Goferjek Driver App iOS Configuration

Trioangle Technologies Pvt Ltd
Madurai, Tamil Nadu, India.
www.trioangle.com
support@trioangle.com
ⓈⓉrioangle

Table of Contents

Table of Contents 2

Open Project In Xcode 3

Steps To Open Project In Xcode 3

Reference Below 3

Updating Google Map SDK API Key 4

Reference Below 4

Paste API key in Xcode Project 6

Updating Appname 6

Reference Below 6

Reference Below 8

Updating APP Icon 8

Steps For Updating App Icon 8

Reference Below 9

Updating Live URL 10

Reference Below 11

Reference Below 12

Steps To update FCM Key 14

Paste API key in Xcode Project 14

Reference Below 14

Enabling Crash Analytics 15

Reference Below 15

API Config document 16

Braintree configuration 16

Steps To update URL From Bundle Identifier 16

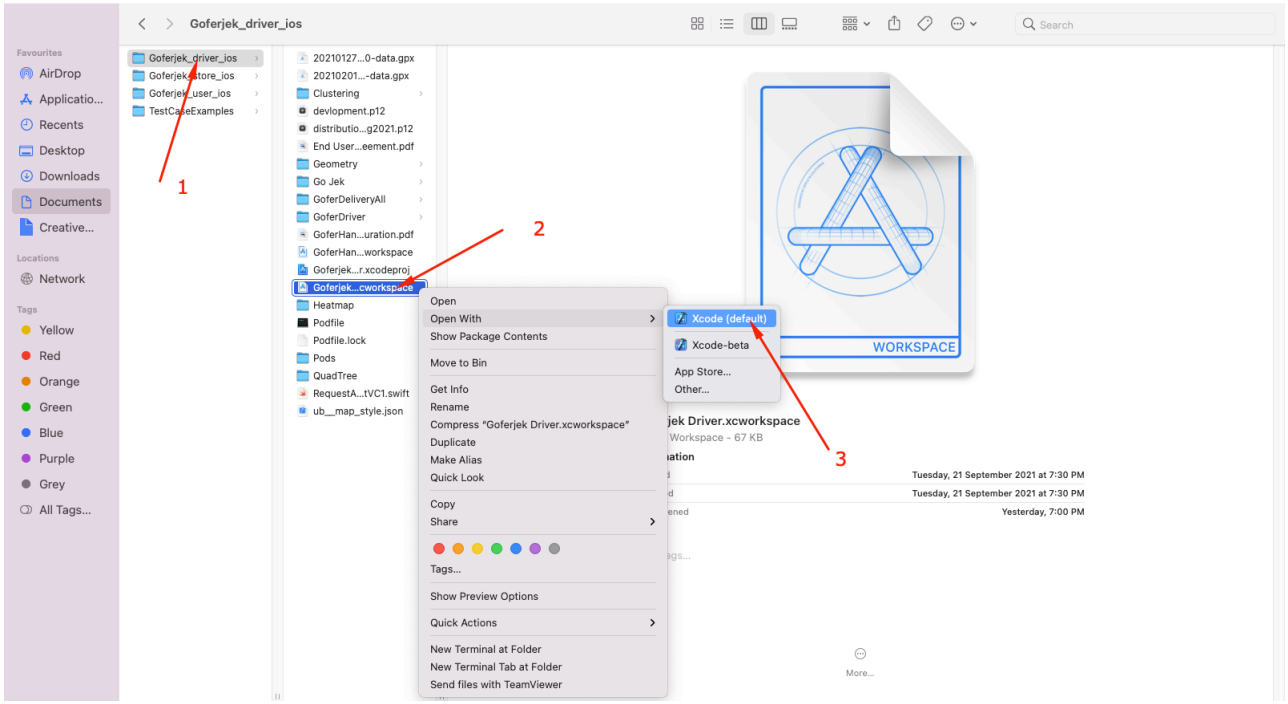
Steps To update Itunes Data 18

Open Project In Xcode

Steps To Open Project In Xcode

1. Open Goferjek Provider Folder
2. Right Click “Goferjek Driver.xcworkspace” file
3. Select Open with
4. Choose Xcode

Reference Below



Updating Google Map SDK API Key

- Get **Google Map API key** and use it to authenticate various types of Web **Application**, **Mobile Application**.

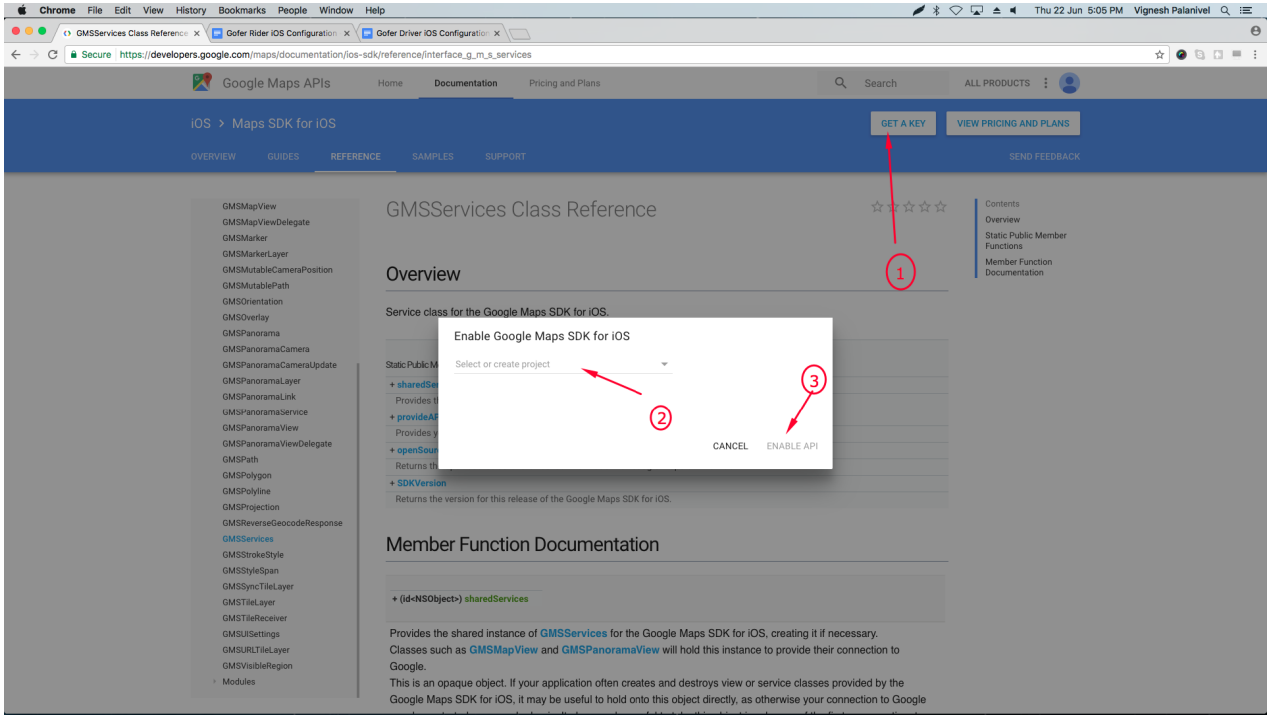
Reference Below

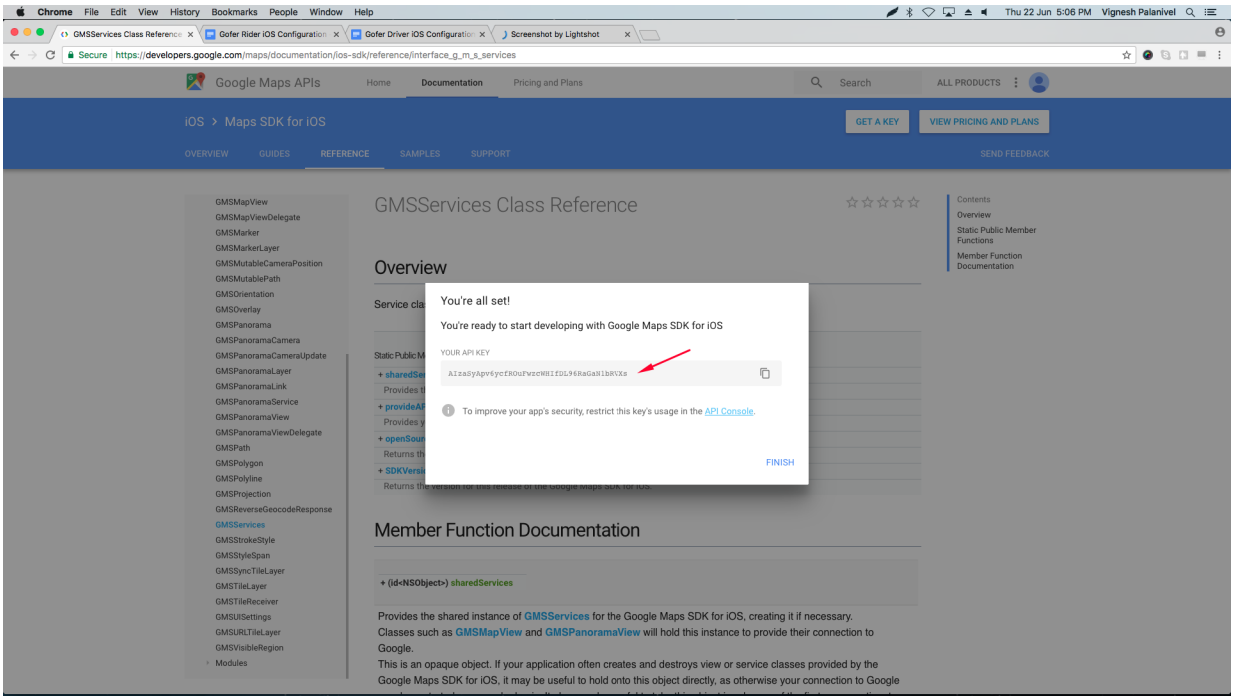
Getting Google Map SDK key
https://developers.google.com/maps/documentation/ios-sdk/reference/interface_g_m_s_services

