

Search



FoodZo User: version 1.0

Online Documentation:

<http://documentation.optimaprotech.com/#/artical>

First of all, Thank you so much for purchasing this application and for being my loyal customer You are awesome!. We are glad to support to your business growth.

Introduction



FoodZo – Multi Restaurant App Admin and Restaurant Panel

Market your restaurant with Foodzo to increase the growth of your food stall. Expect high and advanced features frequently. The most demanding and trendy designs may increase your sales count. Enlarge your customer circle with the vast zones. Foodzo contains more & more advanced real-time features.

App Features

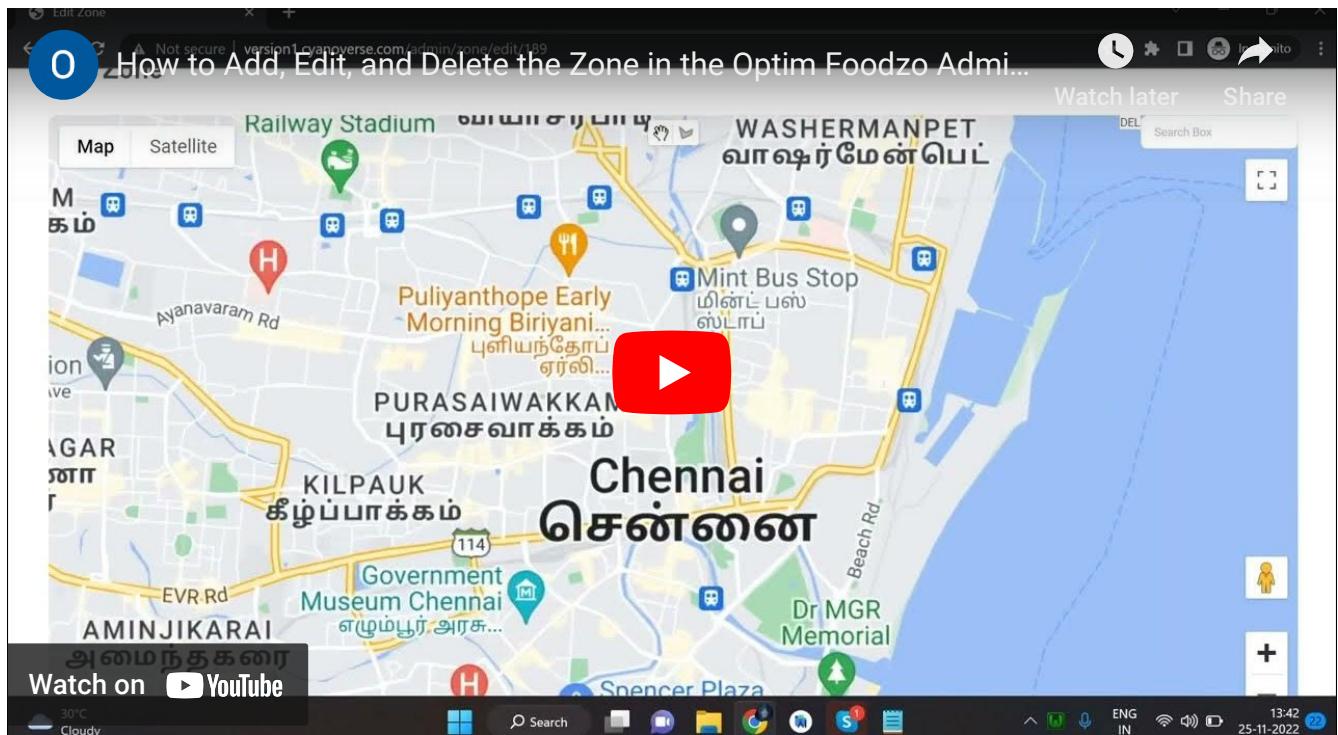
- Smart Login

- Live Track
- Optimized Route Finder
- Auto Menu Rearrange
- Express delivery
- Zone Management
- Real-time status update
- Fast Checkout
- Real-time Notifications
- Light & Dark Themes Supports
- Multi-languages Support (INCLUDING LTR & RTL)
- Multi-currency Support
- Discounts & Coupons
- Review & Rating
- Sales Management
- Easy To Brand & Customize
- Order Tracking
- Display Setting
- Multi User-login Account
- Product Management
- Order Activity Management
- Analytical report
- Delivery tips
- Cooking Instruction
- Delivery Instruction
- Item Filter
- Voice Search
- Thermal printer Support
- Three Different Item Timing slots
- Advanced Outlet Manager
- Multi Variant & Addon Group
- Shop Timing Support
- User Profile
- User Orders
- App Settings
- Notifications
- Help & Supports
- Account Settings
- Tracking Orders..ETC

