Edelivery Provider Documentation

➢Install Xcode On Mac:

1. Open the App Store on your mac.
2. Sign in.
3. Search for Xcode.
4. Click install or update.

Please check the link to download xcode from the app store:-
https://apps.apple.com/us/app/xcode/id497799835?mt=12

Note: To install a specific version or latest version on your mac system your Mac OS must need to be compatible with the version of xcode. For example, to install the latest xcode version 13.1 from the App Store, it requires that your system is updated with Mac OS version 11.3 or later.

For more details you can check this link :
https://medium.com/@LondonAppBrewery/how-to-download-and-setup-xcode-10-for-ios-development-b63bed1865c

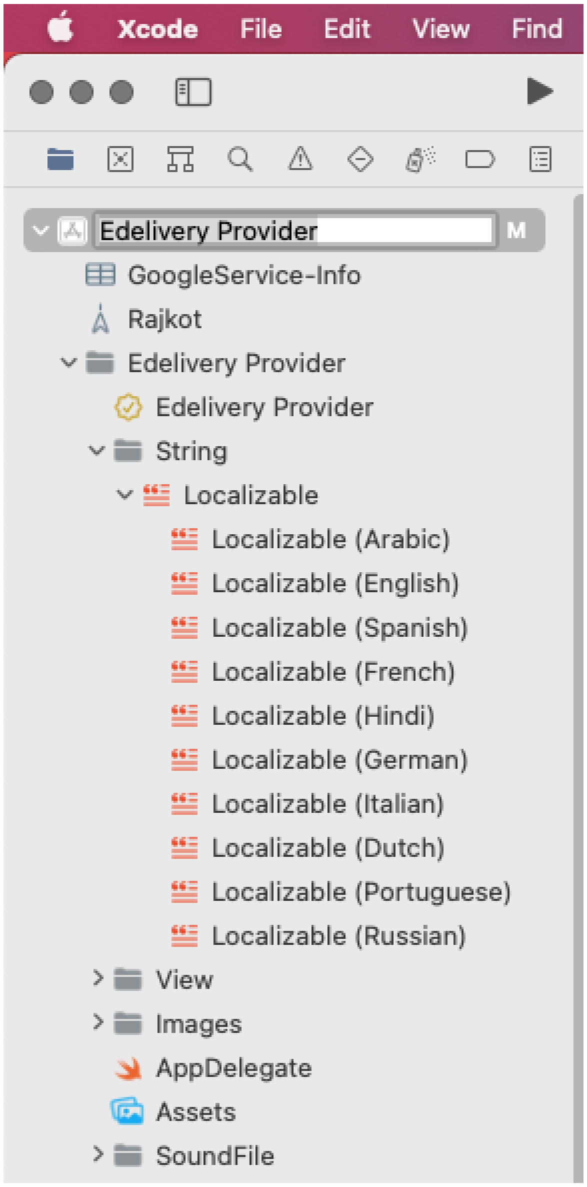
- ## Changes In Projects (iOS)

1. Open Project in Xcode

File->open->Select .xcworkspace file of your project which is located on your system.

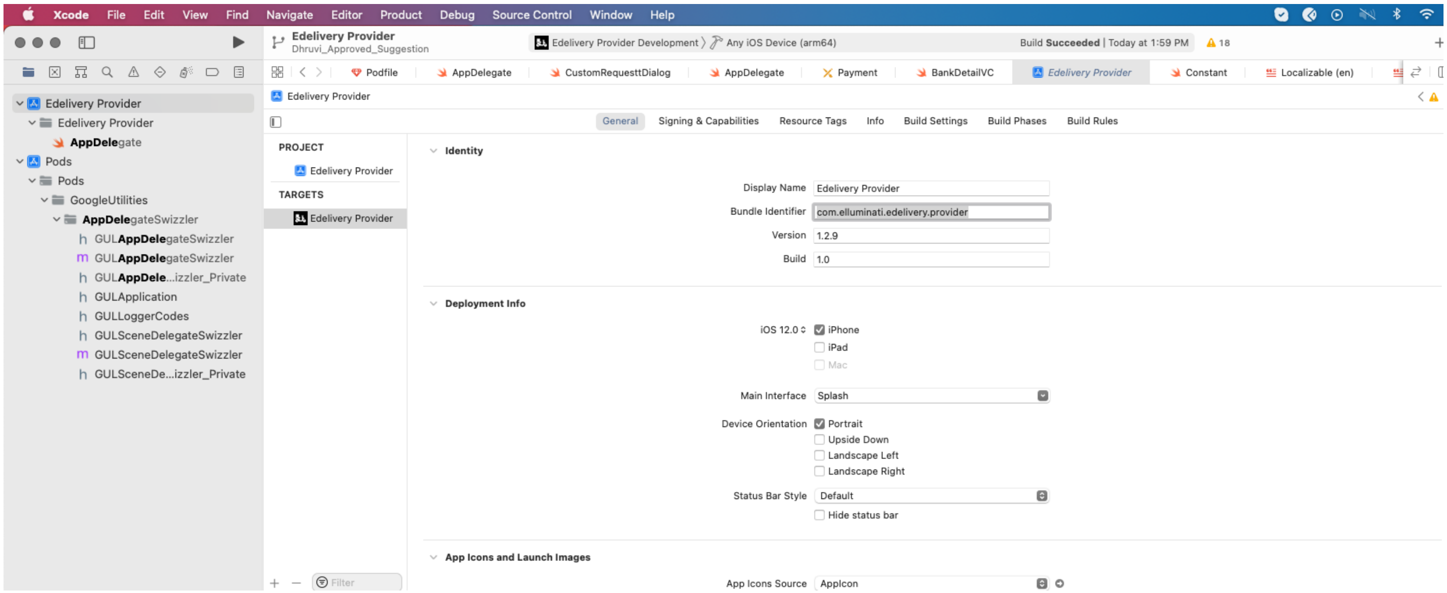
2. Change App name

In the project navigator, select the project and Rename it



3. Change bundle identifier

1. In the project navigator, select the project and your target to display the project editor.
2. Click General Tab
3. In the identity section change the Bundle Identifier field. See the screenshot below for that.



Identity

Display Name: Edelivery Provider
Bundle Identifier: com.elluminati.edelivery.provider
Version: 1.2.9
Build: 1.0

Deployment Info

iOS 12.0 ☑ iPhone
□ iPad
□ Mac

Main Interface: Splash

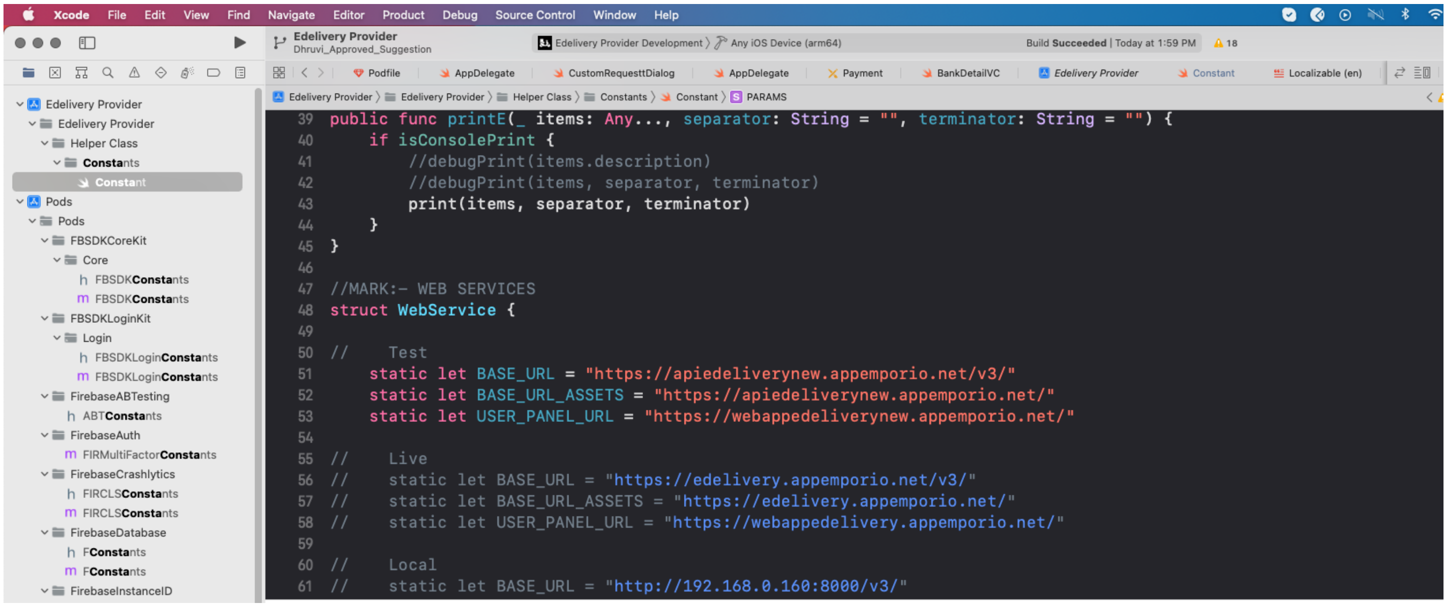
Device Orientation: ☑ Portrait
□ Upside Down
□ Landscape Left
□ Landscape Right

Status Bar Style: Default
□ Hide status bar

App Icons and Launch Images

App Icons Source: AppIcon

4. Change BASE_URL from constant file



```
public func printE(_ items: Any..., separator: String = "", terminator: String = "") {
    if isConsolePrint {
        //debugPrint(items.description)
        //debugPrint(items, separator, terminator)
        print(items, separator, terminator)
    }
}

//MARK:- WEB SERVICES
struct WebService {

//     Test
    static let BASE_URL = "https://apiedeliverynew.appemporio.net/v3/"
    static let BASE_URL_ASSETS = "https://apiedeliverynew.appemporio.net/"
    static let USER_PANEL_URL = "https://webappedeliverynew.appemporio.net/"

//     Live
//     static let BASE_URL = "https://edelivery.appemporio.net/v3/"
//     static let BASE_URL_ASSETS = "https://edelivery.appemporio.net/"
//     static let USER_PANEL_URL = "https://webappedelivery.appemporio.net/"

//     Local
//     static let BASE_URL = "http://192.168.0.160:8000/v3/"
```

5. Set Google API key in Constant file

```
static var API_KEY = ""
static var MAP_KEY = ""
```

6. Change App Update URL in Constant file

- You can change Update URL here.

