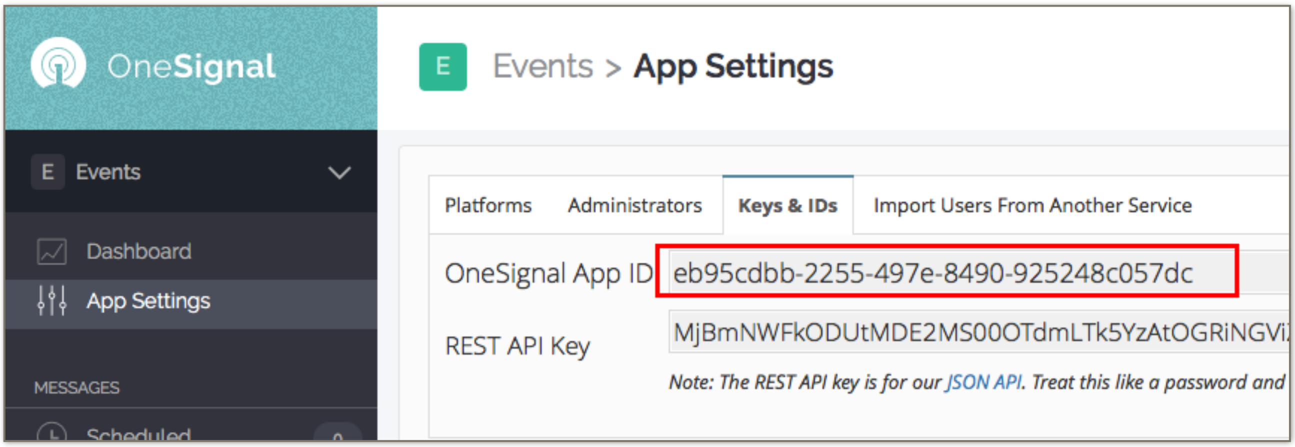
- Push Notifications setup -

1. We're using **OneSignal** push service to send push notifications to all registered users, so first of all create a new account on http://onesignal.com

2. Follow this tutorial in order to create your own APN for Development and Production: https://documentation.onesignal.com/docs/generating-an-ios-push-certificate

3. Enter your OneSignal dashboard and click the Add a new app green button. Follow the Steps provided by the website, unto test Settings, you can ignore it and close the popup window.

4. Click on **App Settings** and copy the OneSignal App ID:



OneSignal App ID eb95cdbb-2255-497e-8490-925248c057dc

REST API Key MjBmNWFkODUtMDE2MS00OTdmLTk5YzAtOGRiNGVi

Note: The REST API key is for our JSON API. Treat this like a password and

5. Then go into Configs.swift file in XCode and replace the red string below with your copied one:

```
let ONESIGNAL_APP_ID = "eb95cdbb-2255-497e-8490-925248c057dc"
```