Steps To Getting Google Map SDK key

1. Click GET AKEY
2. Type / Select App Name
3. Click Enable API
4. Copy API key

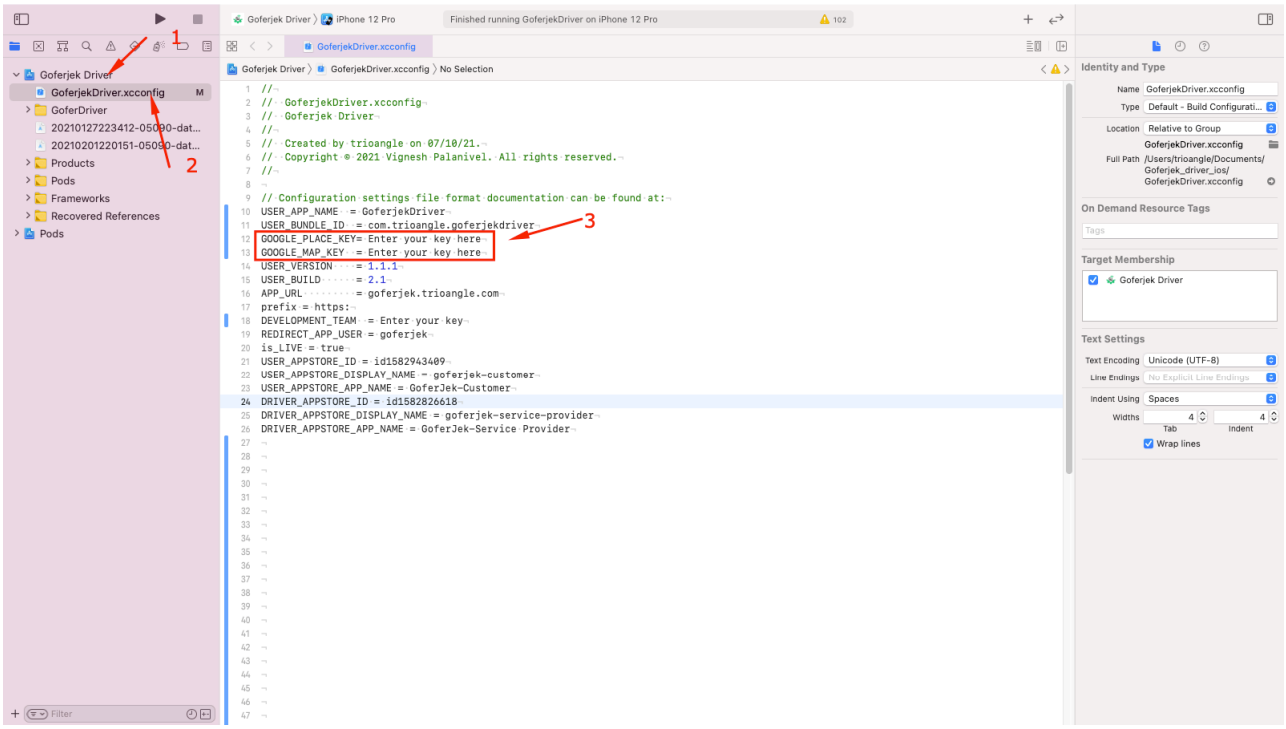
Paste API key in Xcode Project





Steps To Paste API key in Xcode Project

1. Click file folder in xcode and Click “Goferjek Driver”
2. Open GoferjekDriver.xcconfig file
3. And, paste Your Google Key in “GOOGLE_PLACE_KEY” and “GOOGLE_MAP_KEY”(Live and demo)