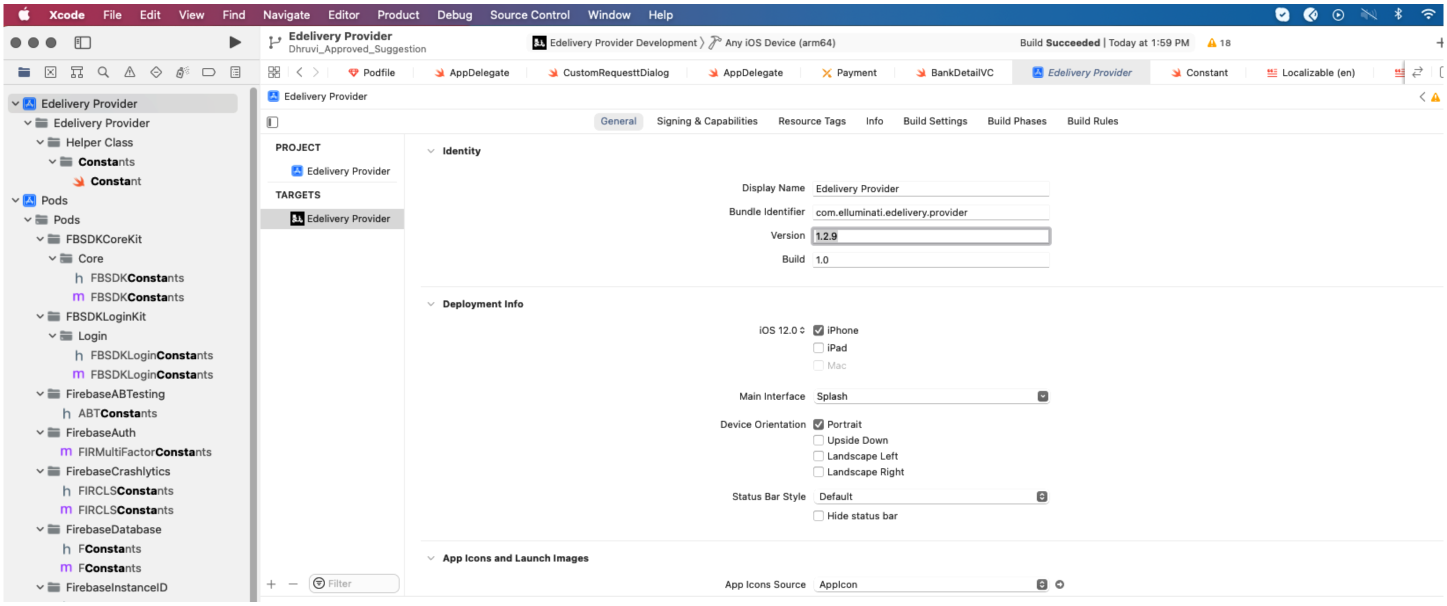
```
static let UPDATE_URL = "https://itunes.apple.com/us/app/id1276556193?ls=1&mt=8"
```

7. Change App name in Localize file

- You can change the app name from the String file.
  File Goto :String folder and change App name in all localized files

8. Change App version number/Build version number

- You can change the app version and build version from the identity section of the general tab.



Identity

Display Name: Edelivery Provider
Bundle Identifier: com.elluminati.edelivery.provider
Version: 1.2.9
Build: 1.0

Deployment Info

iOS 12.0 ☑ iPhone
☐ iPad
☐ Mac

Main Interface: Splash

Device Orientation: ☑ Portrait
☐ Upside Down
☐ Landscape Left
☐ Landscape Right

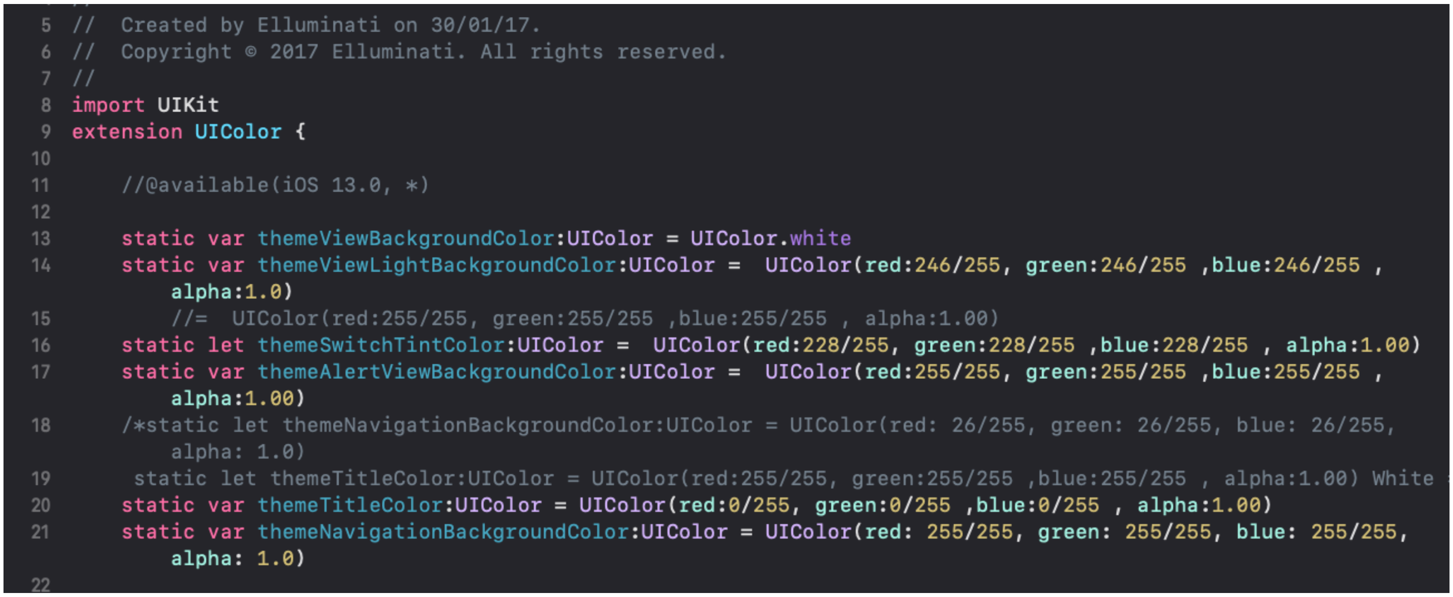
Status Bar Style: Default
☐ Hide status bar

App Icons and Launch Images

App Icons Source: AppIcon

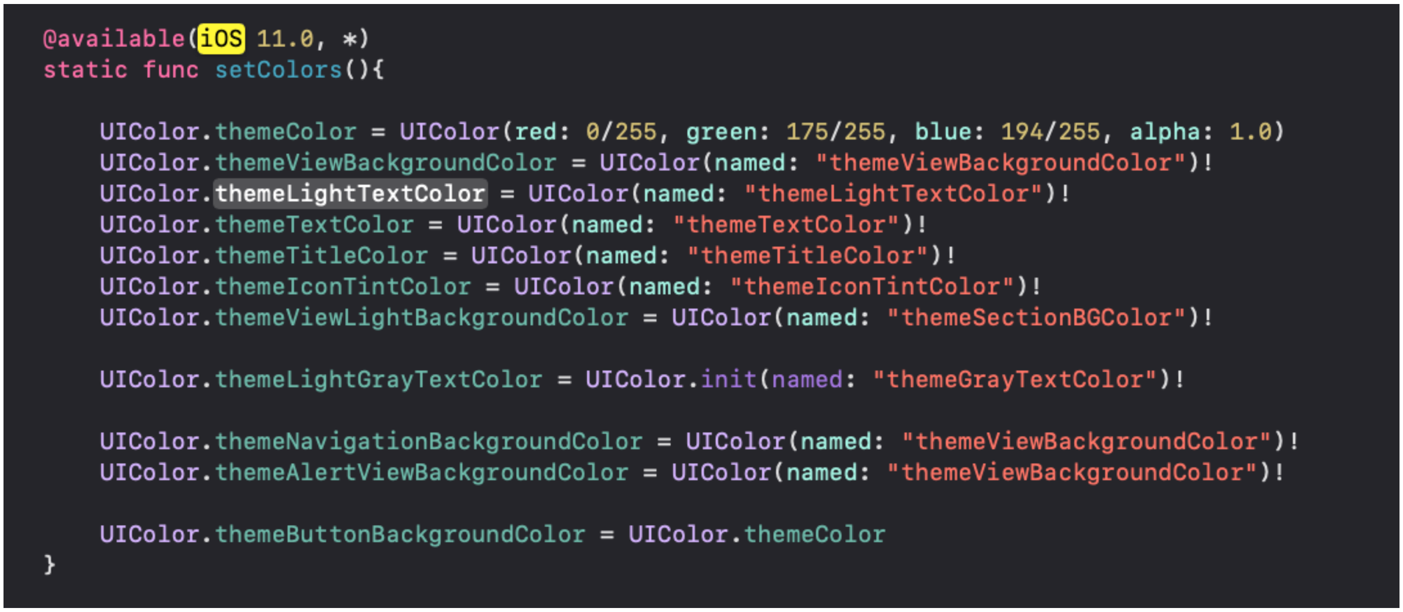
9.Change your Theme color

Goto : HelperClass->AppThemeHelper->myAppTheme.swift , where you can change Section Background Color,Button Background Color,Theme Color and etc