Video Guide

How to Add, Edit, and Delete the Zone in the Admin panel

Please click the youtube Video Link <https://youtu.be/Cqg-ShrhTQM>



How to register Vendor in Foodzo application/Vendor Registration

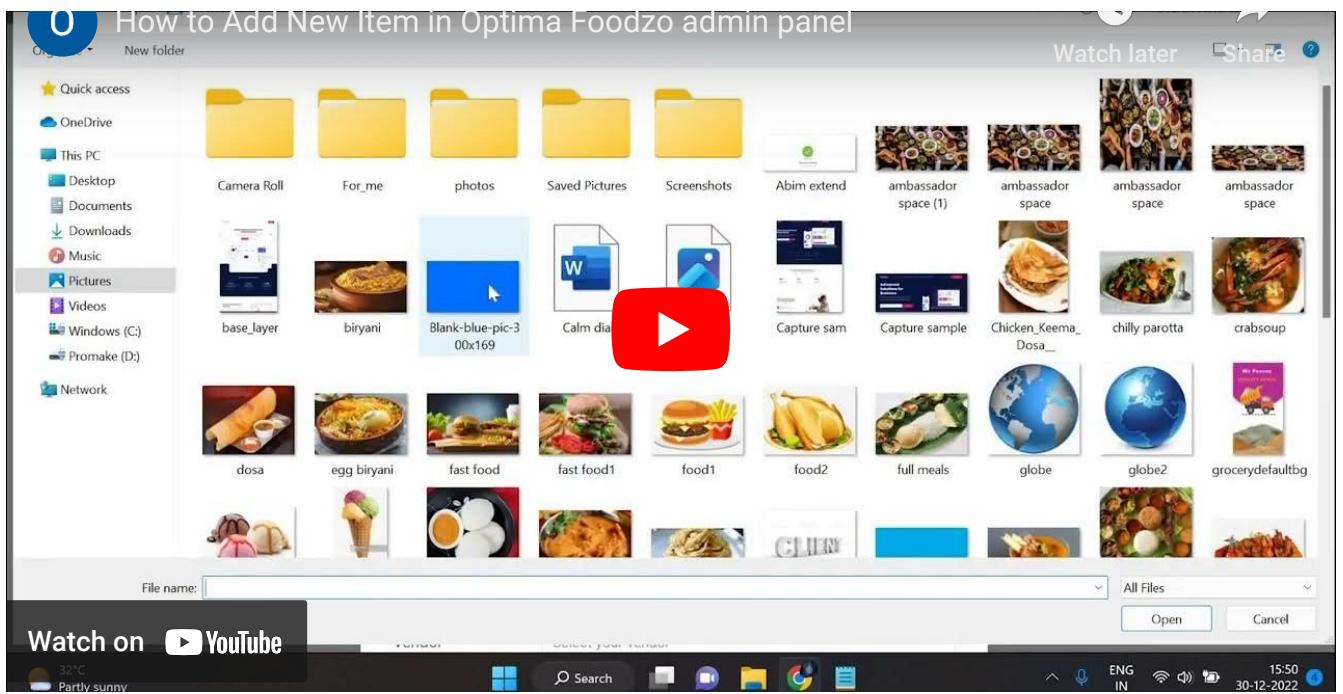
Please click the youtube Video Link https://youtu.be/vJZcKWGmH_c



