



# PlayTube iOS



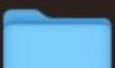
## Getting Started

Follow the steps below to setup your app :

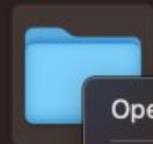
- PlayTube Script, you can get it from [Here](#)
- Download Xcode 2022 [Here](#). Please remember, always use the latest code, latest visual studio, and other latest product versions of products

Here are steps to install the full Xcode application.

- Open the App Store on your Mac and search for Xcode.
- Download Xcode from the App Store.
- After your download is complete, launch Xcode.
- Enter your administrator account credentials.
- Unzip the **PlayTube-IOS** archive, extract it to a new folder, and then open the folder.
- In the main folder, you will find the solution (Name: **PlayTube.xcworkspace** Type: Xcode work Space) double click on it and wait till everything is loaded.
- Install all Project PODS for your Project, Open terminal at folder.



PlayTube-IOs



PlayTu

- Open in New Tab

---

- Move to Bin

---

- Get Info
- Rename
- Compress "PlayTube-IOs"
- Duplicate
- Make Alias
- Quick Look

---

- Copy
- Share...

---

- Tags...

---

- Import from iPhone >
- Quick Actions >

---

- Services >

- Folder Actions Setup...
- MEGA
- New Terminal at Folder
- New Terminal Tab at Folder
- Send files with TeamViewer

PlayTube-IOs — -zsh — 80x24

Last login: Thu Dec 22 19:14:16 on ttys004

[Bookmarked 24-Dec-2022 at 1:20:38 PM]

PlayTube-IOs % pod install

run the command "pod install"

```
PlayTube-IOS — -zsh — 95x38
Last login: Thu Dec 22 19:14:16 on ttys004
[Bookmarked 24-Dec-2022 at 1:20:38 PM]
PlayTube-IOS % pod install
Ignoring ffi-1.15.4 because its extensions are not built. Try: gem pristine ffi --version 1.15.4
Analyzing dependencies
Downloading dependencies
Generating Pods project
Integrating client project
Pod installation complete! There are 35 dependencies from the Podfile and 47 total pods installed.

[!] Automatically assigning platform `iOS` with version `15.2` on target `Playtube` because no platform was specified. Please specify a platform for this target in your Podfile. See `https://guides.cocoapods.org/syntax/podfile.html#platform`.

[!] Automatically assigning platform `iOS` with version `12.2` on target `OneSignalNotificationServiceExtension` because no platform was specified. Please specify a platform for this target in your Podfile. See `https://guides.cocoapods.org/syntax/podfile.html#platform`.

[!] The `Playtube [Debug]` target overrides the `ALWAYS_EMBED_SWIFT_STANDARD_LIBRARIES` build setting defined in `Pods/Target Support Files/Pods-Playtube/Pods-Playtube.debug.xcconfig`. This can lead to problems with the CocoaPods installation
  - Use the `$(inherited)` flag, or
  - Remove the build settings from the target.

[!] The `Playtube [Release]` target overrides the `ALWAYS_EMBED_SWIFT_STANDARD_LIBRARIES` build setting defined in `Pods/Target Support Files/Pods-Playtube/Pods-Playtube.release.xcconfig`. This can lead to problems with the CocoaPods installation
  - Use the `$(inherited)` flag, or
  - Remove the build settings from the target.

[!] Your project does not explicitly specify the CocoaPods master specs repo. Since CDN is now used as the default, you may safely remove it from your repos directory via `pod repo remove master`. To suppress this warning please add `warn_for_unused_master_specs_repo => false` to your Podfile.
```