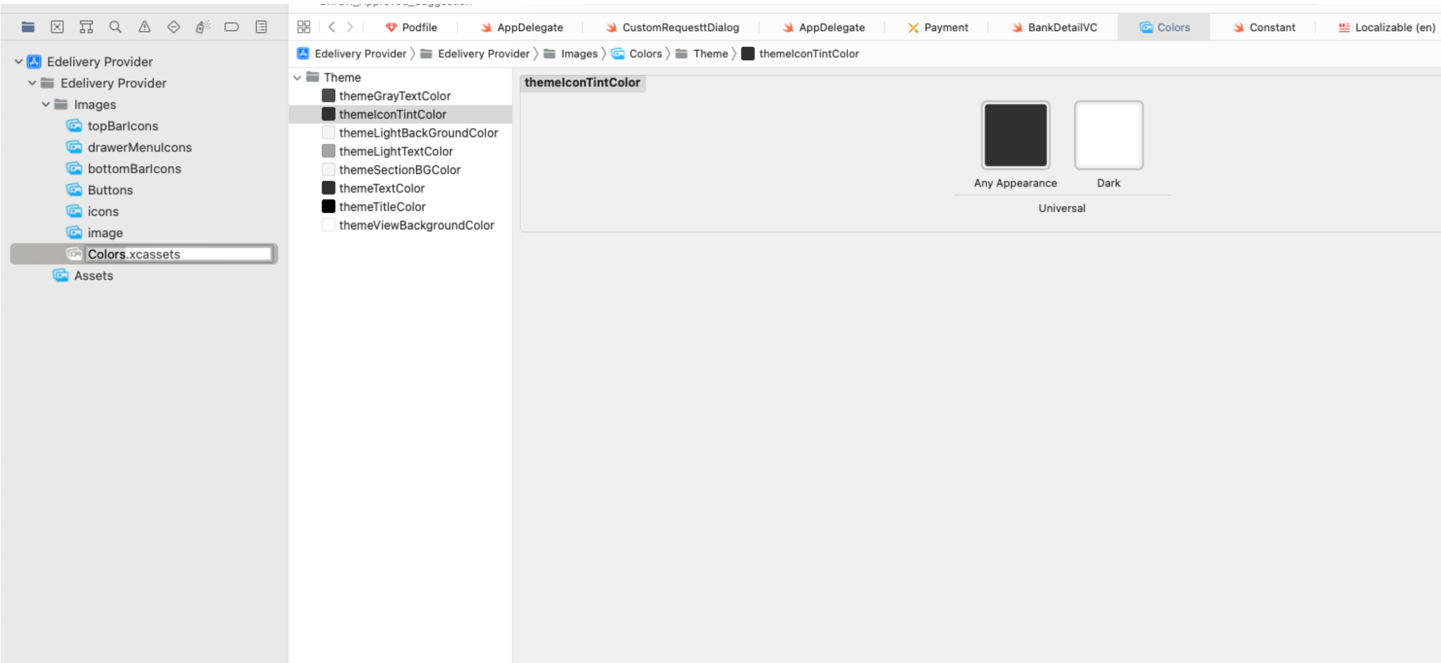
```
//  Created by Elluminati on 30/01/17.
//  Copyright © 2017 Elluminati. All rights reserved.
//
import UIKit
extension UIColor {

    //@available(iOS 13.0, *)

    static var themeViewBackgroundColor:UIColor = UIColor.white
    static var themeViewLightBackgroundColor:UIColor =  UIColor(red:246/255, green:246/255 ,blue:246/255 ,
        alpha:1.0)
        //=  UIColor(red:255/255, green:255/255 ,blue:255/255 , alpha:1.00)
    static let themeSwitchTintColor:UIColor =  UIColor(red:228/255, green:228/255 ,blue:228/255 , alpha:1.00)
    static var themeAlertViewBackgroundColor:UIColor =  UIColor(red:255/255, green:255/255 ,blue:255/255 ,
        alpha:1.00)
    /*static let themeNavigationBackgroundColor:UIColor = UIColor(red: 26/255, green: 26/255, blue: 26/255,
        alpha: 1.0)
     static let themeTitleColor:UIColor = UIColor(red:255/255, green:255/255 ,blue:255/255 , alpha:1.00) White
    static var themeTitleColor:UIColor = UIColor(red:0/255, green:0/255 ,blue:0/255 , alpha:1.00)
    static var themeNavigationBackgroundColor:UIColor = UIColor(red: 255/255, green: 255/255, blue: 255/255,
        alpha: 1.0)

```

- For ios 11.0 and later please follow following screenshot.



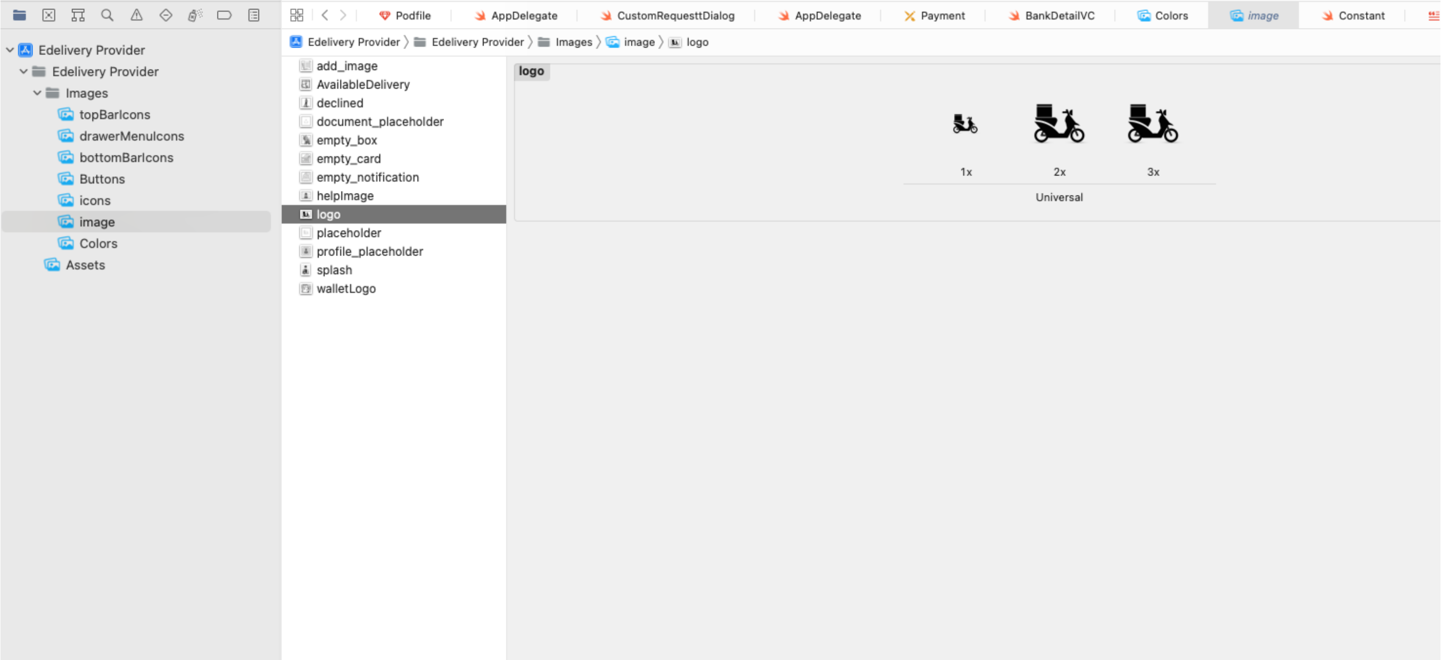
```
@available(iOS 11.0, *)
static func setColors(){

    UIColor.themeColor = UIColor(red: 0/255, green: 175/255, blue: 194/255, alpha: 1.0)
    UIColor.themeViewBackgroundColor = UIColor(named: "themeViewBackgroundColor")!
    UIColor.themeLightTextColor = UIColor(named: "themeLightTextColor")!
    UIColor.themeTextColor = UIColor(named: "themeTextColor")!
    UIColor.themeTitleColor = UIColor(named: "themeTitleColor")!
    UIColor.themeIconTintColor = UIColor(named: "themeIconTintColor")!
    UIColor.themeViewLightBackgroundColor = UIColor(named: "themeSectionBGColor")!

    UIColor.themeLightGrayTextColor = UIColor.init(named: "themeGrayTextColor")!

    UIColor.themeNavigationBackgroundColor = UIColor(named: "themeViewBackgroundColor")!
    UIColor.themeAlertViewBackgroundColor = UIColor(named: "themeViewBackgroundColor")!

    UIColor.themeButtonBackgroundColor = UIColor.themeColor
}
```

Strings like themeIconTintColor and others shown above in screenshots are set in the Colors.xcassets file.