How to Add New Item in Foodzo admin panel

Please click the youtube Video Link <https://youtu.be/smNRLDnqHQo>





Order Flow Management

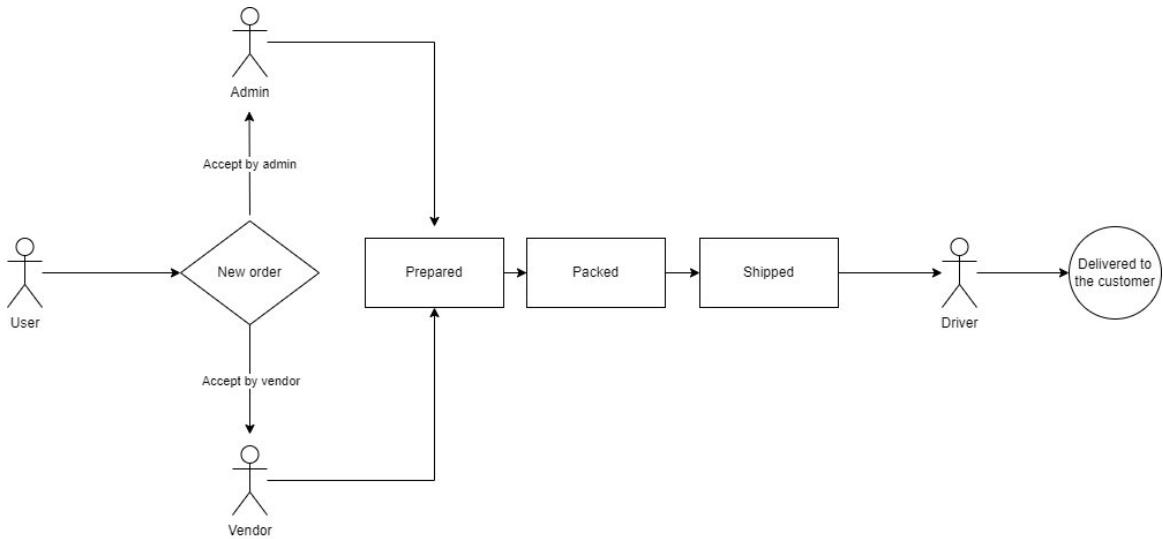
Please click the youtube Video Link <https://youtu.be/RtYr85ip7EQ>

Vendor Payment Settlement process/How to request the vendor wallet amount

Please click the youtube Video Link <https://youtu.be/fne4O9ZbKLg>

Usage Flows(Flowcharts)

Order Flow



Transaction Flow

Item value = item price - (Item discount + Coupon discount for item)

Order value = (Item value + Tax/Vat (on item value) + Delivery Charge) – Coupon discount for delivery charge

For Self Pickup order: Order Value = Item Value + Tax/Vat (on item value) – Coupon discount for delivery charge

Item Total \$ 280.00

```
Tips $ 0.00
Packaging Charge $ 10.00
Delivery Charge $ 300.00
Tax $ 50.40
Discount -$ 5.00
Total $ 635.40
```

Restaurant Earning

Store earning

```
Item Total : $280
Discount : -$5
Admin commission: 10%
Vendor settlement : $247.5 + packaging Charge + Tax
Admin commission: $27.5
```

Requirements

This document is to help you regarding each step of usages. Please go through the documentation carefully to understand how this application works. Flutter, (Dart) knowledge is required to customize this application. You may learn basics here. Best recommended **Flutter (Channel stable, version**

3.3.X) Flutter download

Installation

Step1 : You can start the process by downloading and unzipping the user application file. Open it in any one of the reputed editors(Android studio/ IntelliJ/ Visual code).

Step2 : Go to the file location which is **assets/cfg/configuration.json** in the editor. Now just edit your configuration or remote links to connect the user app with your php server.

Step3 : To change the app icon or logo, go to the file location which is **assets/img/logo.png** in the folder. Now replace your logo instead of the existing image.

Android Reskin

Step1 : Generate your app icons by clicking the given link <https://appicon.co/>.

Now, you can change your app icon and notification icon in the **ic_launcher** and **ic_notification** folders.

Step2 : You also need to replace the icons in the given folder location **android/app/src/main/res**.

Icons should be replaced are /mipmap-hdpi, /mipmap-mdpi, /mipmap-xhdpi, /mipmap-xxhdpi, /mipmap-xxxhdpi

Step3 : Get the dependencies by the command of your editor.