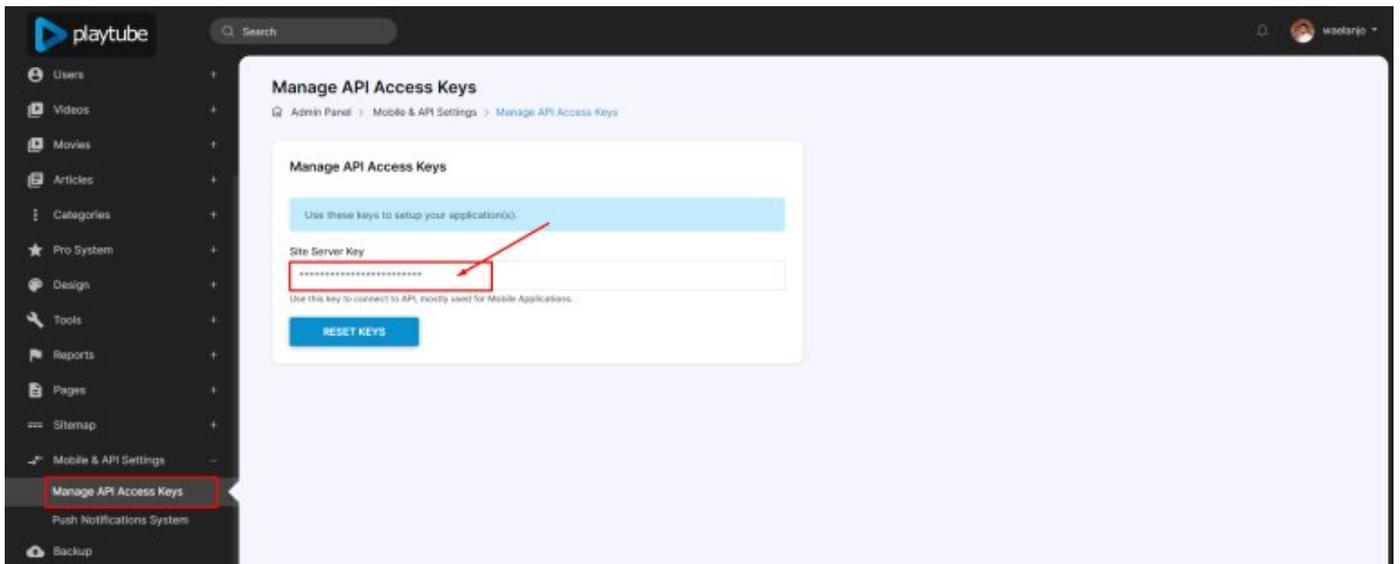
It will take some time, Wait until it is completed.

# Articles

- ☰ Verify Application
- ☰ Deep Links To App Content

## Verify Application

- You can get a purchase code from [Envato](#).
- Get Your *API Server\_Key* which is located [here](#) copy it.



- Create your Cert key for your application from [Doughouzlight-License](#)  
PlayTube Provides Triple DES algorithm encryption system + AES 256-Bit Encryption in your mobile application to safe your own information and your own server side keys from hackers and crackers, once you are a real buyer you will not fear any cracking or **unpacking APK** actions by eligible black hat people, which may lead to leaking your sensitive server side data to the public.

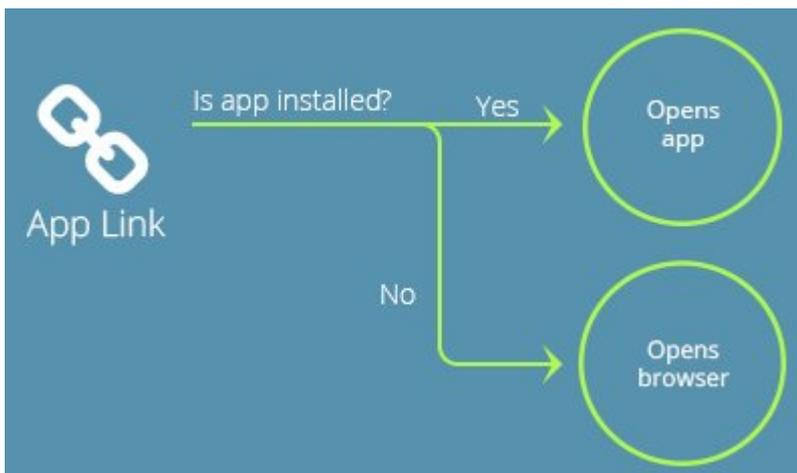


---

## Deep Links To App Content

When a clicked link or programmatic request invokes a web URI intent, the IOS system tries each of the following actions, in sequential order, until the request succeeds:

- Open the user's preferred app that can handle the URI, if one is designated.
- Open the only available app that can handle the URI.
- Allow the user to select an app from a dialog.