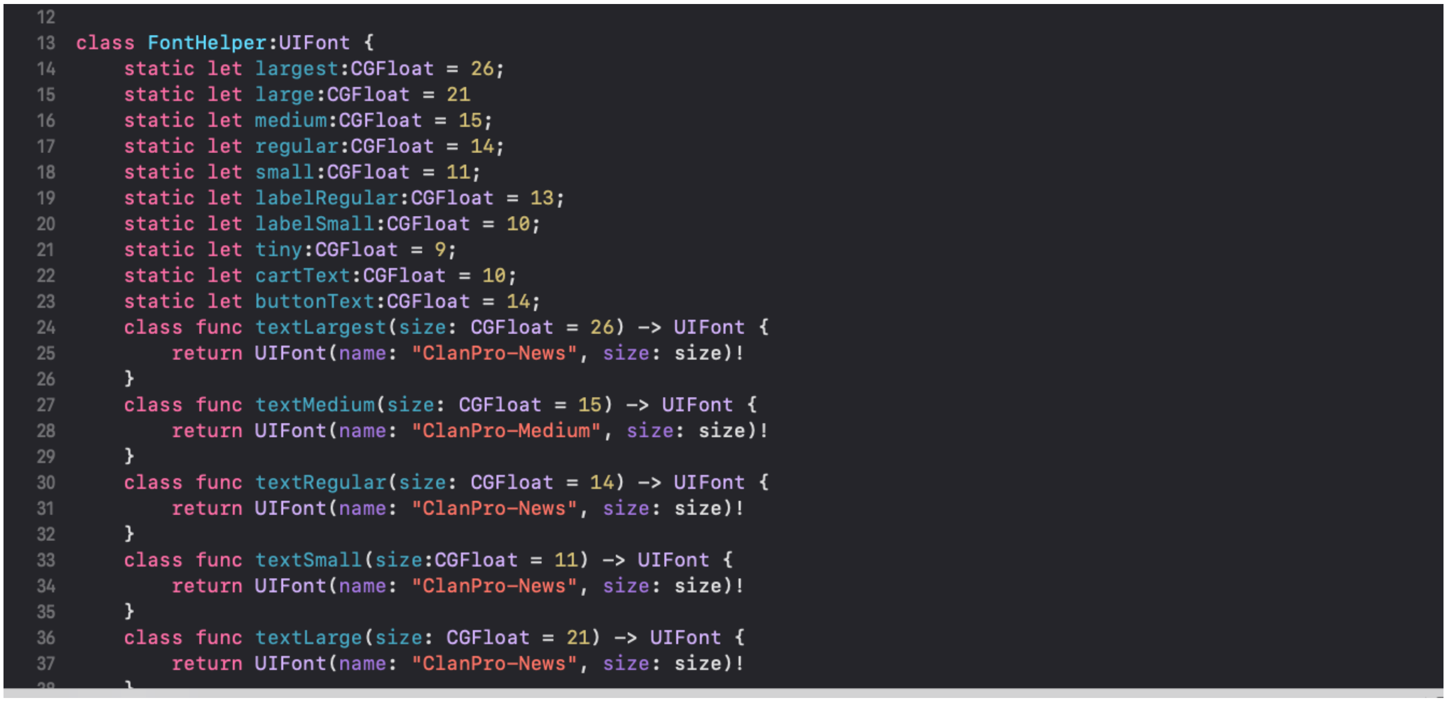
10.Change images

- Goto : Images-> Images Assets with .xcassets extension



11.Change font

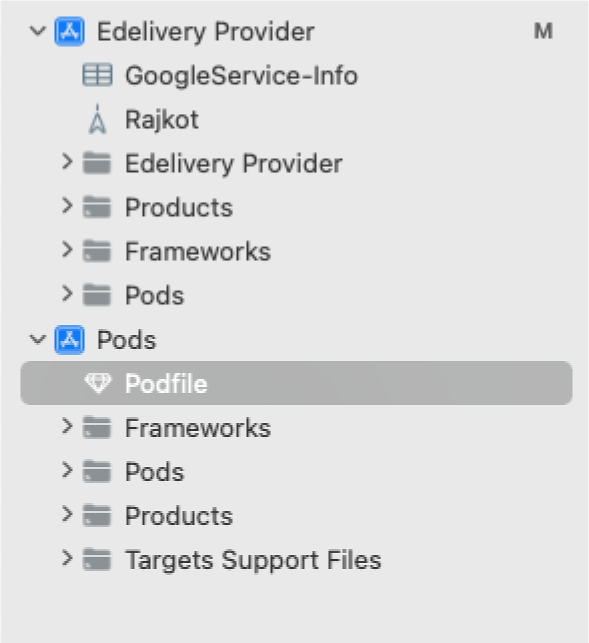
- Goto :HelperClass ->Utility and Extentions ->FontHelper -> Set Font Name at the place of name



```
class FontHelper:UIFont {
    static let largest:CGFloat = 26;
    static let large:CGFloat = 21
    static let medium:CGFloat = 15;
    static let regular:CGFloat = 14;
    static let small:CGFloat = 11;
    static let labelRegular:CGFloat = 13;
    static let labelSmall:CGFloat = 10;
    static let tiny:CGFloat = 9;
    static let cartText:CGFloat = 10;
    static let buttonText:CGFloat = 14;
    class func textLargest(size: CGFloat = 26) -> UIFont {
        return UIFont(name: "ClanPro-News", size: size)!
    }
    class func textMedium(size: CGFloat = 15) -> UIFont {
        return UIFont(name: "ClanPro-Medium", size: size)!
    }
    class func textRegular(size: CGFloat = 14) -> UIFont {
        return UIFont(name: "ClanPro-News", size: size)!
    }
    class func textSmall(size:CGFloat = 11) -> UIFont {
        return UIFont(name: "ClanPro-News", size: size)!
    }
    class func textLarge(size: CGFloat = 21) -> UIFont {
        return UIFont(name: "ClanPro-News", size: size)!
    }
```

10. Change name in pod file and install pod file (if required then)

Go to your project location (in finder) where you can see the Podfile and Podfile.lock



➔ Open Podfile in TextEdit and Edit with your requirements. (like app name changed, add and remove framework, etc..)

➔ After updating the podfile successfully, save it and open the terminal.

➔ In the terminal, go to the path where the pod file is located and write command “**pod Install**” into terminal and enter.

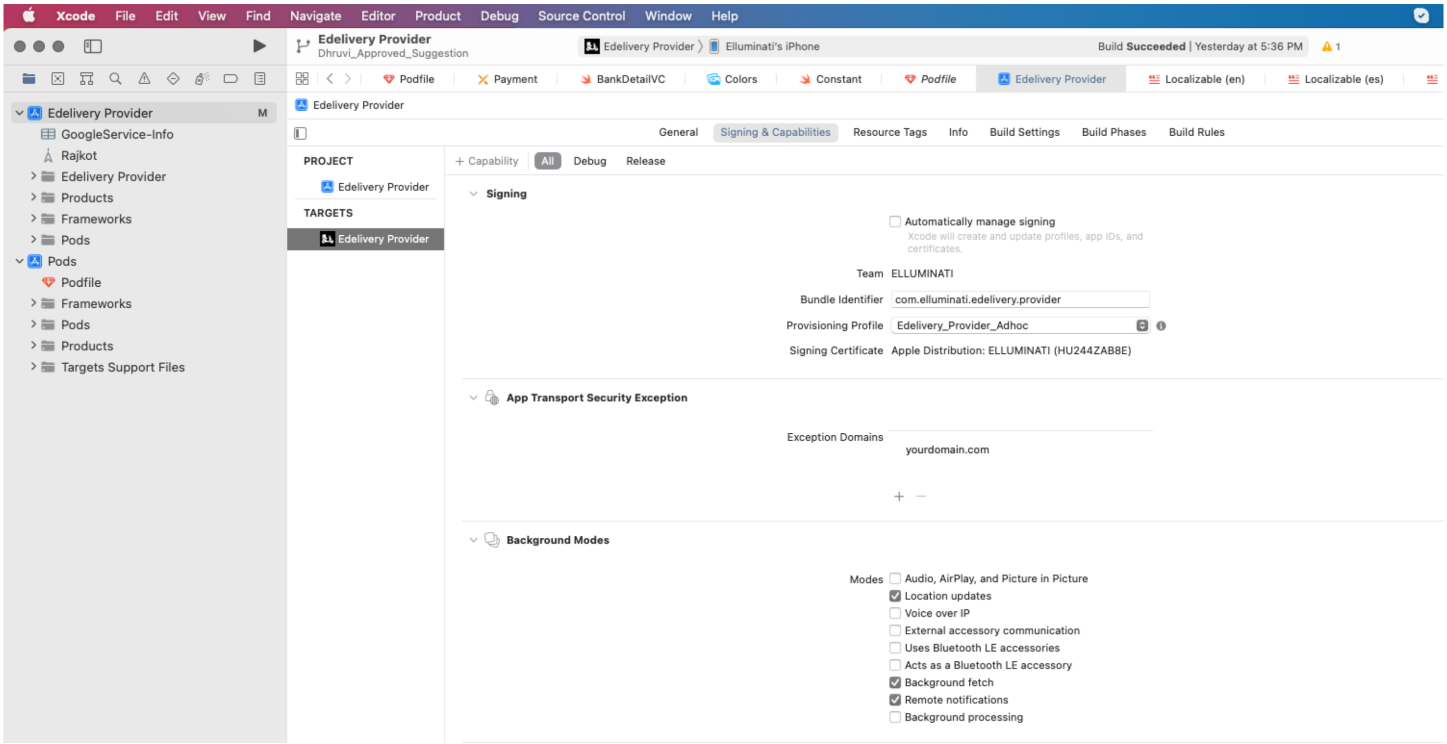
➔ it will create a new workspace of your app name and after you have to open that workspace.