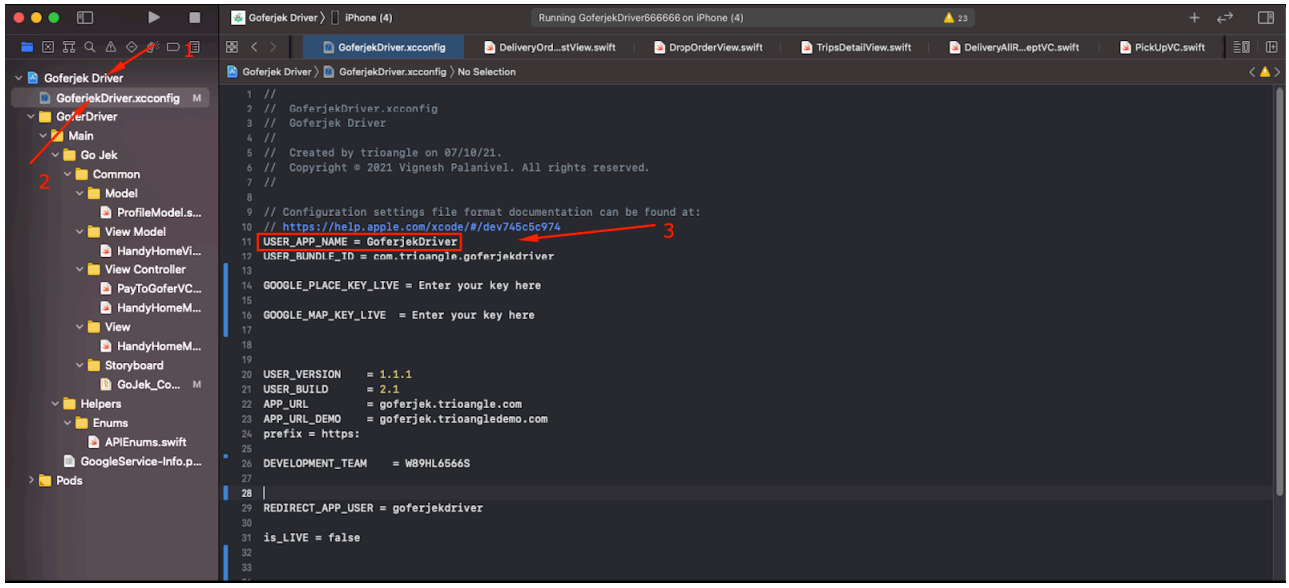


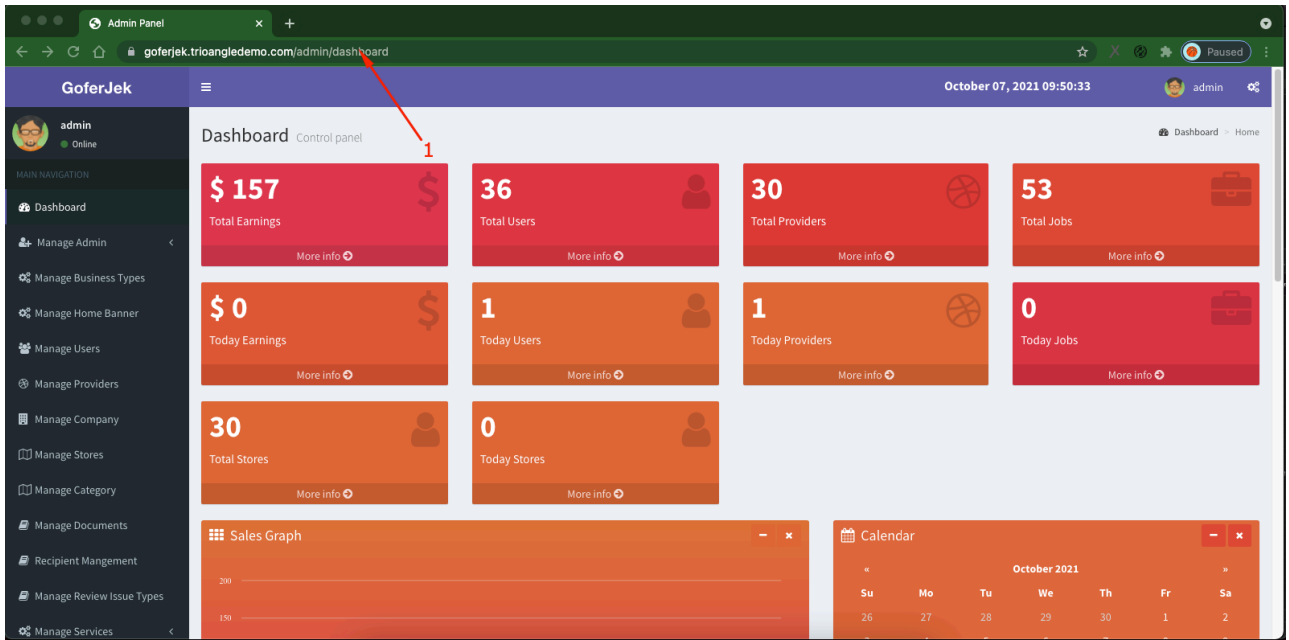
Updating App name

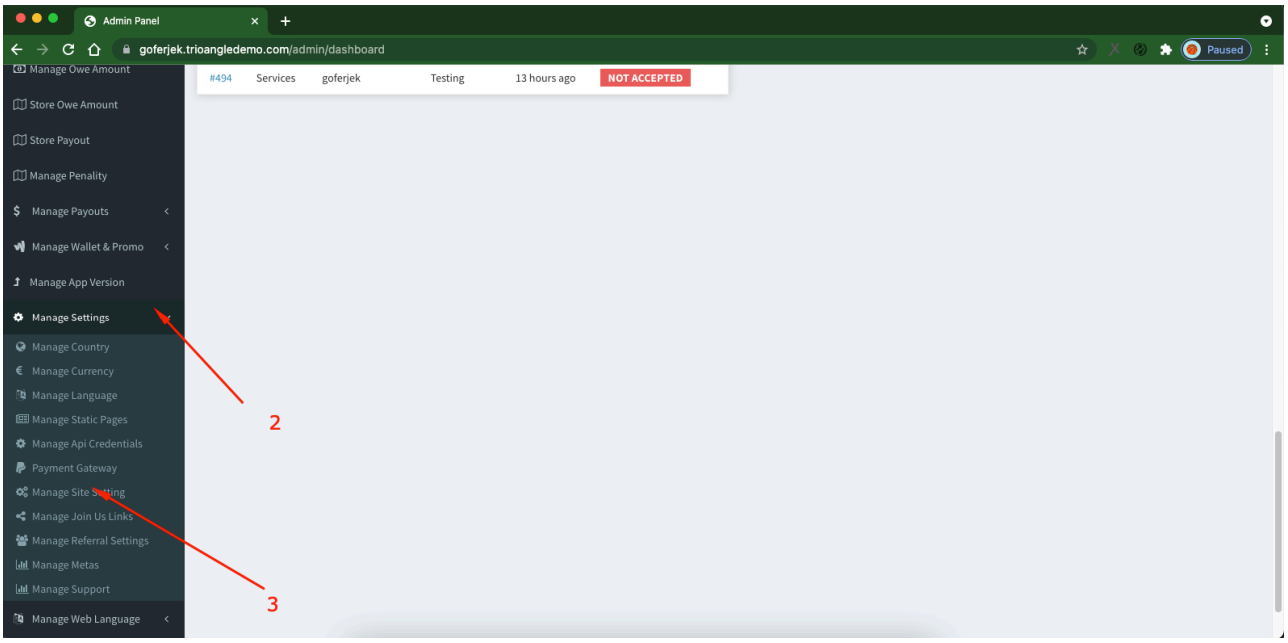
Steps For Updating App name

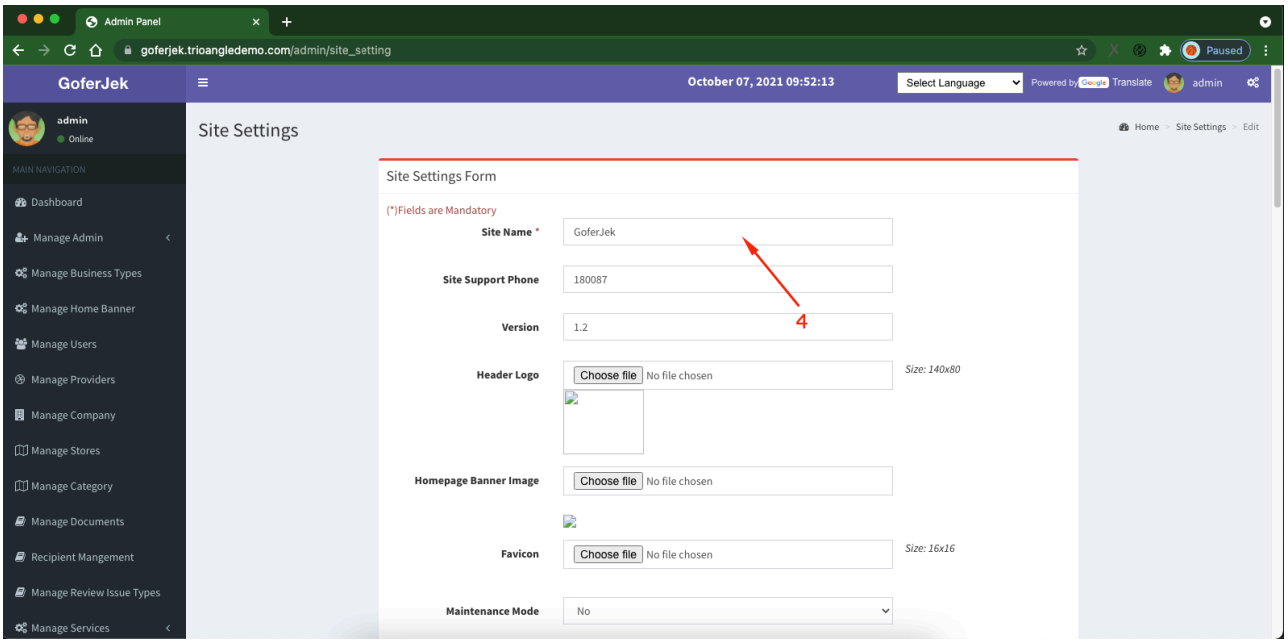
1. Click Goferjek Driver Project
2. Open GoferjekDriver.xcconfig file
3. Change display name in “USER_APP_NAME” key
4. Change app name in admin panel → Click Manage Site Settings → Click Site Settings → Enter Your App Name in Site Name field.

Reference Below


Goferjek Driver
GoferjekDriver.xcconfig
GoferDriver
Main
Go Jek
Common
Model
View Model
View Controller
View
Storyboard
Helpers
Enums
APIEnums.swift
GoogleService-Info.p...
Pods
1
2
3
// GoferjekDriver.xcconfig
// Goferjek Driver
// Created by trioangle on 07/10/21.
// Copyright © 2021 Vignesh Palanivel. All rights reserved.
// Configuration settings file format documentation can be found at:
// https://help.apple.com/xcode/#/dev745c5c974
USER_APP_NAME = GoferjekDriver
USER_BUNDLE_ID = com.trioangle.goferjekdriver
GOOGLE_PLACE_KEY_LIVE = Enter your key here
GOOGLE_MAP_KEY_LIVE = Enter your key here
USER_VERSION = 1.1.1
USER_BUILD = 2.1
APP_URL = goferjek.trioangle.com
APP_URL_DEMO = goferjek.trioangledemo.com
prefix = https:
DEVELOPMENT_TEAM = W89HL6566S
REDIRECT_APP_USER = goferjekdriver
is_LIVE = false


Admin Panel
goferjek.trioangledemo.com/admin/dashboard
GoferJek
admin
Online
MAIN NAVIGATION
Dashboard
Manage Admin
Manage Business Types
Manage Home Banner
Manage Users
Manage Providers
Manage Company
Manage Stores
Manage Category
Manage Documents
Recipient Mangement
Manage Review Issue Types
Manage Services
October 07, 2021 09:50:33
admin
Dashboard Control panel
Dashboard Home
1
$ 157
Total Earnings
More info
36
Total Users
30
Total Providers
53
Total Jobs
$ 0
Today Earnings
1
Today Users
1
Today Providers
0
Today Jobs
30
Total Stores
0
Today Stores
Sales Graph
Calendar
October 2021
Su Mo Tu We Th Fr Sa