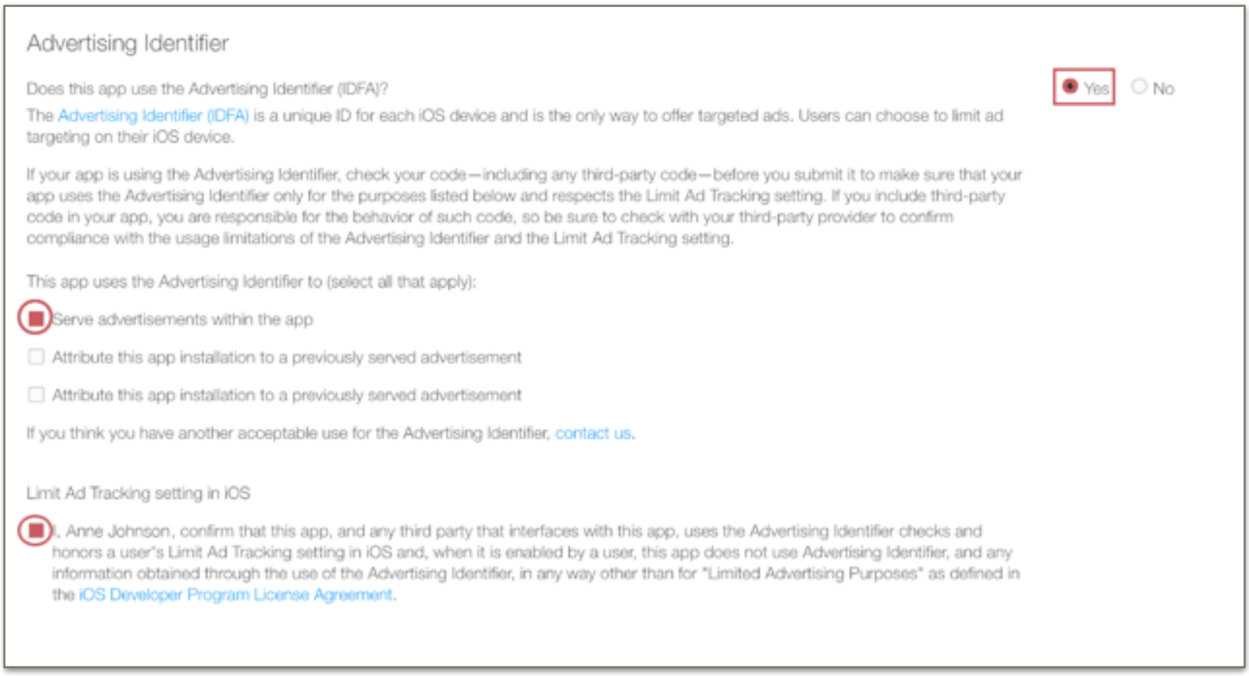
You can now start sending Push Notifications to all registered devices by using the OneSignal Push console:

- Ad banners setup -

```
let ADMOB_UNIT_ID = "ca-app-pub-9733347540588953/7805958028"
```

•**AdMob setup**: In order for you to display your own AdMob banners you must create an AD UNIT ID on www.apps.admob.com. Once you've got it, replace the red string below with your *AD UNIT ID* on the top of the **GameBoardVC.swift:**

•**Apple Review process:** Sometimes it may happen that an app gets rejected once by Apple reviewers because of ads (iAd or AdMob doesn't matter), that's an issue we encountered too and we can't still understand why since we then re-submitted an app without any change and it got approved.
Anyway, check out the screenshot below, you'll get it after clicking on Submit For Review button on iTunes Connect, check the options shown below and everything should go smoothly.
Also, we suggest you to write something in the **Notes** field of your app's page on iTC telling the reviewer that "this app shows AdMob banners, test it for a while to see them showing up".



Advertising Identifier

Does this app use the Advertising Identifier (IDFA)? ● Yes ○ No

The Advertising Identifier (IDFA) is a unique ID for each iOS device and is the only way to offer targeted ads. Users can choose to limit ad targeting on their iOS device.

If your app is using the Advertising Identifier, check your code—including any third-party code—before you submit it to make sure that your app uses the Advertising Identifier only for the purposes listed below and respects the Limit Ad Tracking setting. If you include third-party code in your app, you are responsible for the behavior of such code, so be sure to check with your third-party provider to confirm compliance with the usage limitations of the Advertising Identifier and the Limit Ad Tracking setting.

This app uses the Advertising Identifier to (select all that apply):

☑ Serve advertisements within the app

☐ Attribute this app installation to a previously served advertisement

☐ Attribute this app installation to a previously served advertisement

If you think you have another acceptable use for the Advertising Identifier, contact us.

Limit Ad Tracking setting in iOS

☑ I, Anne Johnson, confirm that this app, and any third party that interfaces with this app, uses the Advertising Identifier checks and honors a user's Limit Ad Tracking setting in iOS and, when it is enabled by a user, this app does not use Advertising Identifier, and any information obtained through the use of the Advertising Identifier, in any way other than for "Limited Advertising Purposes" as defined in the iOS Developer Program License Agreement.

Good luck with Events, and don’t forget to rate it on your Downloads page!

www.fvimagination.com