---

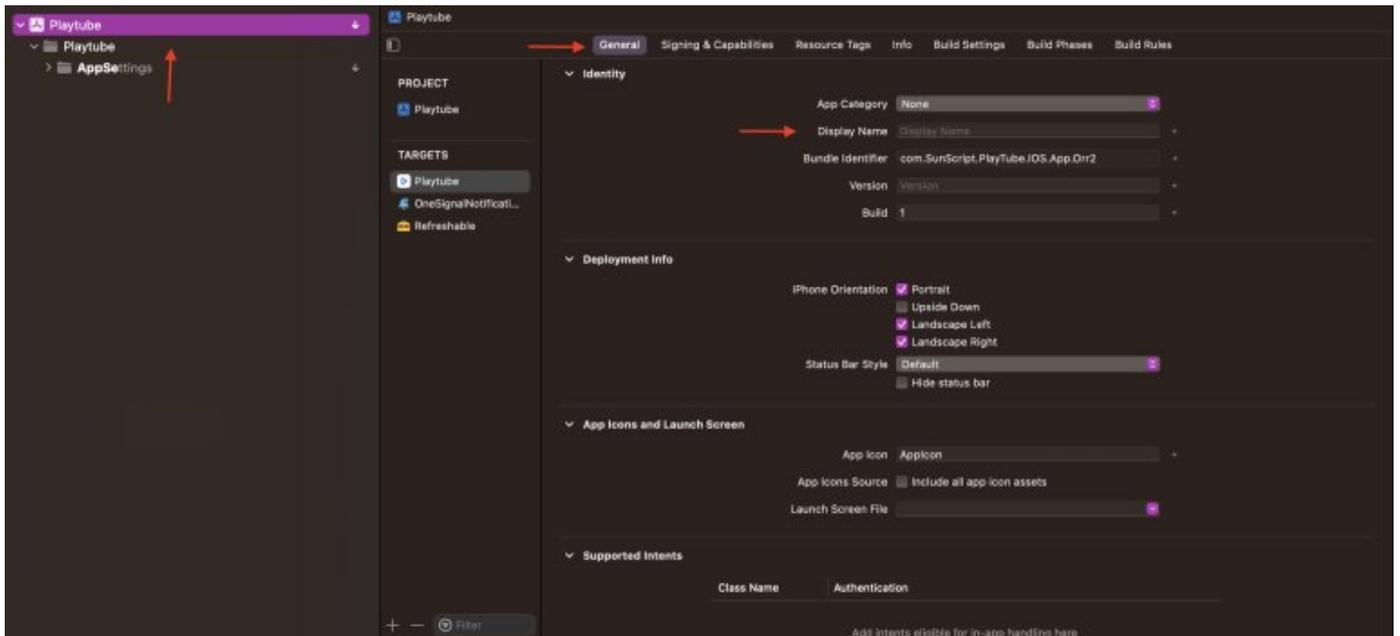
## Theme Setting

### Articles

- ☰ Colors and theme
- ☰ Bundle identifier
- ☰ Logo Icons and splash Screen

## Application Name

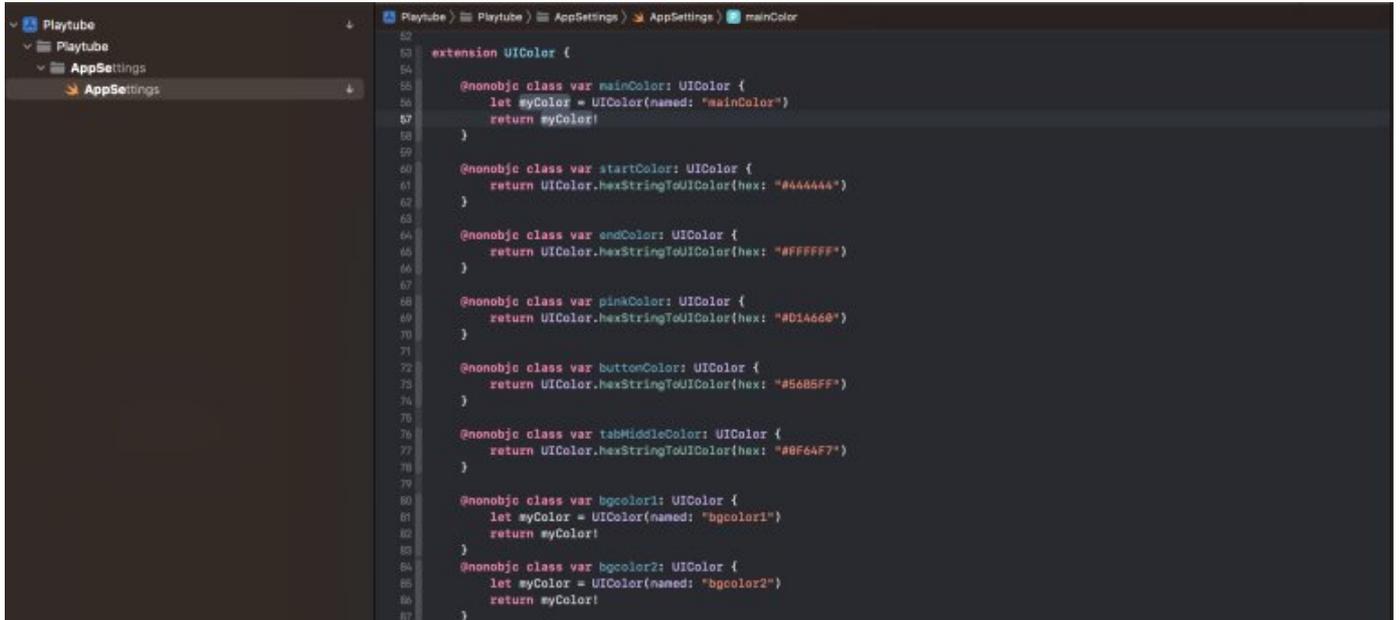
From your main solution right click on the **QuickDate > General > Display Name** Change the names as you like and the versions also depend on your [AppStoreConnect](#) the last version if you have one.