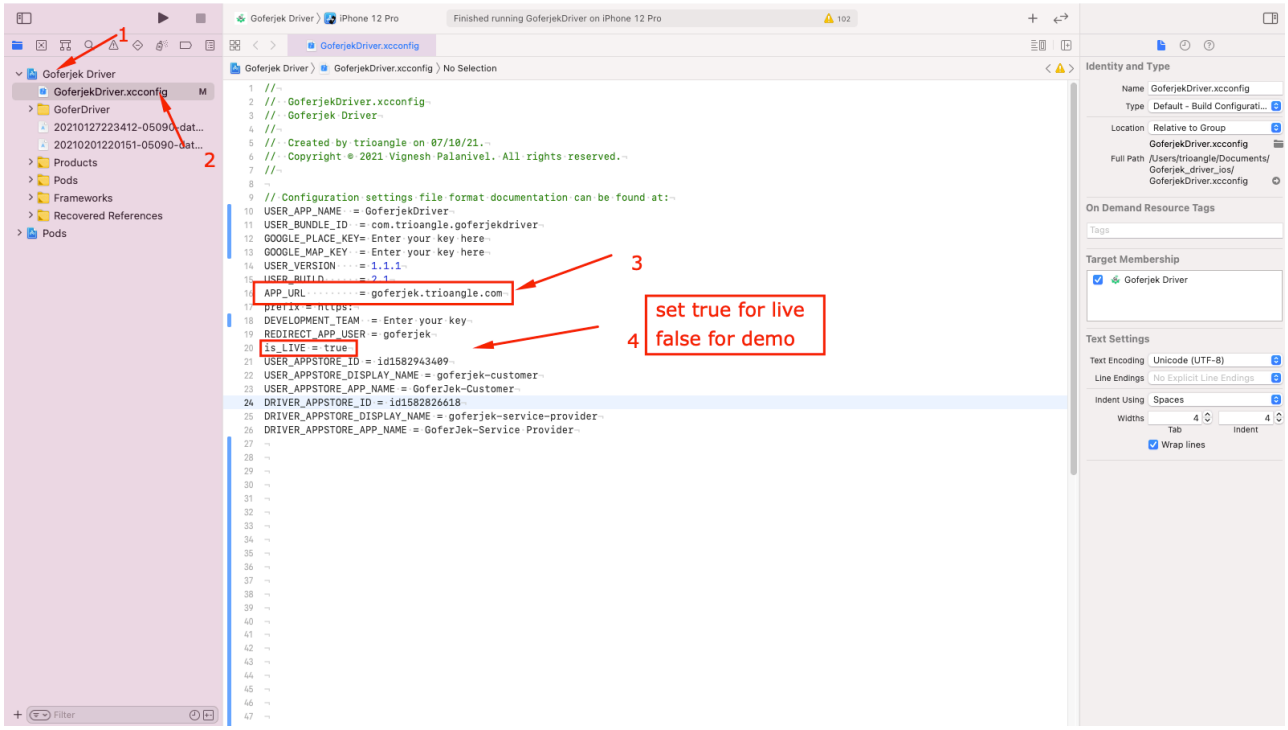




Updating Live URL

Steps To Update Live URL

1. Click file folder in xcode and Click “Goferjek Driver”
2. Open GoferjekDriver.xcconfig file
3. And, Change Your Live Url in “APP_URL” key
4. Finally Set “is_LIVE” value true.



Updating APP Logo

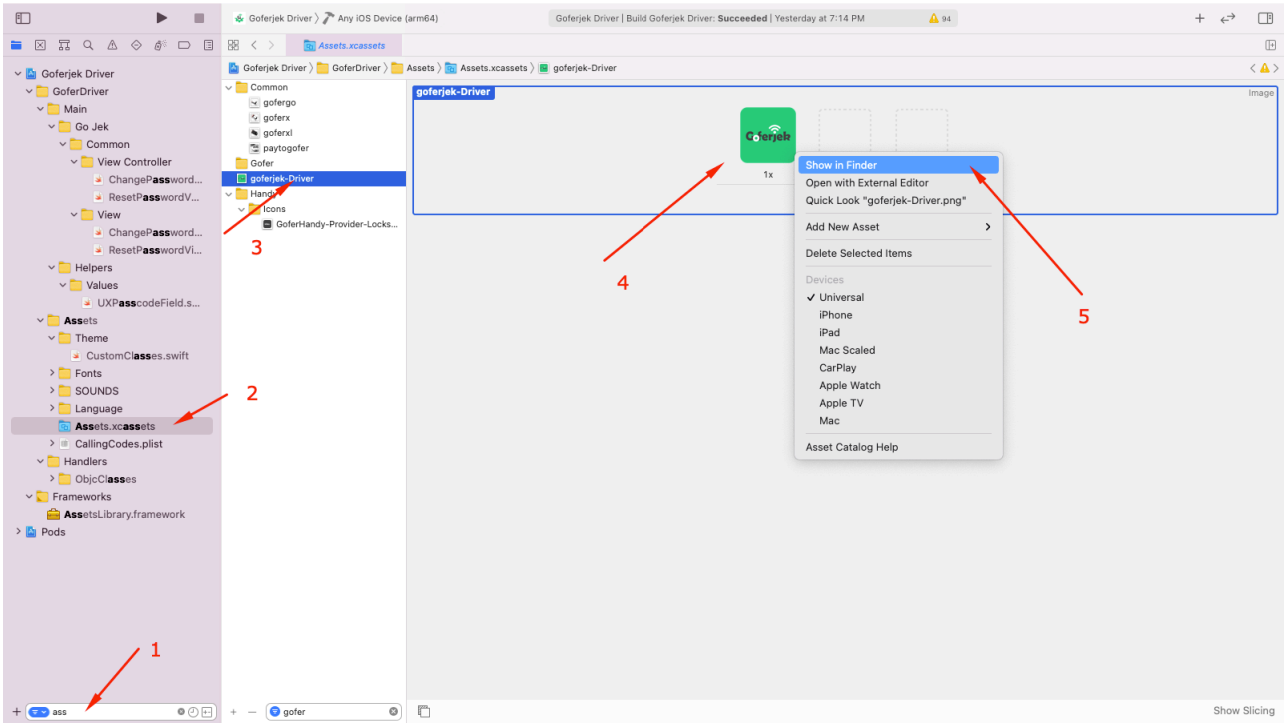
- Modifying App Logo

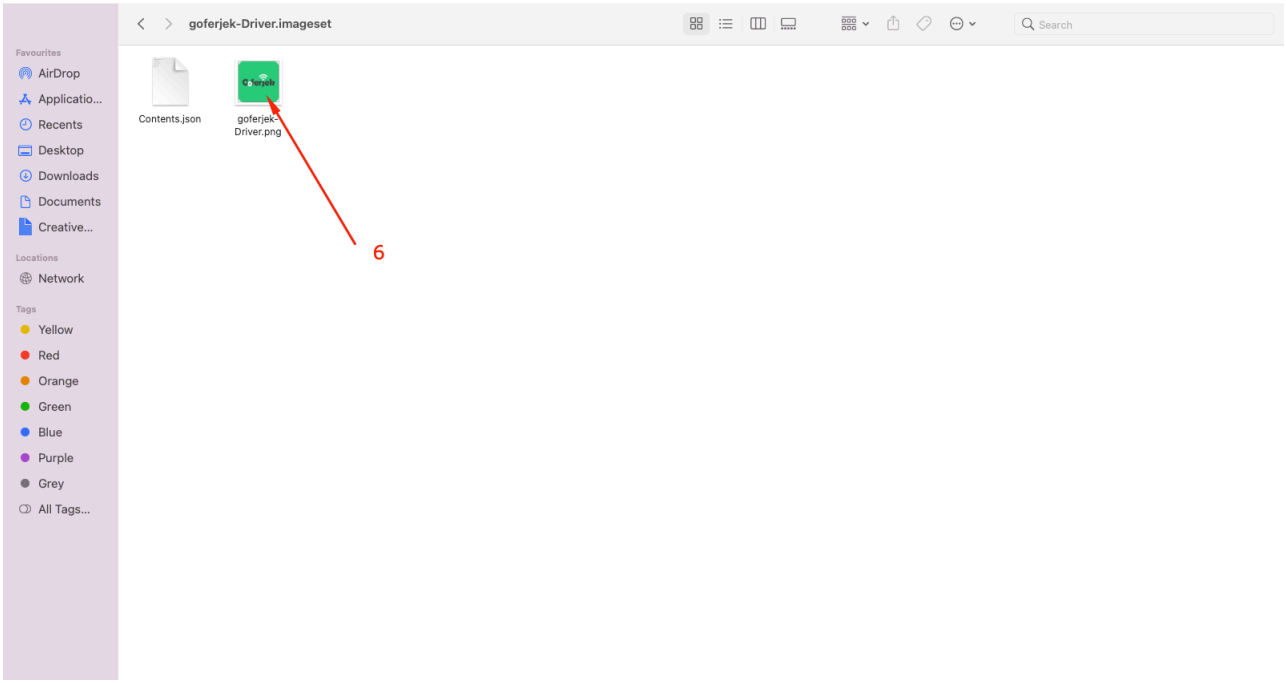
  [Note: Do not use transparent png image for logo]

1. Search for assets
2. Select Assets.xcassets
3. In Assets search for goferjek Driver
4. Right click the goferjek Driver.png and Click “Show in finder”
5. Open the folder
6. Replace the image

(Note:Don’t Change the Image Name)

Reference Below


Goferjek Driver
GoferDriver
Main
Go Jek
Common
View Controller
ChangePassword...
ResetPasswordV...
View
ChangePassword...
ResetPasswordVi...
Helpers
Values
UXPasscodeField.s...
Assets
Theme
CustomClasses.swift
Fonts
SOUNDS
Language
Assets.xcassets
CallingCodes.plist
Handlers
ObjcClasses
Frameworks
AssetsLibrary.framework
Pods
Assets.xcassets
Common
gofergo
goferx
goferxl
paytogofer
Gofer
goferjek-Driver
Icons
GoferHandy-Provider-Locks...
goferjek-Driver
Image
1x
Show in Finder
Open with External Editor
Quick Look "goferjek-Driver.png"
Add New Asset
Delete Selected Items
Devices
Universal
iPhone
iPad
Mac Scaled
CarPlay
Apple Watch
Apple TV
Mac
Asset Catalog Help
Show Slicing
1
2
3
4
5


goferjek-Driver.imageset
Search
Favourites
AirDrop
Applicatio...
Recents
Desktop
Downloads
Documents
Creative...
Locations
Network
Tags
Yellow
Red
Orange
Green
Blue
Purple
Grey
All Tags...
Contents.json
goferjek-Driver.png
6

