



YogaStudio - Ultimate Yoga App

Universal app + Back4app backend + AdMob
Push notifications

YogaStudio is a full app that you can use to create a powerful Yoga Exercises app, all data is managed by backend so you can edit Exercises in real time. Made in Xcode 8 and Swift 3, very simple to use, intuitive and have a powerful **Parse** backend powered by back4app.com (**you don't need to configure you're own server and its FREE**). You can create your Yoga app in minutes! All data is managed by back{4}app (Parse) dashboard.

With **YogaStudio** you can custom unique Yoga & Meditation classes with Videos. Users can choose exercises from categories menu, preview Yoga Videos inside the app, add exercises to favorites, search and find Yoga exercises by keywords, share exercises by email or socials, and much more!

YogaStudio include a full detailed Yoga exercises inside with Photo slider, exercise level, exercise ability, exercises time, exercises pose, exercise intensity, exercise focus and a step by step instructions.

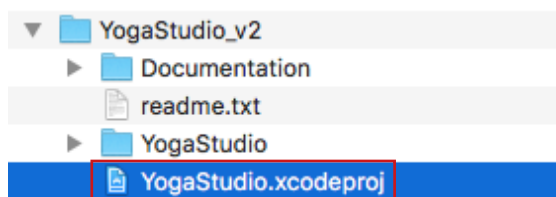
YogaStudio use **OneSignal** to send in realtime push notifications to your users, its FREE, just create your own login access.

This Project comes with a **Sketch 3** artwork file for easy reskin in no time.

Ready to Appstore, very simply to reskin, just configure back{4}app dashboard, configure your **OneSignal** keys for Push notifications and your done!

Configuration

1. Located the **.xcodeproj** file within that folder
2. Double click and launch it in Xcode 8.x or newer




Important:

Parse announced that they are shutting down their hosting servers but **NOT THE PARSE SDK**, the Parse SDK now is opensource and continues its development, so we've moved the hosting to a powerful new service called back{4}app.

Create your Bundle ID on iTunes Developers portal

You may need to create your Bundle ID on iTunes Developers Portal.



Name: GoCityCK


Prefix: 7GQEY52HJQ

Application Services:

Service	Development	Distribution
App Group	<input type="radio"/> Disabled	<input type="radio"/> Disabled
Associated Domains	<input type="radio"/> Disabled	<input type="radio"/> Disabled
Data Protection	<input type="radio"/> Disabled	<input type="radio"/> Disabled
Game Center	<input checked="" type="radio"/> Enabled	<input checked="" type="radio"/> Enabled
HealthKit	<input type="radio"/> Disabled	<input type="radio"/> Disabled
HomeKit	<input type="radio"/> Disabled	<input type="radio"/> Disabled
Wireless Accessory Configuration	<input type="radio"/> Disabled	<input type="radio"/> Disabled

Configured Bundle IDs in Xcode

You may need to change your Bundle ID & Team ID generated on iTunes Devs portal.

 < > YogaStudio

PROJECT

YogaStudio

TARGETS

YogaStudio

▼ Identity

Display Name

YogaStudio

Bundle Identifier

com.yourdomain.yogastudio

Version

2.0

Build

1

▼ Signing

☒ Automatically manage signing

Xcode will create and update profiles, app IDs, and certificates.

Team

Provisioning Profile

Xcode Managed Profile ⓘ

Signing Certificate

iPhone Developer: Ricardo Schuaman Campoverd...

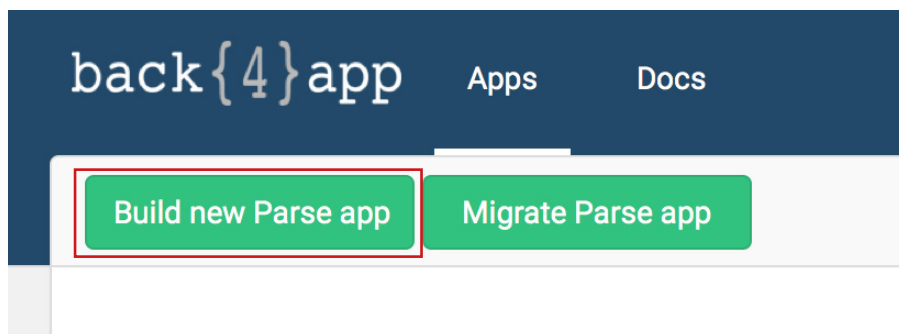
Configure your “Configs.swift”

The **Configs.swift** file contains most of the keys/strings you have to set in order to customize your app and get connected to your own Back4app database. So please insert your own data.