## Colors and theme

To customize and change the main color of the application follow the steps below.

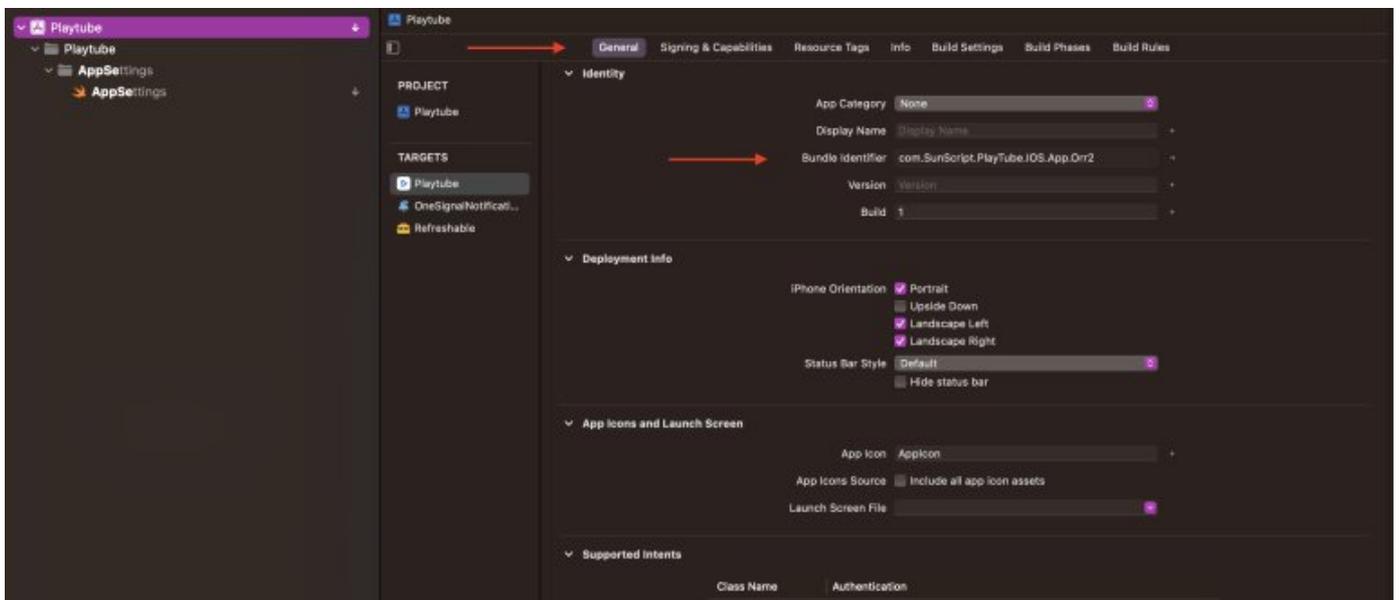
Go to App Constants class and change the colors below to your own colors.



```
52
53
54 extension UIColor {
55
56     @nonobjc class var mainColor: UIColor {
57         let myColor = UIColor(named: "mainColor")
58         return myColor!
59     }
60
61     @nonobjc class var startColor: UIColor {
62         return UIColor.hexStringToUIColor(hex: "#444444")
63     }
64
65     @nonobjc class var endColor: UIColor {
66         return UIColor.hexStringToUIColor(hex: "#FFFFFF")
67     }
68
69     @nonobjc class var pinkColor: UIColor {
70         return UIColor.hexStringToUIColor(hex: "#D14668")
71     }
72
73     @nonobjc class var buttonColor: UIColor {
74         return UIColor.hexStringToUIColor(hex: "#5685FF")
75     }
76
77     @nonobjc class var tabMiddleColor: UIColor {
78         return UIColor.hexStringToUIColor(hex: "#8F64F7")
79     }
80
81     @nonobjc class var bgcolor1: UIColor {
82         let myColor = UIColor(named: "bgcolor1")
83         return myColor!
84     }
85
86     @nonobjc class var bgcolor2: UIColor {
87         let myColor = UIColor(named: "bgcolor2")
88         return myColor!
89     }
90 }
```