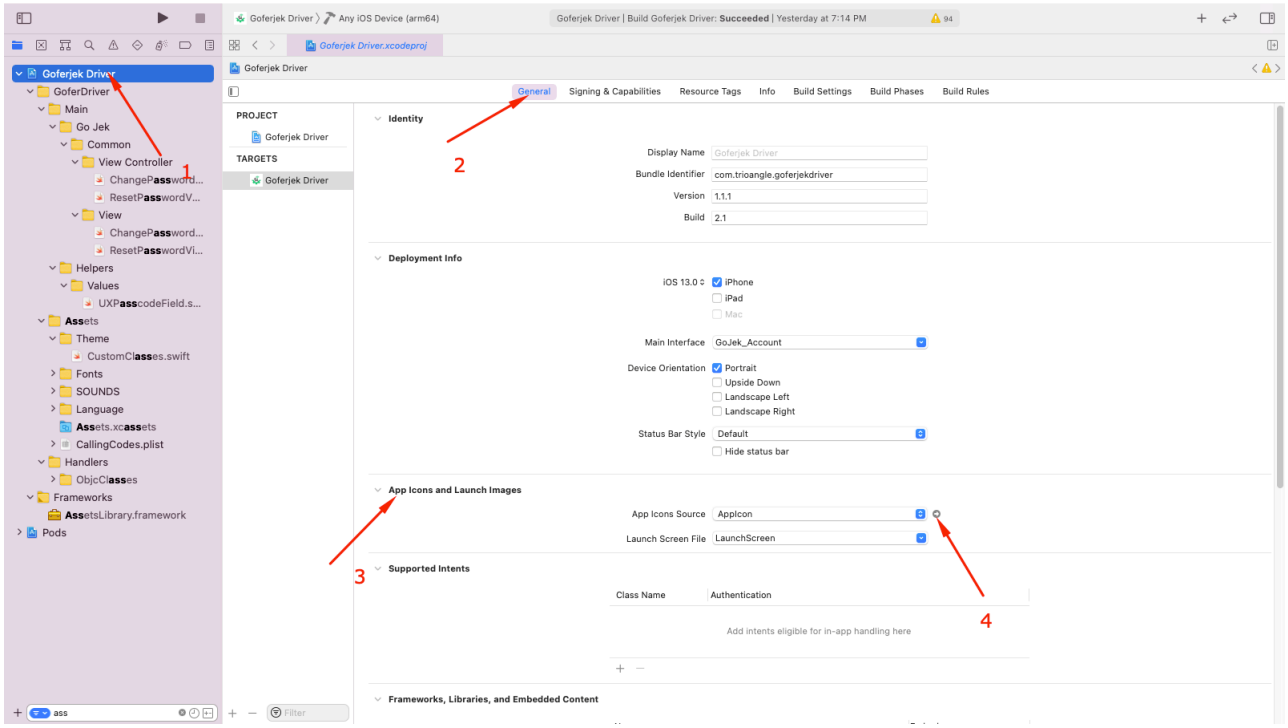
Updating APP Icon

- Updating App Icon.

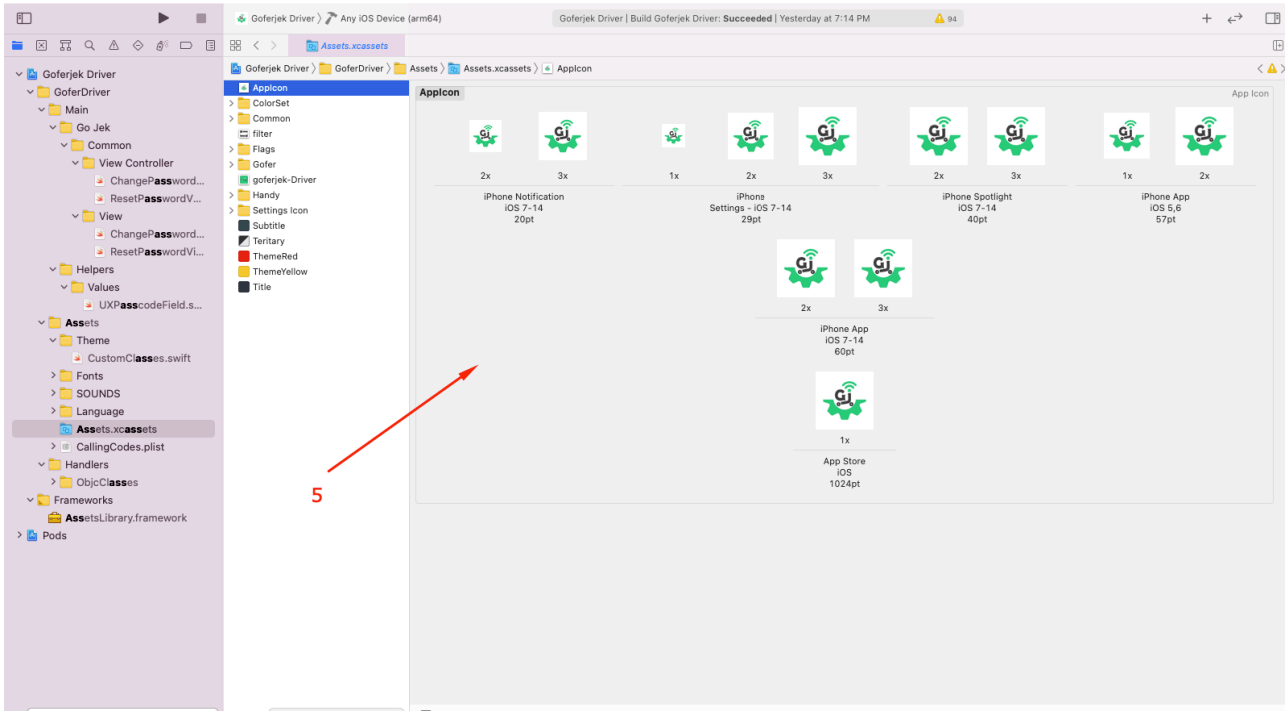
Steps For Updating App Icon

1. Click Goferjek Driver Project
2. Click general tap
3. Below You can see app icons and launch
4. Click arrow icon near app icon source

Reference Below



5. Change/Drag and drops All image according to image size
    a. To change the app Icon in selected folder
    b. Use to get all size app icon, follow the link https://makeappicon.com/



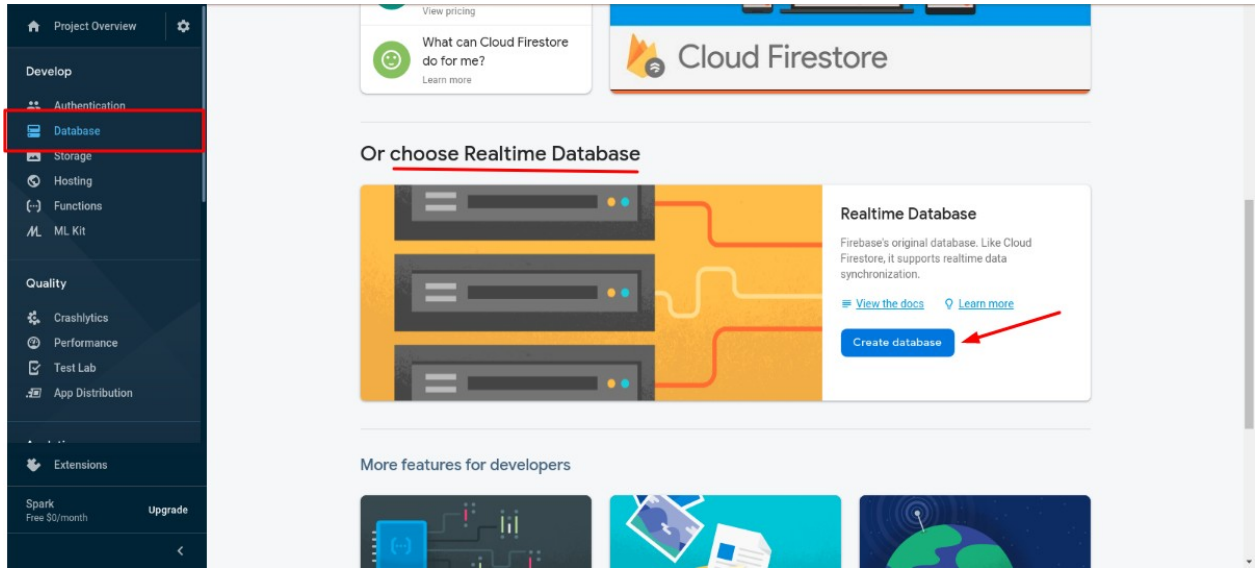
Updating FCM Key (Firebase Cloud Messaging)