Parse Database setup with back{4}app

Before using the this app you need to register and create and configure your own app on:
<http://www.back4app.com>

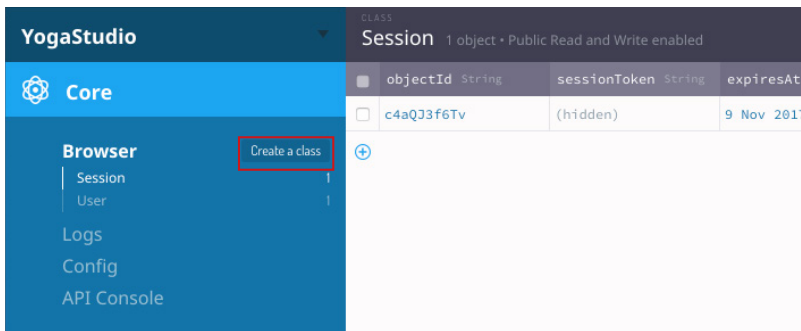
Once you'll be signed in in back4app, simply click on the Build new Parse app button, give your app a name then you'll be redirected to the setting page where you'll be able to copy the App ID and Client Key to replace the existing ones into **Configs.swift**



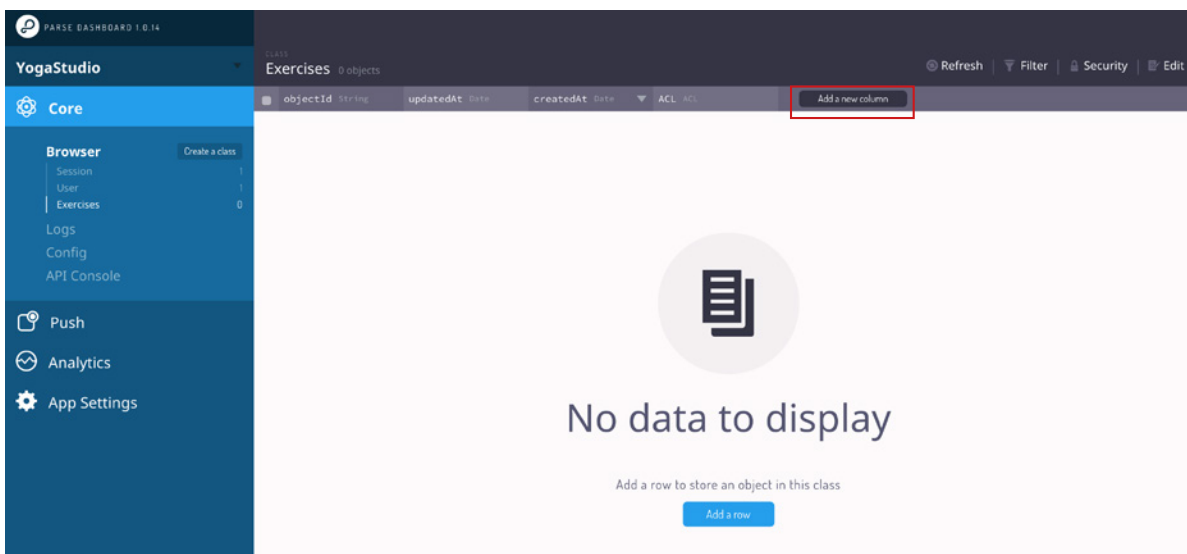
OK at this point you need to create a **Exercises Class** in your dashboard to add your own Exercises. It's super easy just follow this steps:

Create a Class in your dashboard

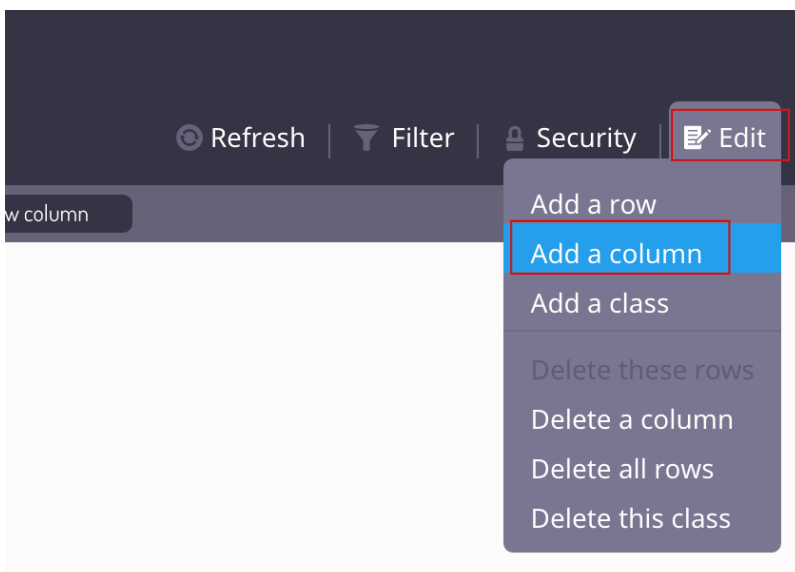
1. Go to your Back4App dashboard and click on **Create Class** and called this class Exercises.



2. Now inside your Exercise class created click to **Add a new column** to populate your tables.



3. At this point you need to create the tables titles that contain all the exercis info. Just click **Edit > Add a column**. Named the column and type of data with the info below.



Add a new column
Store another type of data in this class.

What type of data do you want to store? String

What should we call it?
Don't use any special characters, and start your name with a letter. title

Never mind, don't. Add column

Attribute for each column

Your Column name

Add this 15 columns with its respective attributes and then click **Add column** for each one.

title > String
image1 > File
image2 > File
image3 > File
titleLowercase > String
category > String
aboutExercises > String
ability > String
time > String
poses > String
intensity > String
focus > String
youtube > String
videoTitle > String
steps > String

Perfect! now you're ready to add exercises content inside this columns.

Click Edit > Add row and populate the 15 columns you inserted before with your Exercise details

CLASS Exercises 0 objects

Refresh Filter Security Edit

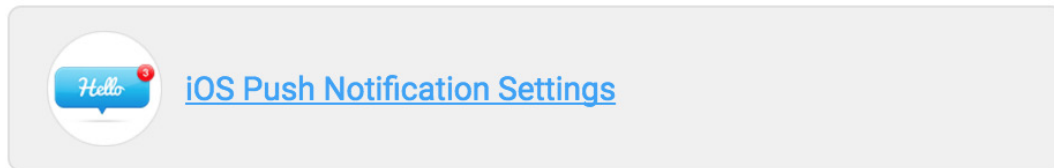
objectId	String	updatedAt	Date	createdAt	Date	ACL	ACL	title	String	image1	File	image2	File
<input type="checkbox"/>	(undefined)	(undefined)	(undefined)	(undefined)	(undefined)	Public	Read + Write	(undefined)	(undefined)	(undefined)	(undefined)	(undefined)	(undefined)

Add a row
Add a column
Add a class
Delete these rows
Delete a column
Delete all rows
Delete this class

Then you can Build an Run your app from Xcode!

Push Notifications

1. You can sending Push Notifications to all registered devices by using the back4app Push web console.
2. Read this tutorial to create your own **APN** for Development and Production:
<http://docs.back4app.com/docs/ios/push-notifications-with-dashboard-and-swift/>
3. Now that you have your Development and Production .p12 certificates, you must upload them in your iOS Push Notification Settings panel:



5. Upload your two **.p12** files and click to Send button.

A screenshot of a web form titled "Apple Push Certificates". At the top is a blue header with a white logo. Below the header, there are two columns labeled "BUNDLE ID" and "TYPE". Under the "BUNDLE ID" column, there is a section titled "Add New Certificate". Below this title, there is a label "Certificate file *" followed by a "Choose File" button. At the bottom of the form, there are two buttons: "BACK" and "SEND". The "SEND" button is highlighted with a red border.

Perfect! you're done, now you can send push notifications to all registered users by the Push web console on your back4app.