## Bundle identifier

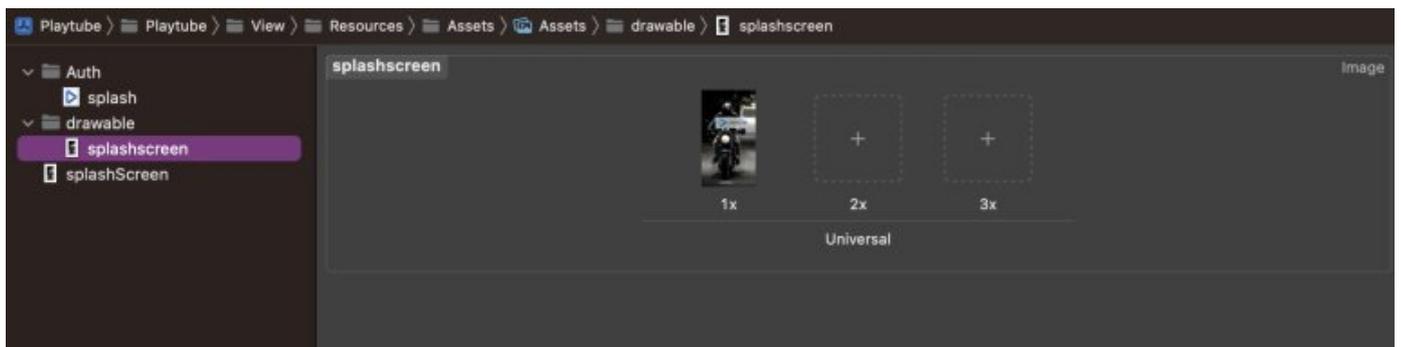
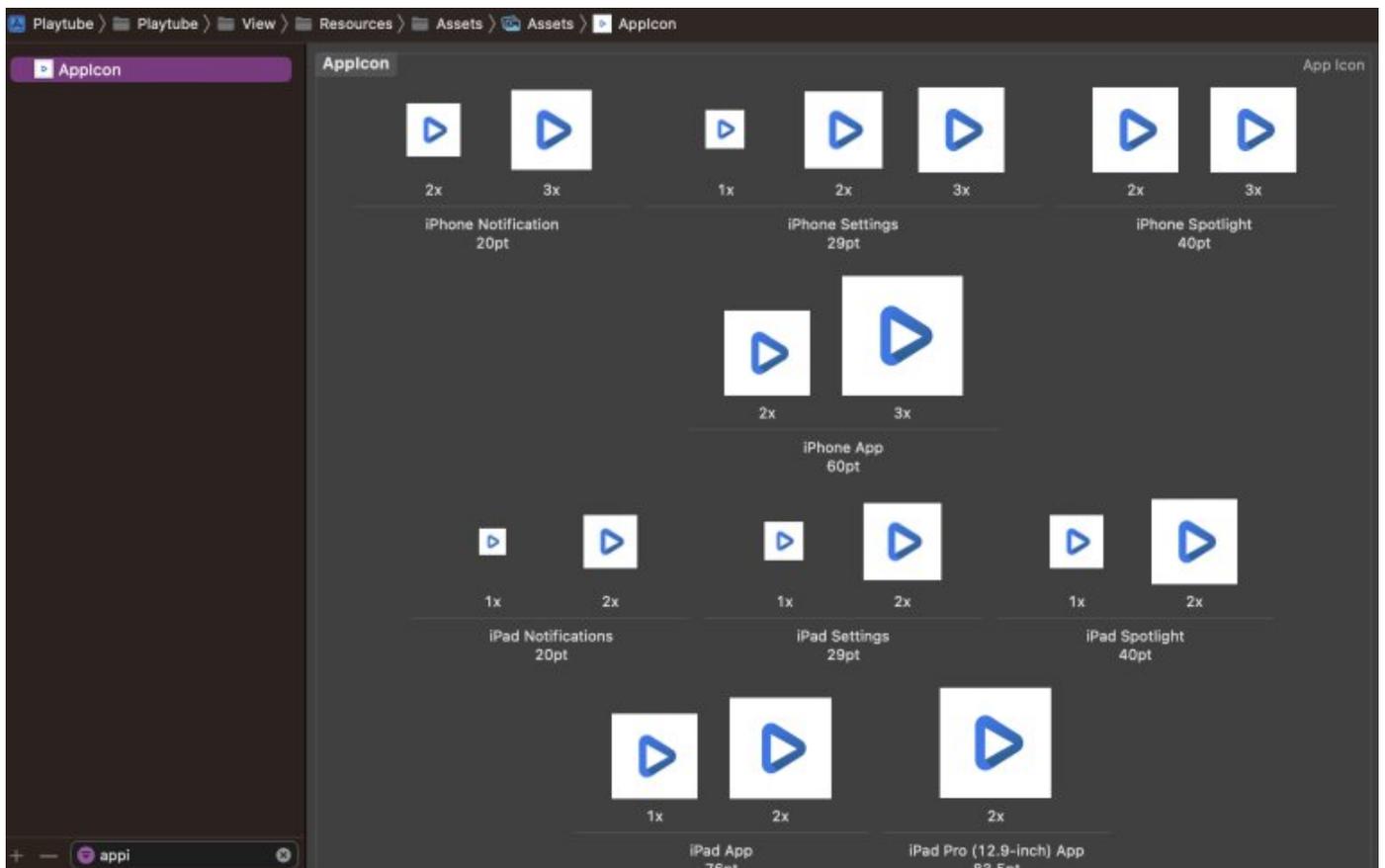
Change the package name by edit "Bundle identifier" for App "TARGETS"