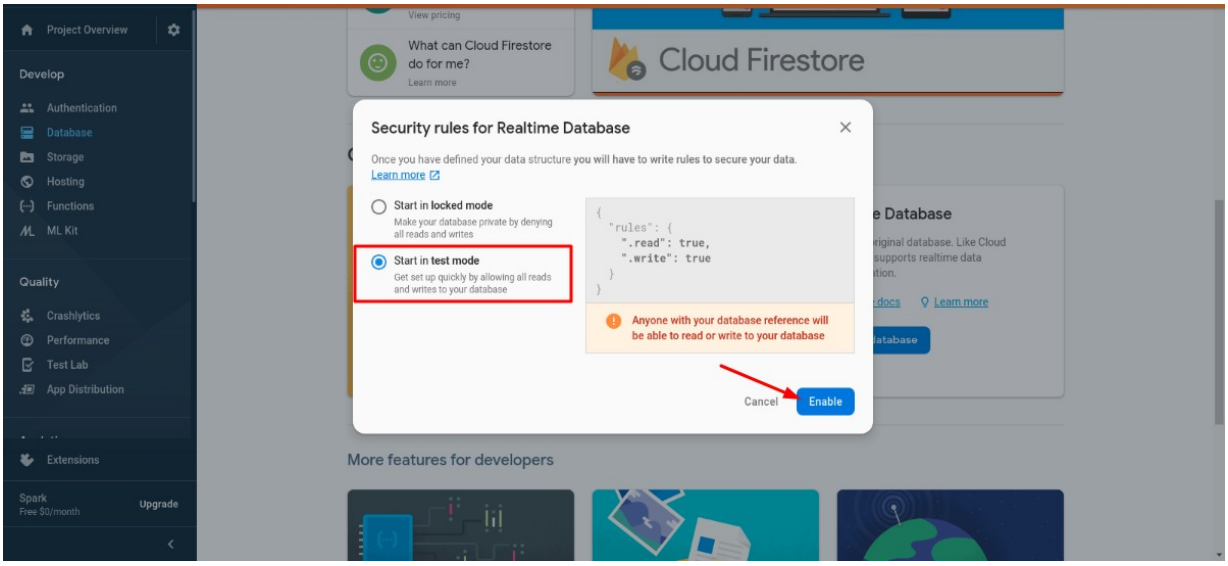
- Get **FCM Key** used to push notification into different types of Web **Application**, **Mobile Application**.
- Follow the link to create FCM https://console.firebase.google.com
- Create the project and create both provider and user app
- Click setting in selected app
- Download the **GoogleService-Info.plist** from firebase console

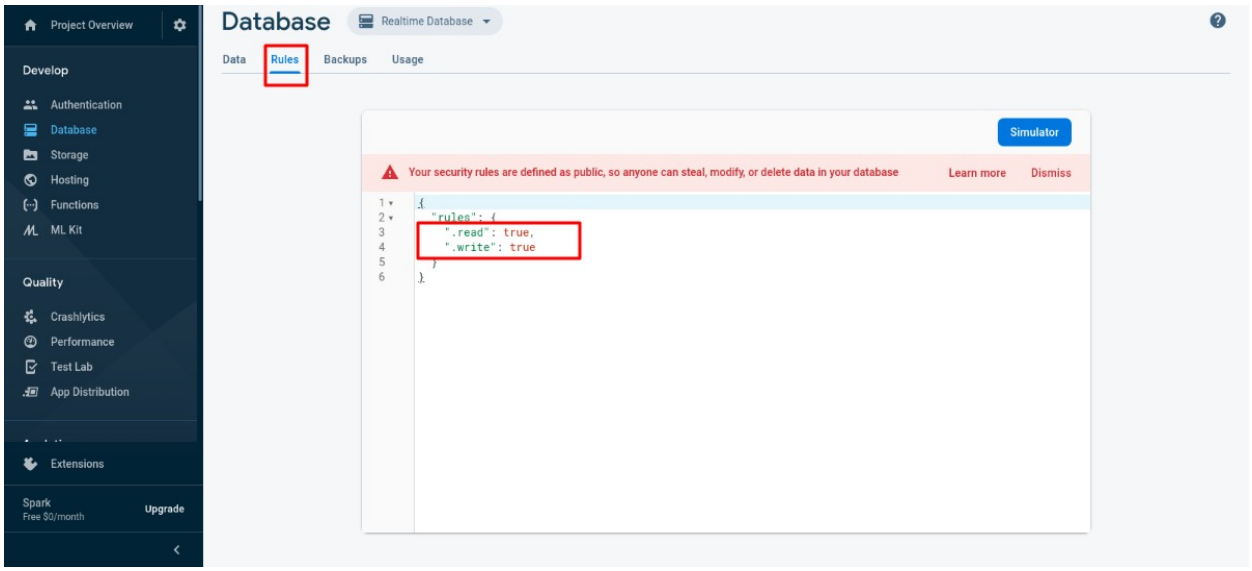
Note:

Use same FCM key in web,android and iOS also.

Configuring Firebase console (Firebase Cloud Messaging)


Project Overview
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics
Performance
Test Lab
App Distribution
Extensions
Spark
Free $0/month
Upgrade
What can Cloud Firestore do for me?
Learn more
Cloud Firestore
Or choose Realtime Database
Realtime Database
Firebase's original database. Like Cloud Firestore, it supports realtime data synchronization.
View the docs
Learn more
Create database
More features for developers


Project Overview
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics
Performance
Test Lab
App Distribution
Extensions
Spark
Free $0/month
Upgrade
What can Cloud Firestore do for me?
Learn more
Cloud Firestore
Security rules for Realtime Database
Once you have defined your data structure you will have to write rules to secure your data.
Learn more
Start in locked mode
Make your database private by denying all reads and writes
Start in test mode
Get set up quickly by allowing all reads and writes to your database
{
"rules": {
".read": true,
".write": true
}
}
Anyone with your database reference will be able to read or write to your database
Cancel
Enable
More features for developers



Update Rules like below :

```
{
 "rules": {
      "live": {
       "Store": {
         ".read": "auth.uid != null",
         ".write": "auth.uid != null",
           "$store_id": {
                  "uid": {
                     ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
                        },
                       }
                },
        "Provider": {
         ".read": "auth.uid != null",
         ".write": "auth.uid != null",
            "$provider_id": {
                   "uid": {
                     ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
                         },
                             }
                    },
        "User": {
         ".read": "auth.uid != null",
         ".write": "auth.uid != null",
            "$user_id": {
                  "uid": {
                     ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
                        },
                       }
```

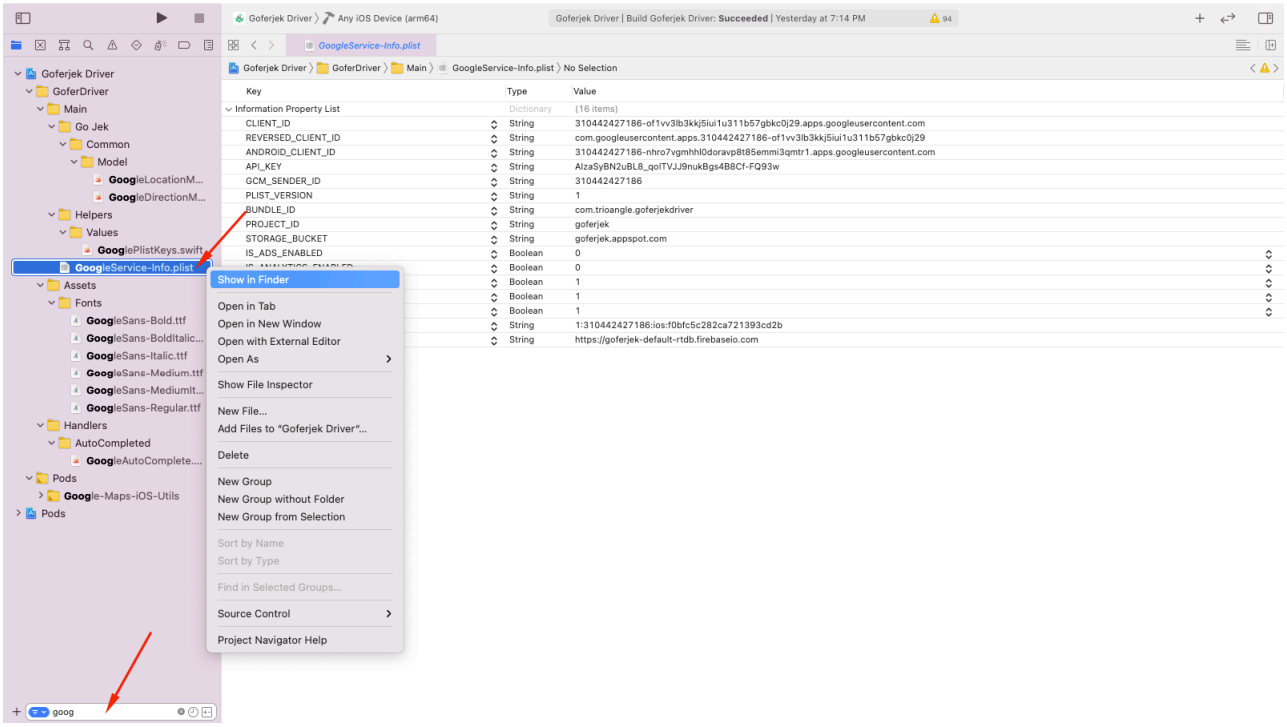
```
                    },
        "GeoFire": {
          ".read": "auth.uid != null",
          ".write": "auth.uid != null",
                  "$trip_id": {
                    "uid": {
                     ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
                          },
                          }
                          },
        "driver_rider_trip_chats": {
           ".read": "auth.uid != null",
           ".write": "auth.uid != null",
                  "$trip_id": {
                  "uid": {
                     ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
                          },
                          }
                         },
        "trip": {
           ".read": "auth.uid != null",
           ".write": "auth.uid != null",
                  "$trip_id": {
                   "uid": {
                    ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
                          },
                          }
                          }
},
     "demo": {
       "Store": {
         ".read": "auth.uid != null",
         ".write": "auth.uid != null",
         "$store_id": {
                  "uid": {
                  ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
                         },
                        }
                        },
        "Provider": {
         ".read": "auth.uid != null",
         ".write": "auth.uid != null",
          "$provider_id": {
                  "uid": {
                  ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
                          },
                          }
                         },

         "User": {
         ".read": "auth.uid != null",
        ".write": "auth.uid != null",
```