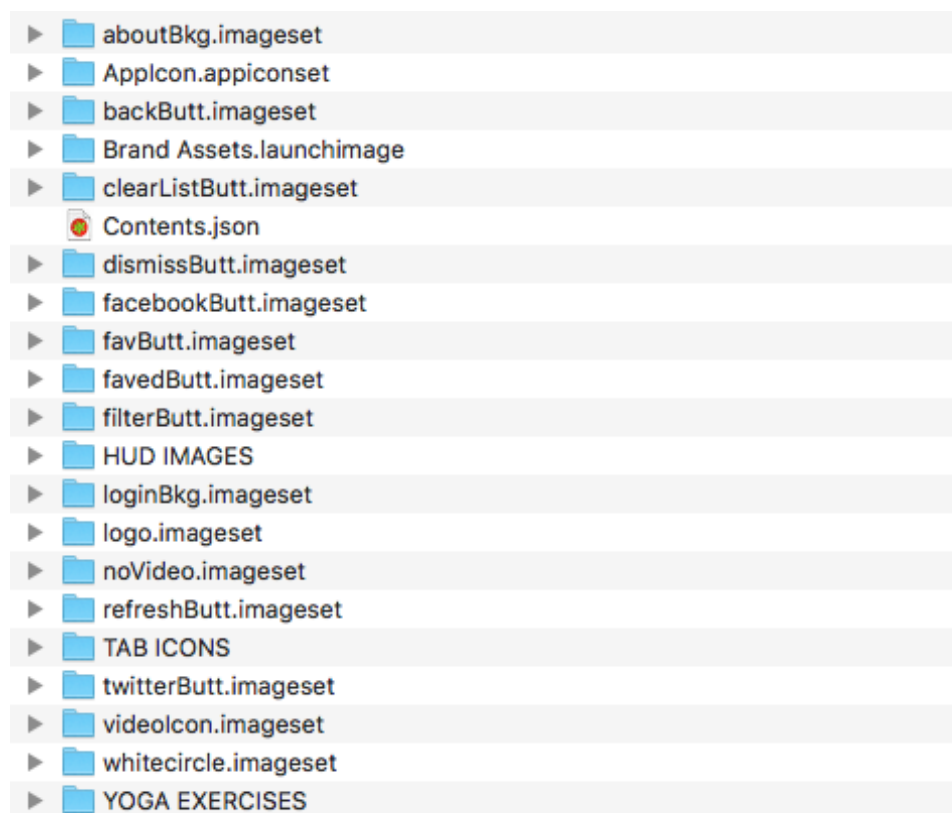
AdMob Ads

Go into **Configs.swift** at line 18 in XCode and replace the App ID with your AdMob ID.

```
var ADMOB_UNIT_ID = "ca-app-pub-3940256099942544/6300978111" // Your AdMob ID
```