# Logo Icons and splash Screen

Change logo of app by changing the icons in the file named Applcon in assets folder.

For the accuracy of the icon and logo, please use this website <https://appicon.co>





---

## General Setups

### Articles

-  OneSignal Notification
-  Google Map/Places
-  Firebase Account

---

## OneSignal Notification

### What is OneSignal?

[OneSignal](#) is a high-volume and reliable push notification service for websites and mobile

applications. They support all major native and mobile platforms by providing dedicated SDKs for each platform, a Restful server API, and an online dashboard for marketers to design and send push notifications.

- Sign in to the OneSignal Console at <https://onesignal.com>.
- Click on **New App/Website**
- Choose your app name and platform then click next.
- Fill out the form with your website information, then click save.
- On the next page, ignore the page and click on Finish.
- On the top navbar, click on **Keys & IDs**
- Grab your OneSignal App ID & Rest API Key then go to **Admin Panel -> API Settings -> Push Notifications Settings**
- Edit the following options for IOS:
  1. **OneSignal APP ID** – Enter the OneSignal App ID you created in the previous chapter
  2. **REST API Key** – Enter the Rest API Key you created in the previous chapter
- Finally, go to the file **ControlSettings Struct** in the project and add the app id.

```
/*  
 * change it with your onesignal app ID  
 */  
static let oneSignalAppId = "f24d6c75-6e6d-4797-aa30-553bda1c32e9"  
/*
```