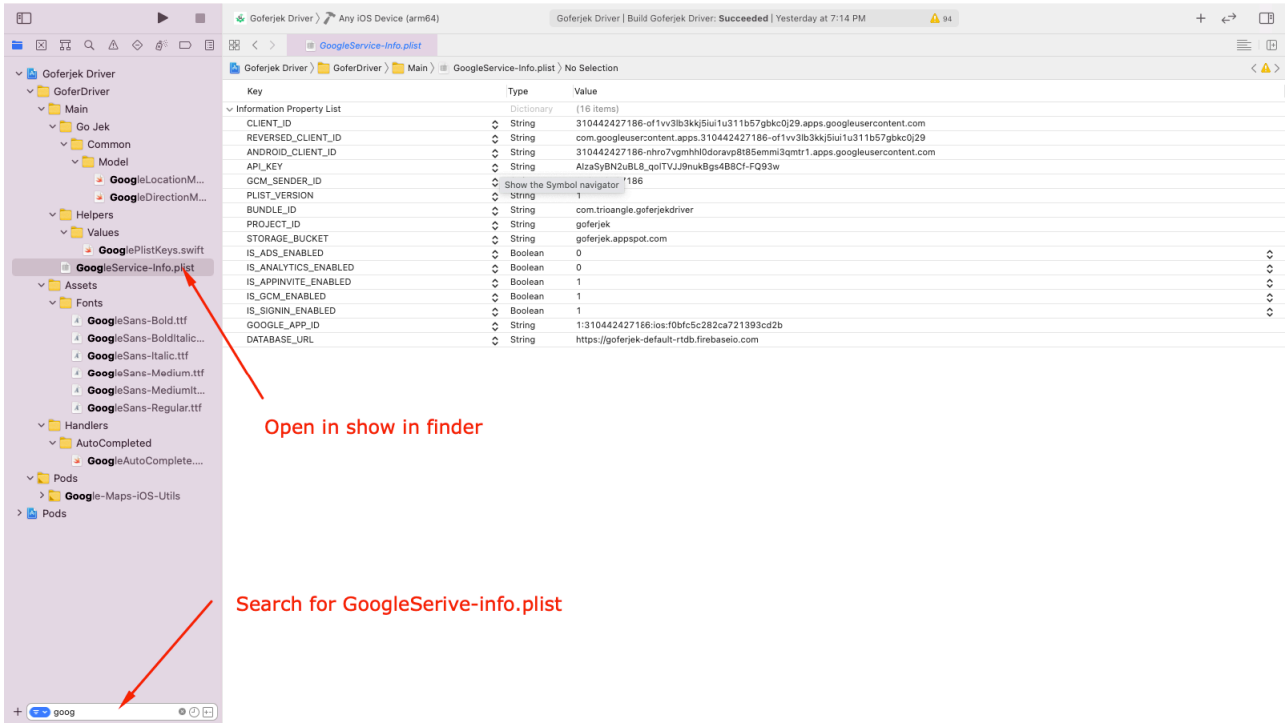
```
        "$user_id": {
         "uid": {
           ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
             },
             }
             },
    "driver_rider_trip_chats": {
              ".read": "auth.uid != null",
               ".write": "auth.uid != null",

              "$trip_id": {
                        "uid": {
                        ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
                                  },
                                  }
                                  },
    "GeoFire": {
               ".read": "auth.uid != null",
               ".write": "auth.uid != null",

              "$trip_id": {
                        "uid": {
                        ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
                                  },
                                  }
                                  },

    "trip": {
              ".read": "auth.uid != null",
              ".write": "auth.uid != null",

              "$trip_id": {
                        "uid": {
                        ".validate": "(data.exists() && data.val() == newData.val()) || newData.val() == auth.uid"
                                 },
                                  }
                                  }
  }
  }
}
```

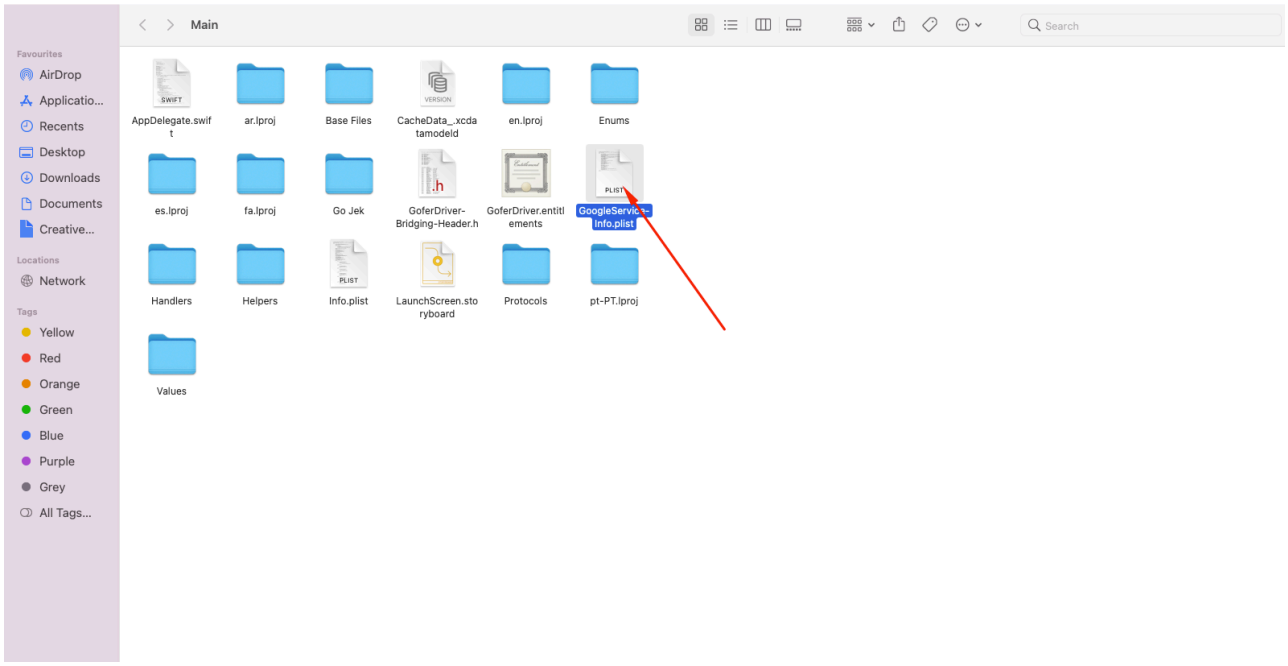
Steps To update FCM Key

- Open the Source folder
- And then open project folder (Goferjek Driver)
- Now to place / replace the **GoogleService-Info.plist** file

Reference Below







4. After replacing the file the uninterrupted notification flows between Provider and User.

Enabling Crash Analytics

- Setup crash analytics by following below steps

Steps For enabling crash analytics

➢ Open [firebase console]

➢ Select your application (refer **Crashlytics 1.1**)