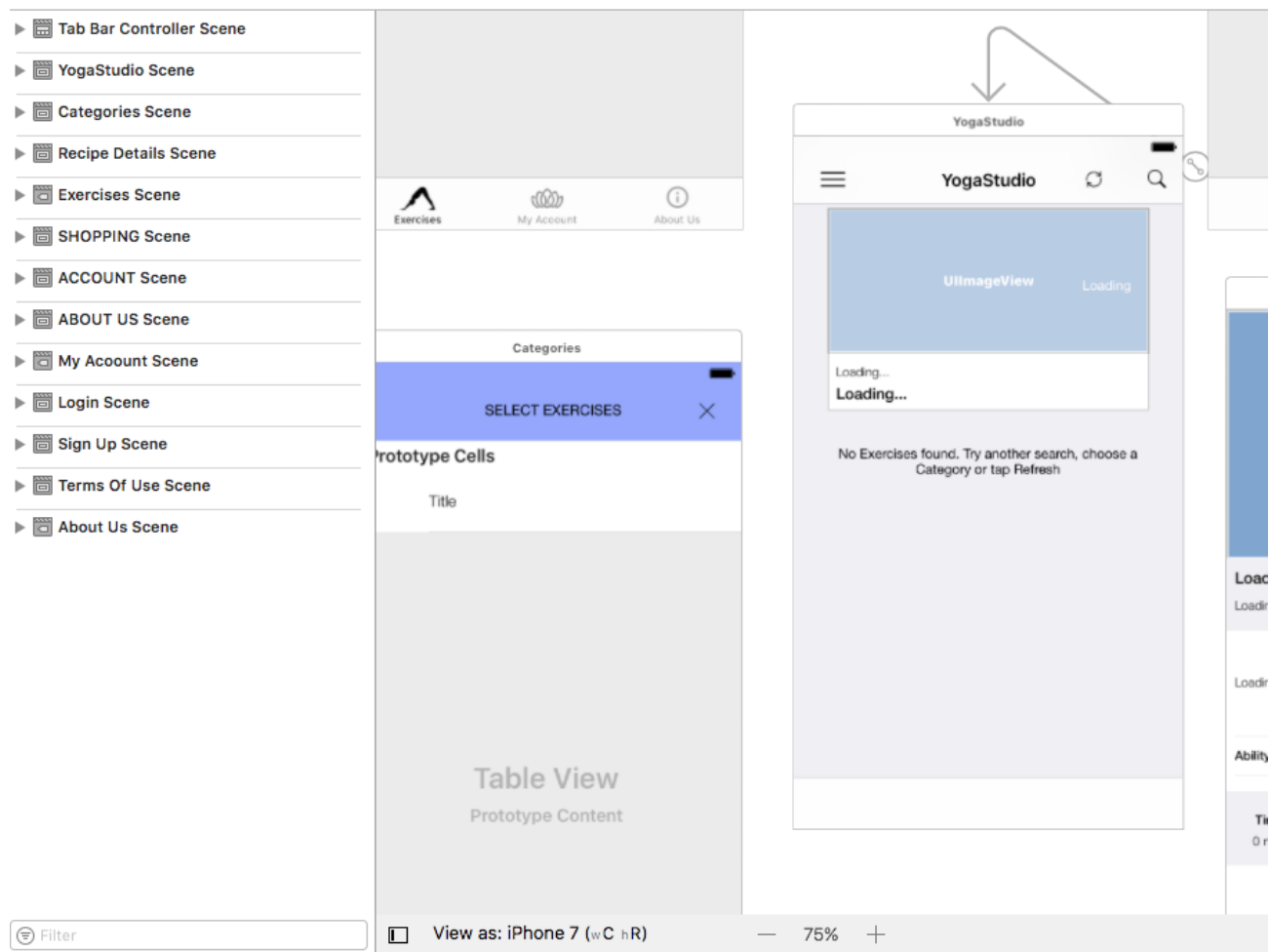
Exchanging Assets

The XCode Project contains a folder “**Assets.xcassets**”. Exchange your assets for all the icons inside this directory via drag & drop. Please ensure you use the dimensions of the images already included within this project.



Customize storyboard






















The **Main.storyboard** in the project can be customized very easy. Just change colors and fonts.



Frameworks

Your frameworks and libraries must match the screenshot below.

▼ Linked Frameworks and Libraries

Name	Status
 libz.tbd	Required ⇅
 libsqlite3.tbd	Required ⇅
 QuartzCore.framework	Required ⇅
 ParseUI.framework	Required ⇅
 MessageUI.framework	Required ⇅
 EventKitUI.framework	Required ⇅
 EventKit.framework	Required ⇅
 CoreTelephony.framework	Required ⇅
 CoreMedia.framework	Required ⇅
 AVFoundation.framework	Required ⇅
 AudioToolbox.framework	Required ⇅
 AdSupport.framework	Required ⇅
 SystemConfiguration.framework	Required ⇅
 GoogleMobileAds.framework	Required ⇅
 Parse.framework	Required ⇅
 Security.framework	Required ⇅
 MobileCoreServices.framework	Required ⇅
 CoreLocation.framework	Required ⇅
 Bolts.framework	Required ⇅
 CoreGraphics.framework	Required ⇅
 CFNetwork.framework	Required ⇅
+ -	

That's it 😊

Happy coding and don't forget to rate it on your dashboard!

Apple Review Process

Sometimes it may happen that an app gets rejected once by Apple reviewers because of ads (Ad-Mob doesn't matter), that's an issue we encountered too and we can't still understand why since we then resubmitted an app without any change and it got approved.

We suggest you to write something in the **Notes field** of your app's page on iTunes Connect 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 an action taken within this app 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, _____, 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](#).

FAQ

Important: If you encountered at the first launch any error please clean your **Build folder**:

*select **Product** item from Xcode top bar and select **Clean Build Folder***

Support

This code is fully documented and we provide basic support associated with issues concerning the template and the code.

We do not provide support for issues concerning features that are not part of the purchase bundle.