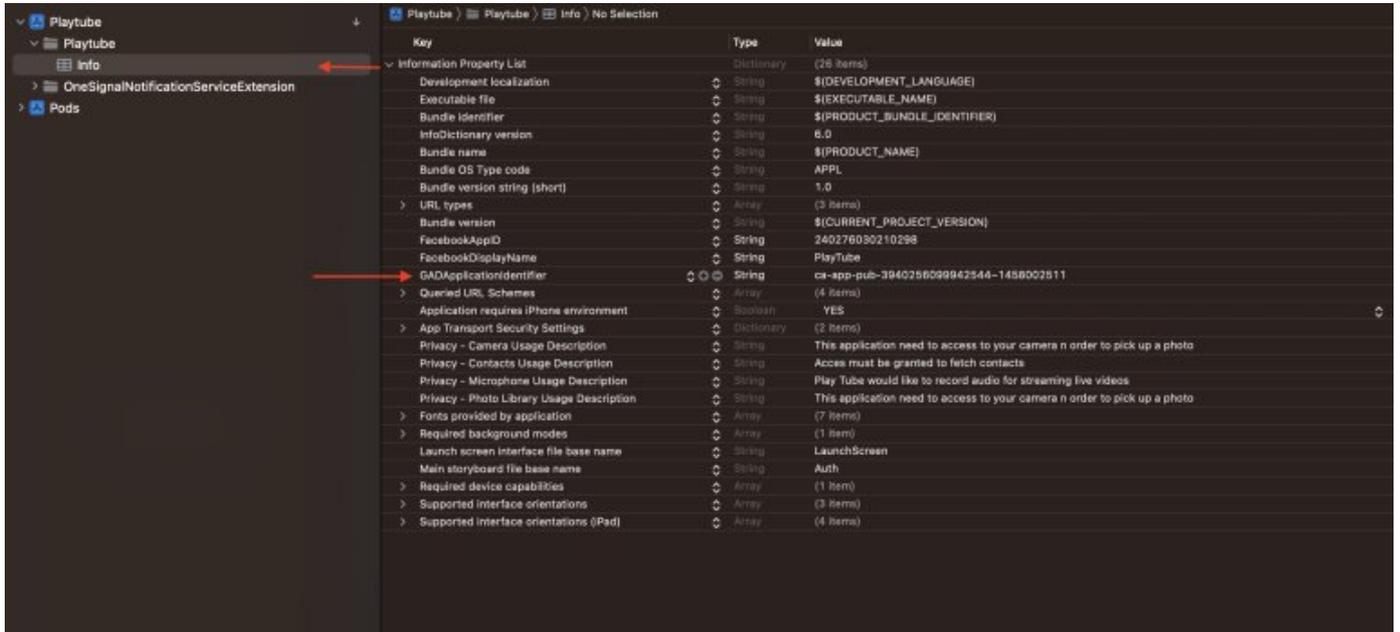
---

## Google Map/Places

### Google Maps API is required for GEO and viewing Google Maps.

- Go to the [Google API Manager](#) console and login using your Google credentials.
- After this click on [Create Project](#) link to create a new project.
- Enter **Project Name** and click on **Create** button.
- Select the project from the dropdown menu beside **Google Cloud Platform**.
- Click on [APIs & Services](#) then click on **+ENABLE APIS AND SERVICES**.
- Search for **Maps Embed API** and enable it.
- Search for **Maps Static API** and enable it.

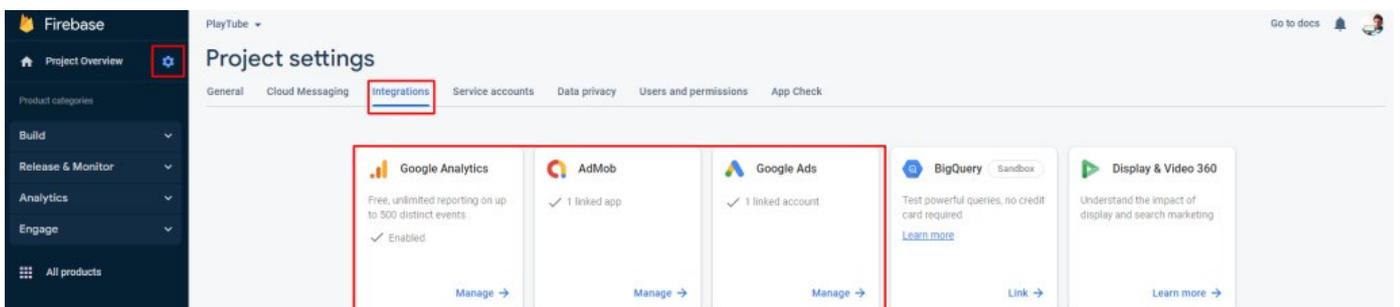
- Once enabled, click on [Credentials](#), on the top nav-bar, click on **+Create credentials** then click on the API key
- Grab the key and Click **Close**.
- Finally, go to the file `info.plist` in the project and add the key.