➔ After installing the pod file and open new workspace, you can see 2 pods file framework in libraries so delete the old one.

★ Build project

Target Name : Edelivery Provider

- Select appropriate certificates and profiles like : development, adhoc, distribution
- Select Device
- Run the project



Signing

☐ Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.

Team ELLUMINATI

Bundle Identifier com.elluminati.edelivery.provider

Provisioning Profile Edelivery_Provider_Adhoc

Signing Certificate Apple Distribution: ELLUMINATI (HU244ZAB8E)

App Transport Security Exception

Exception Domains yourdomain.com

Background Modes

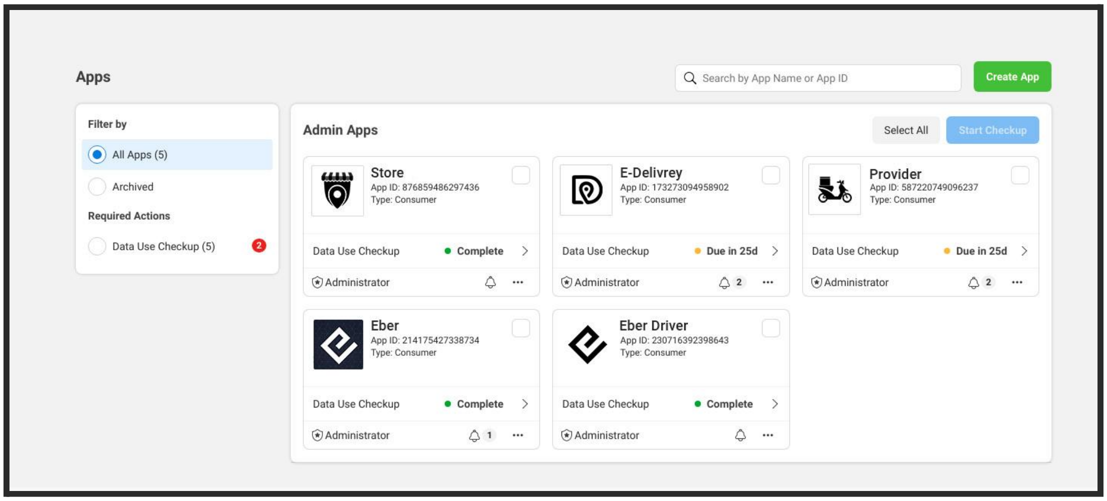
Modes
☐ Audio, AirPlay, and Picture in Picture
☑ Location updates
☐ Voice over IP
☐ External accessory communication
☐ Uses Bluetooth LE accessories
☐ Acts as a Bluetooth LE accessory
☑ Background fetch
☑ Remote notifications
☐ Background processing

Create App in Facebook Developer Account (for Facebok SignIn Feature)

For enabling facebook social login

Create facebook account after open facebook developer site https://developers.facebook.com/apps/

1. Create a New App



Store
App ID: 876859486297436
Type: Consumer
Data Use Checkup ● Complete
Administrator

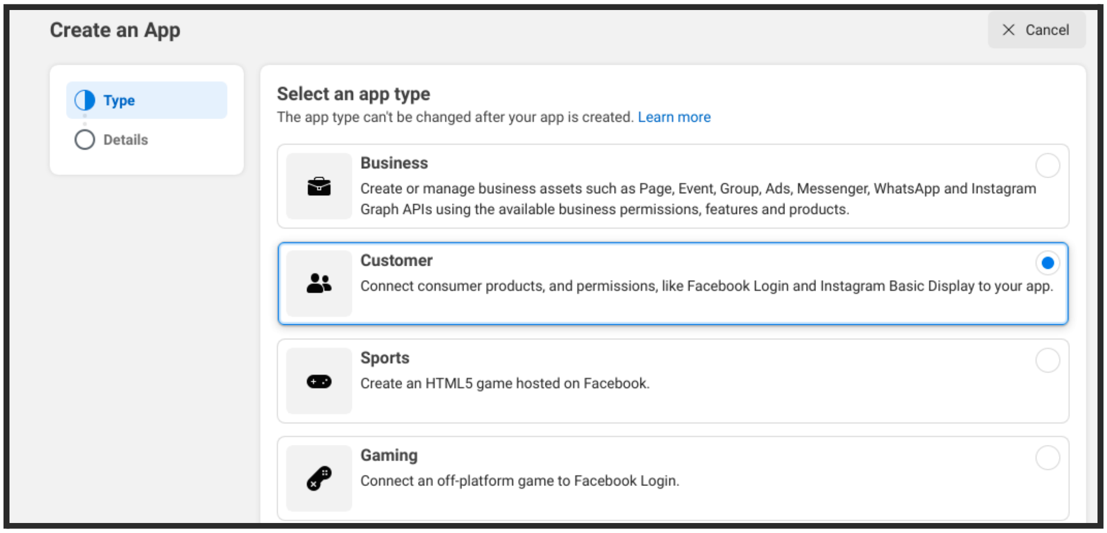
E-Delivrey
App ID: 173273094958902
Type: Consumer
Data Use Checkup ● Due in 25d
Administrator 2

Provider
App ID: 587220749096237
Type: Consumer
Data Use Checkup ● Due in 25d
Administrator 2

Eber
App ID: 214175427338734
Type: Consumer
Data Use Checkup ● Complete
Administrator 1

Eber Driver
App ID: 230716392398643
Type: Consumer
Data Use Checkup ● Complete
Administrator

2. Click on customer after click next



Select an app type

The app type can't be changed after your app is created. Learn more

○ Business
Create or manage business assets such as Page, Event, Group, Ads, Messenger, WhatsApp and Instagram Graph APIs using the available business permissions, features and products.

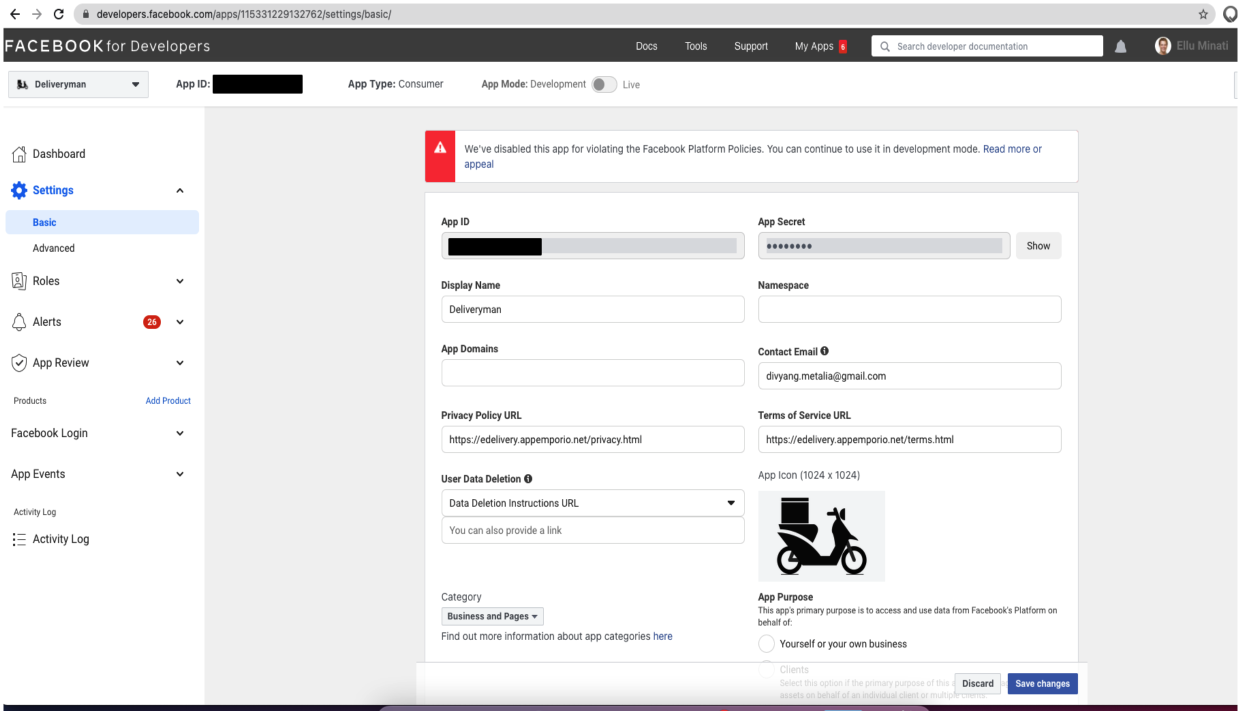
● Customer
Connect consumer products, and permissions, like Facebook Login and Instagram Basic Display to your app.