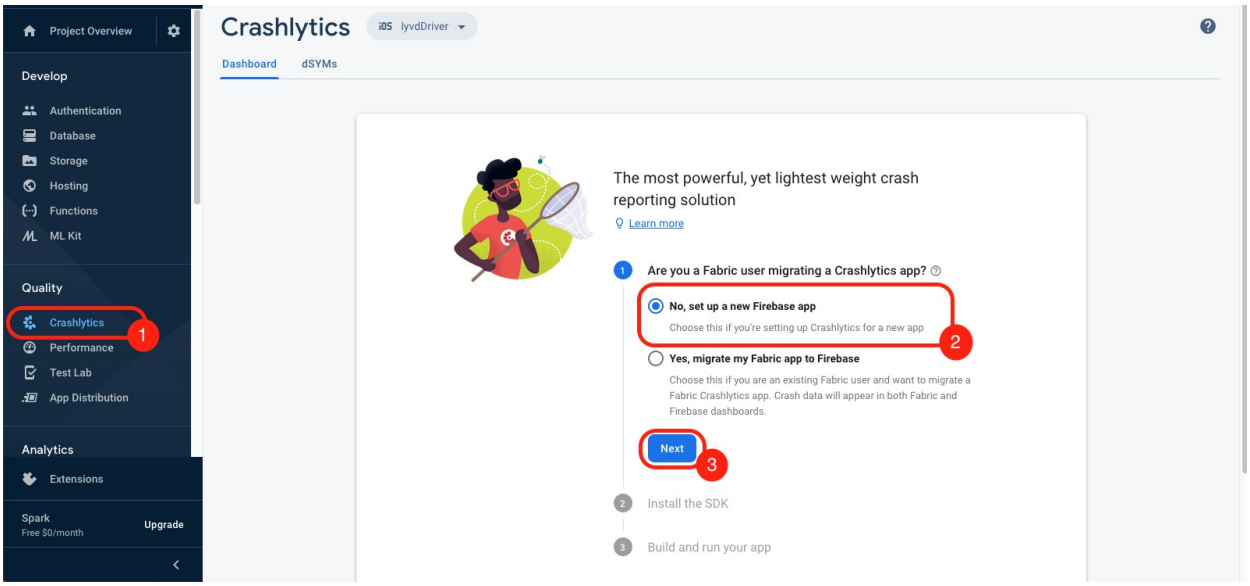
a. Select crashlytics

b. On step 1 of setup select ‘setup new firebase app’

c. And proceed till step 3

➢ Note : [you will need to crash your application to see the crashlytics dashboard]

Reference Below



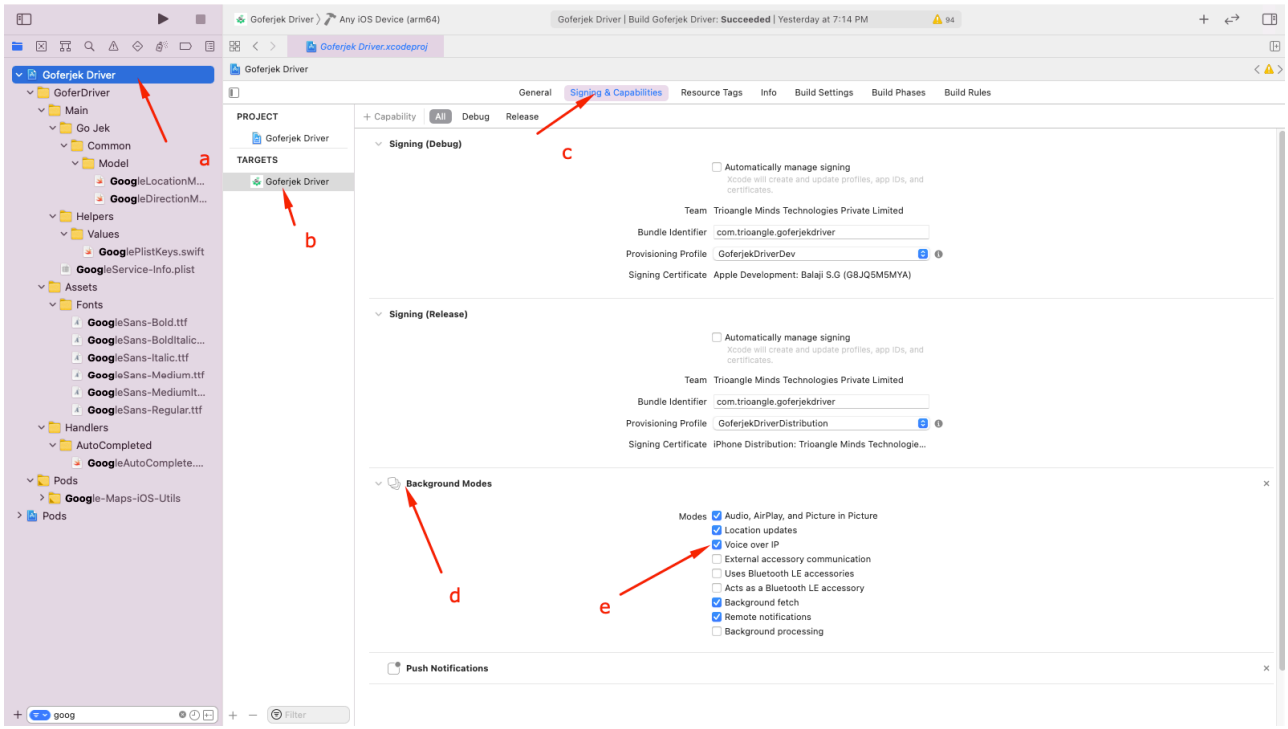
Crashlytics 1.1

API Config document

- Follow the link to get a list of API details (Prams and response).
- **https://documenter.getpostman.com/view/10633012/U16opPB2**

Sinch Call configuration

1. Enable VoIP in Project (Refer below image)



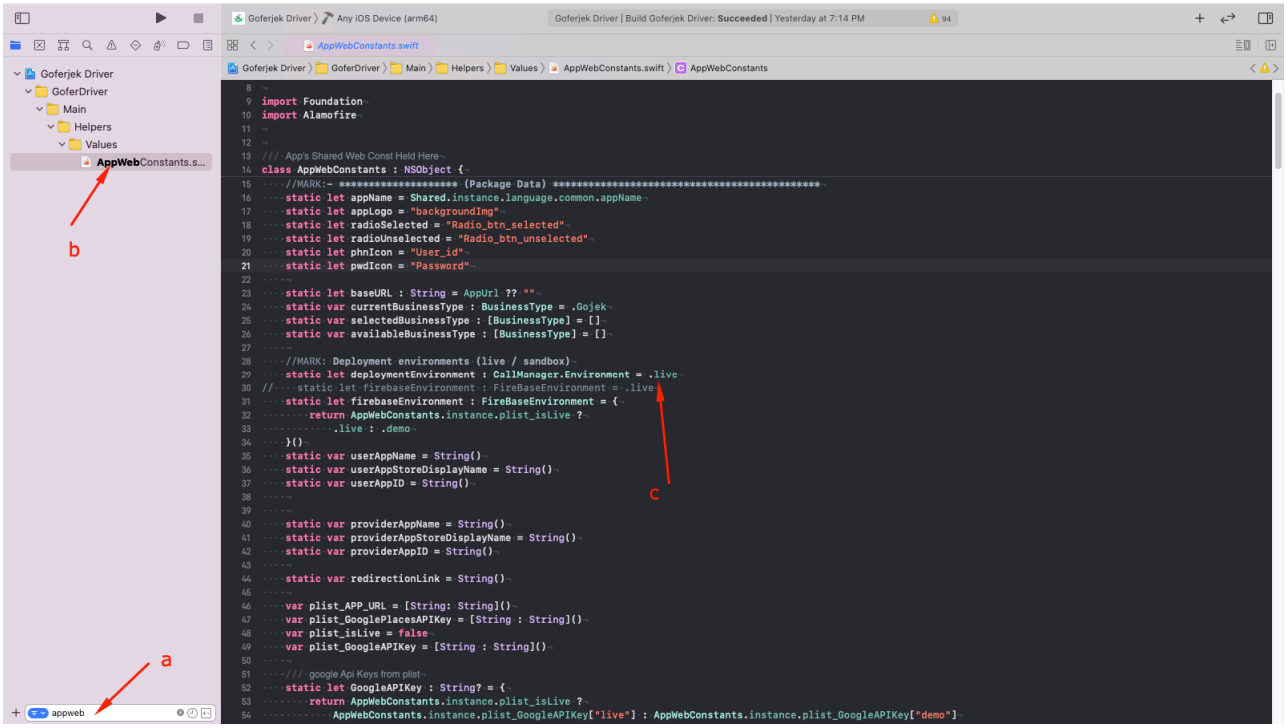
(VoIP Enabling Reference Image)

2. Login To Sinch Admin panel : https://portal.sinch.com/#/login

3. Setting up your Apple Push Certificates with Sinch (for more reference)
   - Login to the Dashboard
   - Open the Apps tab, select the application of your choice and click on the Push Notification icon on the left-most side.
   - Drag and drop the certificates associated to your application
   - If your certificate has a password, enter it and hit Enter

4. Your push notification certificates are now uploaded and ready for use.

5. In Project set "**deploymentEnvironment**" to ".**live**" for production. (Refer below image)
   a. Open Project Search for AppWebConstant.swift
   b. Select AppWebConstant.swift
   c. Change the content to live



(Production environment configuration - reference image)

[Recommended : For exact distance calculation, App should be in open status during the job]