## Firebase Account

[Firebase](#) is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

- After creating a firebase account from the [link](#) you will need to create a new project as well.
- then go to **project settings**.
- Go to the **Integrations** tab and select the item below as the screenshot:



**Customize your nav!**  
You can now focus your console experience by customizing your navigation  
[Learn more](#) [Got it](#)

<b>Google Play</b> Enable integrations that unlock features in Firebase, Google Analytics, and Google Play <a href="#">Learn more</a> No matching Play apps found	<b>Slack</b> Send your team important alerts detected by Firebase <a href="#">Install →</a>	<b>Jira</b> Automatically create issues in Jira <a href="#">Install →</a>	<b>PagerDuty</b> Send your team important alerts detected by Firebase <a href="#">Install →</a>	<b>Cloud Logging</b> View, filter, and search logs from your projects in real-time <a href="#">Learn more</a> <a href="#">Link →</a>
--	---	---	---	---

## To connect with AdMob

**App settings**

**App info**

App name ⓘ	WoWonder Timeline
App ID ⓘ	ca-app-pub-5135691635931982~1668785995
Package name ⓘ	com.wowondertimeline.app
App stores ⓘ	Google Play
Approval status ⓘ	Ready
Linked services ⓘ	Link your AdMob apps or account to other Google services <a href="#">Manage linked services</a>

**Ad serving settings**

Frequency capping ⓘ	5 impressions per user per 1 minute   2 ad unit-level cap(s)
---------------------	--

Add your Package android app name and pre

# Advertisement Setting

## Articles

## Integrate AdMob (Google ADS)

After creating a [Google AdMob](#) account and you are all ready to start your own AdMob ads system on your mobile app.

You will need to follow a few steps before seeing your ADS appearing in your mobile application.

- Find your app IDs & ad unit IDs
  - An app ID is a unique ID number assigned to your apps when they're added to AdMob. The app ID is used to identify your apps.
- An ad unit ID is a unique ID number assigned to each of your ad units when they're created in AdMob. The ad unit ID is added to your app's code and used to identify ad requests from the ad unit.
- Find an app ID:
  1. Sign in to your AdMob account at <https://apps.admob.com>.
  2. Click Apps in the sidebar.
  3. Click View all apps.
  4. Click the icon in the App ID column to copy the ID of an app.
- Find an ad unit ID:
  1. Sign in to your AdMob account at <https://apps.admob.com>.
  2. Click Apps in the sidebar.
  3. Click the name of the app associated with the ad unit. Note: If you don't see it in the list of recent apps, click View all apps. Then, click the name of the app.
  4. Click Ad units in the sidebar.
  5. Click the icon in the Ad unit ID column to copy the ID of an ad unit.

Change the Key In APP Setting Struct

The screenshot shows the Xcode interface. On the left, the Project Navigator displays a hierarchy: Playtube > AppSettings > AppSettings. The AppSettings pod is selected. On the right, the code editor shows the following Swift code:

```
21 class AppSettings{
22     static let showSocialLogin = true
23     static let googleClientKey = "497109148599-u0g40f3e5uh53286hdropsj10v505tra1.apps.googleusercontent.com"
24     static let omeSignalAppId = "b64b10e7-4cc4-450b-adcb-84e980e6b8ff"
25     static let addUnitId = "ca-app-pub-3940256099942544/2934736716"
26     static let interstitialAddUnitId = "ca-app-pub-3940256099942544/4411468918"
27     static let isPurchase = false
28 }
```

## Socail Login

### Articles

 Facebook Login

## FaceBook Login

- Log in to your Facebook account.
- Go to [Facebook for Developers](#), click on My Apps and press Create App.
- Select Consumer. and click Next.
- Set the Display Name of your application.
- Enter the Contact Email.
- Click on Create App.
- Navigate to Facebook Login and press the Set up button.
- Select IOS from the displayed platforms.
- Add your package name that uniquely identifies your IOS app
- Put that key in Info.plist file

The screenshot shows the Xcode interface with the Info.plist file open. The following keys are visible:

FacebookAppID	String	381859194065993
FacebookDisplayName	String	QuickDate

To show and hide facebook login button, You can go to **struct** ControlSettings. In that struct you can change the setting accordingly

```
struct ControlSettings {  
  
    //    static let AppName = "QuickDate"  
  
    // MARK: Social Login Settings  
    // Change these values according to your choice  
    static let showFacebookLogin = true  
    static let showGoogleLogin = true  
    static let showWoWonderLogin = true  
    static let showAppleLogin = true
```

## Archives

July 2022

May 2020

April 2020

## Categories

Design

Docuemntation

FAQs

[Laravel](#)

[Magento](#)

[Menu](#)

[Uncategorized](#)

[WooCommerce](#)

[WordPress](#)

[Blog](#)

[Knowledgebase](#)

© 2022 All Rights Reserved by DoughouzLight