○ Sports
Create an HTML5 game hosted on Facebook.

○ Gaming
Connect an off-platform game to Facebook Login.

3. Open app -> Click on settings -> basic -> + Add Platform -> after select iOS
https://developers.facebook.com/docs/facebook-login/ios

4. Now you get one app id



developers.facebook.com/apps/115331229132762/settings/basic/

App ID: [redacted]
App Type: Consumer
App Mode: Development Live

We've disabled this app for violating the Facebook Platform Policies. You can continue to use it in development mode. Read more or appeal

App ID: [redacted]
App Secret: ••••••••
Display Name: Deliveryman
Namespace:
App Domains:
Contact Email: divyang.metalia@gmail.com
Privacy Policy URL: https://edelivery.appemporio.net/privacy.html
Terms of Service URL: https://edelivery.appemporio.net/terms.html
User Data Deletion: Data Deletion Instructions URL
You can also provide a link
App Icon (1024 x 1024)
Category: Business and Pages
Find out more information about app categories here
App Purpose
This app's primary purpose is to access and use data from Facebook's Platform on behalf of:
○ Yourself or your own business
○ Clients
Select this option if the primary purpose of this a... assets on behalf of an individual client or multiple clients.

After you get App ID, you first need to set this id in your project. For more details you can follow this tutorial : https://www.youtube.com/watch?v=P6uZ0o6xDA4

Configure the Info.plist file with an XML snippet that contains data about your app.

1. Right-click Info.plist, and choose Open As ‣ Source Code.
2. Copy and paste the following XML snippet into the body of your file (<dict>...</dict>).

```
<key>CFBundleURLTypes</key>
 <array>
```

```
<dict>
<key>CFBundleURLSchemes</key>
<array>
<string>fbAPP-ID</string>
</array>
  </dict>
</array>
<key>FacebookAppID</key>
<string>APP-ID</string>
<key>FacebookClientToken</key>
<string>CLIENT-TOKEN</string>
<key>FacebookDisplayName</key>
<string>APP-NAME</string>
```

1. In <array><string> in the key [CFBundleURLSchemes], replace *APP-ID* with your App ID.
2. In <string> in the key FacebookAppID, replace *APP-ID* with your App ID.
3. In <string> in the key FacebookClientToken, replace *CLIENT-TOKEN* with the value found under Settings > Advanced > Client Token in your App Dashboard.
4. In <string> in the key FacebookDisplayName, replace *APP-NAME* with the name of your app.
5. To use any of the Facebook dialogs (e.g., Login, Share, App Invites, etc.) that can perform an app switch to Facebook apps, your application's Info.plist also needs to include: <dict>...</dict>).

```
<key>LSApplicationQueriesSchemes</key>
<array>
  <string>fbapi</string>
```

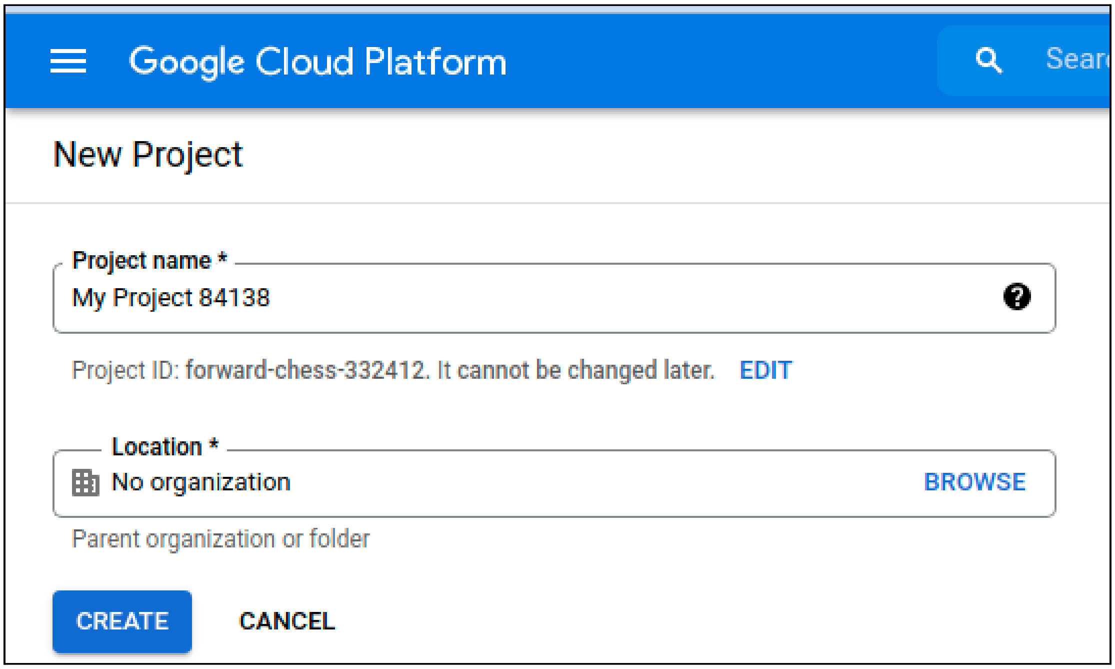
```
<string>fbapi20130214</string>
    <string>fbapi20130410</string>
    <string>fbapi20130702</string>
    <string>fbapi20131010</string>
    <string>fbapi20131219</string>
    <string>fbapi20140410</string>
    <string>fbapi20140116</string>
    <string>fbapi20150313</string>
```

```
<string>fbapi20150629</string>
<string>fbapi20160328</string>
<string>fbauth</string>
<string>fb-messenger-share-api</string>
<string>fbauth2</string>
<string>fbshareextension</string>
</array>
```

Create App in Google Developer Account

- For Using Google Apis (Google Map Api, Geocoding Api, Distance matrix Api etc) In our project we need to create project in google cloud console

1. Open the Google Cloud Console.
2. Next to "Google Cloud Platform," click the Down arrow . A dialog listing current projects appears.
3. Click **New Project**. The New Project screen appears.
4. In the **Project Name** field, enter a descriptive name for your project. If you're executing a quickstart, use "Quickstart."
5. Click **Organization** and select your organization.
6. In the **Location** field, click **Browse** to display potential locations for your project.
7. Click a location and click **Select**.
8. Click **Create**. The console navigates to the Dashboard page and your project is created within a few minutes.



New Project

Project name *
My Project 84138

Project ID: forward-chess-332412. It cannot be changed later.

Location *
No organization

Parent organization or folder

For further information on GCP projects, refer to Creating and managing projects.

- Activate Billing

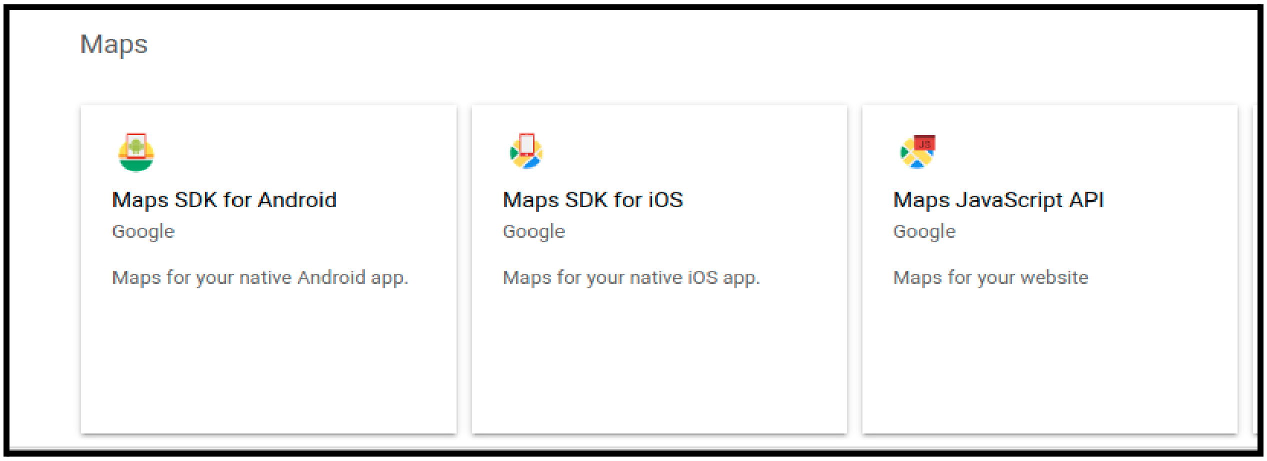
After successfully registering for a trial account you will be entitled to ~$300 free credits that you can spend within the Google Cloud Platform (GCP). However It would recommend to set up billing by adding a valid credit / debit card.

You can create a Billing Account here and its worthing remembering that one billing account can be used across multiple GCP projects.

- Enable a Google Workspace API

1. Open the Google Cloud Console.
2. Next to "Google Cloud Platform," click the Down arrow and select a project.
3. In the top-left corner, click Menu > **APIs & Services**.
4. Click **Enable APIs and Services**. The **Welcome to API Library** page appears.
5. In the search field, enter the name of the API you want to enable.
   For example, type "Map API" to find the Gmail API. If you are enabling an API for a quickstart, refer to the quickstart's Prerequisites section for the API to enable.
6. Click the API to enable. The API page appears.
7. Click **Enable**. The Overview page appears.
8. To enable an additional API, repeat steps 3 - 7.

For Example:

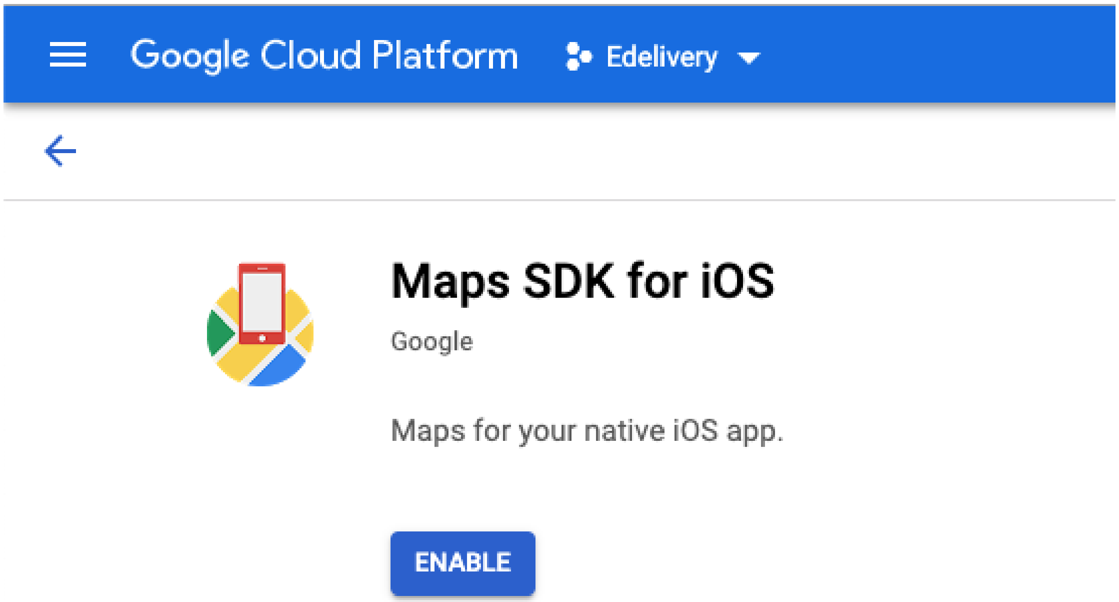


Maps

Maps SDK for Android
Google
Maps for your native Android app.

Maps SDK for iOS
Google
Maps for your native iOS app.

Maps JavaScript API
Google
Maps for your website



Maps SDK for iOS

Google

Maps for your native iOS app.

Make these libraries enable

- **Maps SDK for IOS**
  With the Maps SDK for IOS, add maps to your IOS app including Wear OS apps using Google Maps data, map displays, and map gesture responses.
  on web pages and mobile devices.Geolocation API

  For more detail :-
  https://developers.google.com/maps/documentation/ios-sdk/overview

- **Geocoding API**

  Geocoding is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers on a map, or position the map.
  The Geocoding API provides a direct way to access these services via an HTTP request.

  For more detail :-
  https://developers.google.com/maps/documentation/geocoding/overview

- **Distance Matrix API**

The Distance Matrix API is a service that provides travel distance and time for a matrix of origins and destinations.

For more detail :-
https://developers.google.com/maps/documentation/distance-matrix/overview

- ➔ **Directions API**
  Provide directions for multiple transportation modes, featuring real-time traffic information.

  For more detail :-
  https://developers.google.com/maps/documentation/directions

- ➔ **Places API**
  The Places API is a service that returns information about places using HTTP requests. Places are defined within this API as establishments, geographic locations, or prominent points of interest.

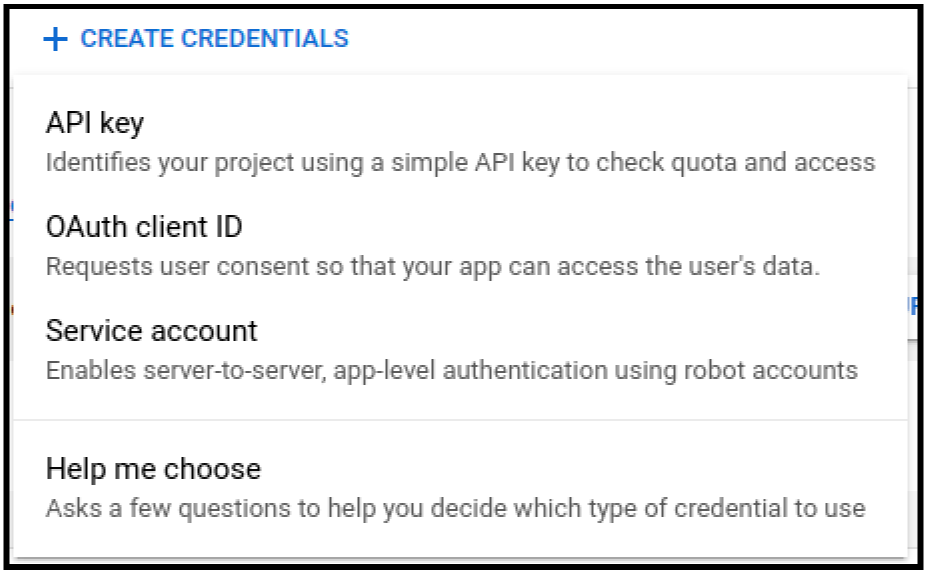
  For more detail :-
  https://developers.google.com/maps/documentation/places/web-service/overview

For more information on apis you can refer : https://developers.google.com/maps/documentation

Create Api key

Go to the Google Maps Platform > Credentials page.
Go to the Credentials page

- On the Credentials page, click Create credentials > API key.
  The API key created dialog displays your newly created API key.
- Click Close.
  The new API key is listed on the Credentials page under API keys.
  (Remember to restrict the API key before using it in production.)
-



CREATE CREDENTIALS

API key
Identifies your project using a simple API key to check quota and access

OAuth client ID
Requests user consent so that your app can access the user's data.

Service account
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use

After paste this key in project constant file

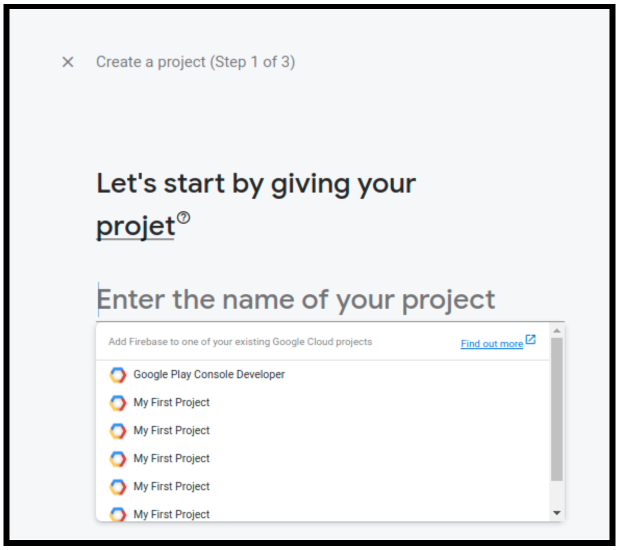
★ Firebase Account

Firebase provides many utilities like cloud messaging, Crashalytics ,Analytics , RealTime Databases ,In-App Messaging , Dynamic Links etc.
You can learn more about firebase products from https://firebase.google.com/

- Create Project in FirebaseConsole

- ➔ login into google account
- ➔ Goto Firebase console https://console.firebase.google.com/u/4/

- ➔ Click Add Project
- ➔ You can see Google cloud projects you created in https://console.cloud.google.com here , Select Your Project and continue

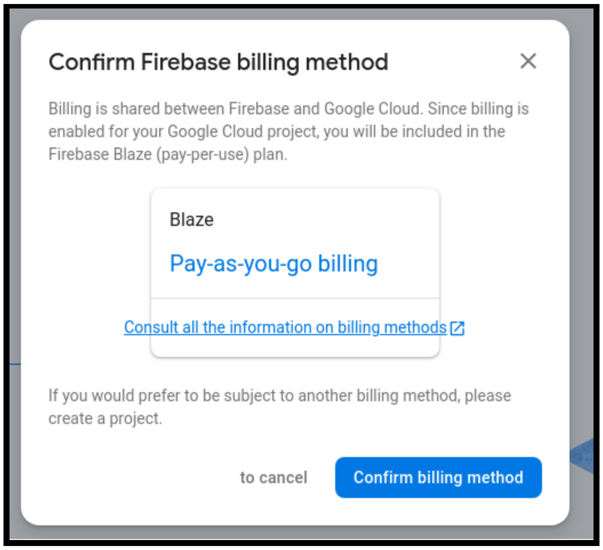


Create a project (Step 1 of 3)

Let's start by giving your projet

Enter the name of your project

Add Firebase to one of your existing Google Cloud projects



Confirm Firebase billing method

Billing is shared between Firebase and Google Cloud. Since billing is enabled for your Google Cloud project, you will be included in the Firebase Blaze (pay-per-use) plan.

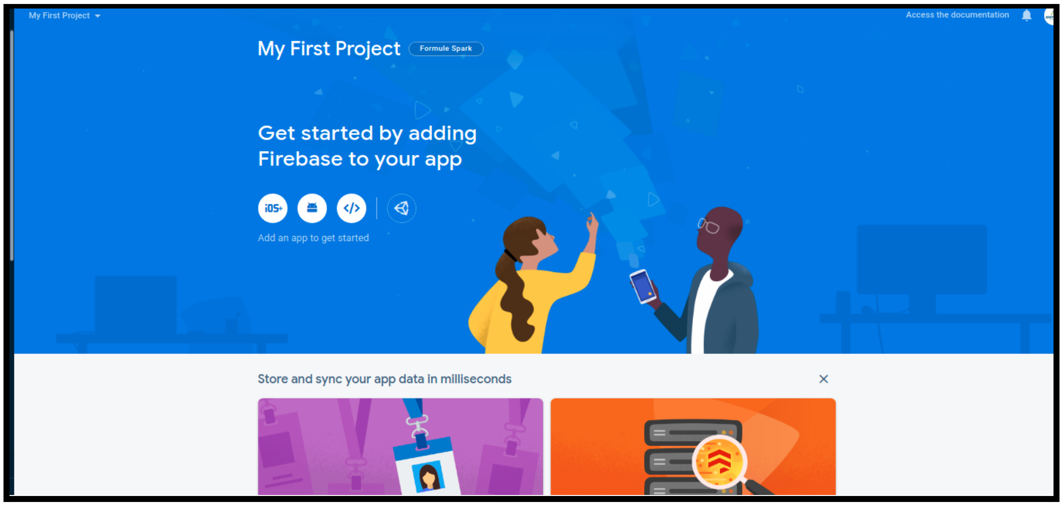
Blaze

Pay-as-you-go billing

Consult all the information on billing methods

If you would prefer to be subject to another billing method, please create a project.

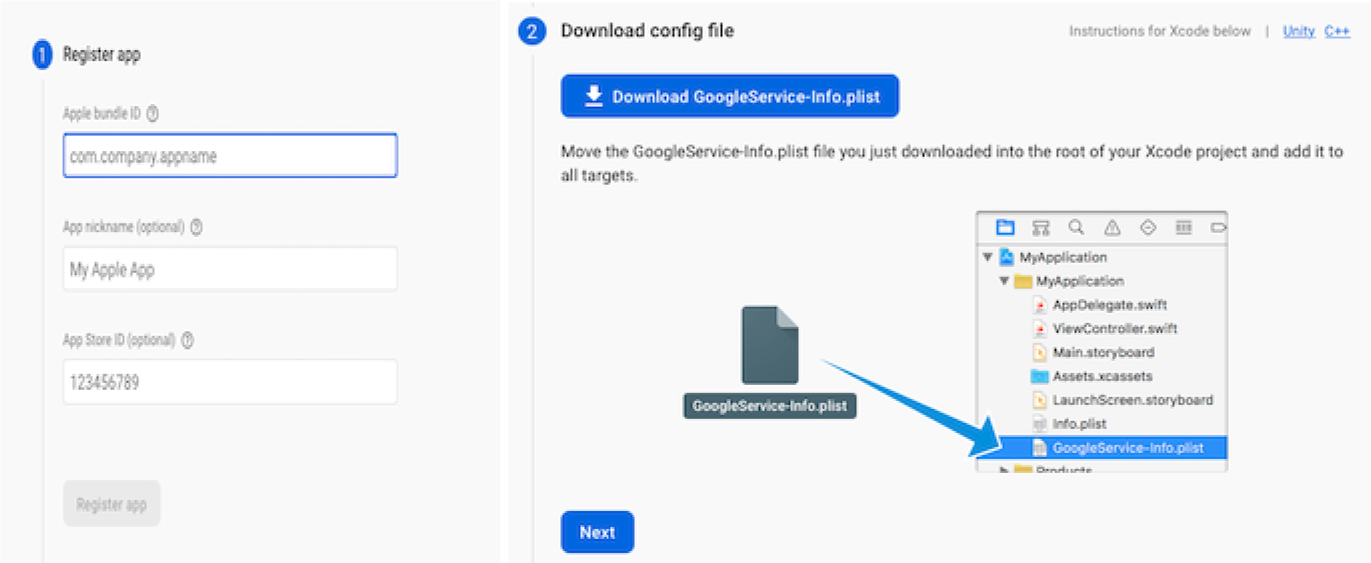
- ➔ Unselect switch. You can set it up later .Continue to create project
- ➔ Confirm Your Billing Method
- ➔ In the next steps, you will be asked whether to set up Google Analytics.



My First Project
Access the documentation
My First Project
Formule Spark
Get started by adding Firebase to your app
iOS+
Add an app to get started
Store and sync your app data in milliseconds

Create iOS App

- ➔ Now the Project is created. Add iOS app by clicking on iOS icon
- ➔ Add your apps package name and App Name
- ➔ Register your app and download **GoogleService-Info.plist** file



1 Register app

Apple bundle ID: com.company.appname

App nickname (optional): My Apple App

App Store ID (optional): 123456789

Register app

2 Download config file

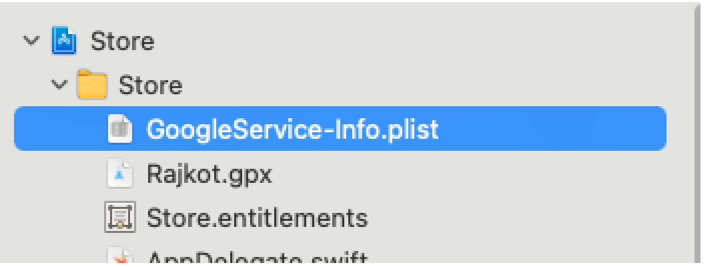
Instructions for Xcode below | Unity C++

Download GoogleService-Info.plist

Move the GoogleService-Info.plist file you just downloaded into the root of your Xcode project and add it to all targets.

Next

➔ Add this **GoogleService-Info.plis** file to the module root directory of your iOS app. Refer below image.(This Downloaded GoogleService-Info.plist File put on your project Folder)



➔ Change CLIENT ID and REVERSED CLIENT ID in Info.plist file (which is given in **GoogleService-Info.plist file**)

➔ **Path:-** Select Project —> Edelivery / Edelivery Provider / Store Folder —> Info.plist file open with source code

➔ Go to Constant file —> Constants.swift —> static var CLIENT_ID = “**GIVE YOUR CLIENT ID**”

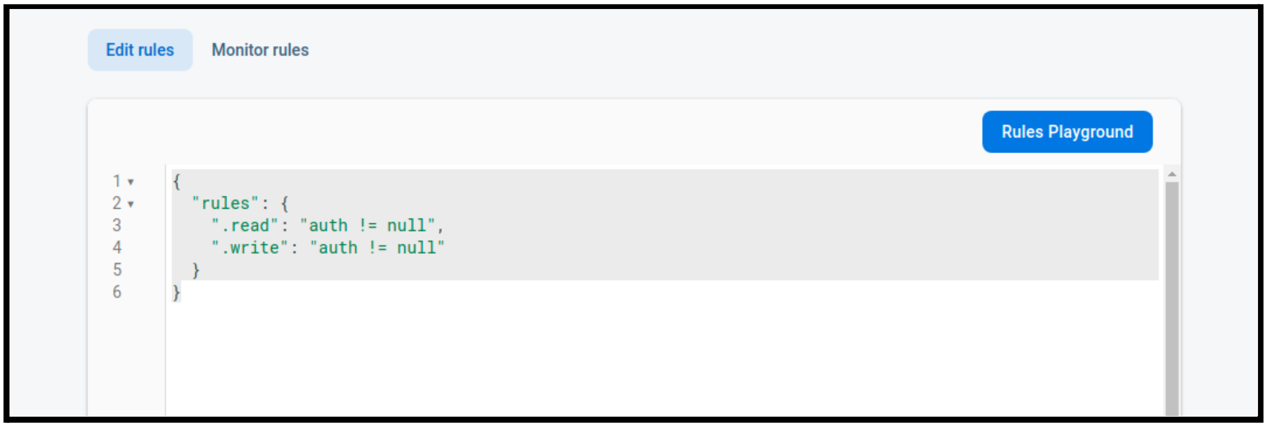
➔ Replace your CLIENT ID at above place

★ Create RealTimeDatabase

Store and sync data with our NoSQL cloud database. Data is synced across all clients in realtime, and remains available when your app goes offline.Firebase Realtime Database Security Rules determine who has read and write access to your database, how your data is structured, and what indexes exist.

For more info check this https://firebase.google.com/docs/database
We are using Firebase realTimeDatabase for sending,retrieving ,storing chat data

- ➔ GoTo Firebase console
- ➔ Side menu -> Realtime Database -> Create Database ->select locked mode -click
- ➔ rules -> read : auth != null and write : auth != null -> click on publish



```
{
  "rules": {
    ".read": "auth != null",
    ".write": "auth != null"
  }
}
```