Step4 : Package names should be change. The locations in which the package name should be changed are,

```
/android/app/src/main/AndroidManifest.xml  
android/app/src/debug/AndroidManifest.xml  
/android/app/src/profil/AndroidManifest.xml
```

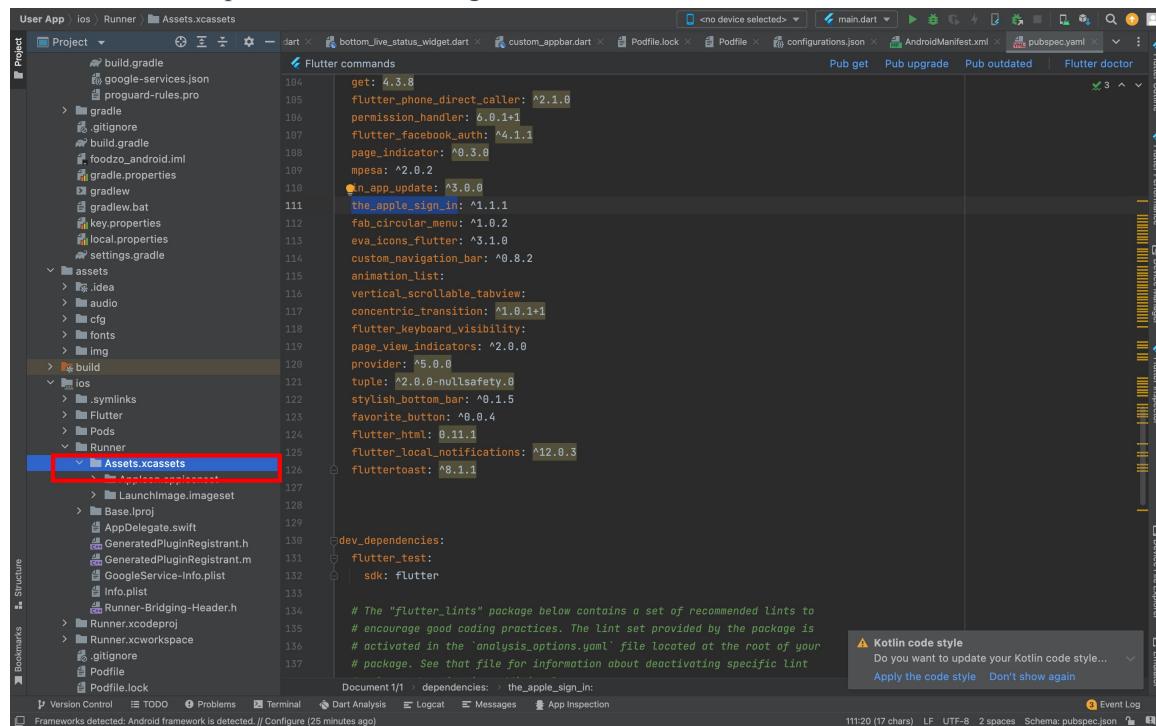
```
/android/app/src/main/kotlin/Your Package name  
folders/MainActivity.kt
```

and change the package name

Open /android/app/build.gradle and change the package name

IOS configuration

You also need to replace the icons in the given folder location



Specify your API key in the application delegate ios\Runner\AppDelegate.m

1. Specify your API key in the application delegate

```
ios/Runner/AppDelegate.m
```

```

#include "AppDelegate.h"
#import "GeneratedPluginRegistrant.h"
#import "GoogleMaps/GoogleMaps.h"

@implementation AppDelegate

@UIApplicationMain
objc class AppDelegate: FlutterAppDelegate
override func application(
    _ application: UIApplication,
    didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool
    GMSServices.provideAPIKey("your key")
    FirebaseApp.configure()
    GeneratedPluginRegistrant.register(with: self)
    return super.application(application, didFinishLaunchingWithOptions)
}

```

Or in your swift code, specify your API key in the application delegate

ios\Runner\AppDelegate.swift

1. Opt-in to the preview of the embedded view by adding a boolean property to the app's Info.plist file with the key `io.flutter.embedded_views_preview` and the value YES.
2. Click the Get dependencies or Packages get to install the libraries from pubspec.yaml file.
3. Open the simulator to run iOS (as the step above)
4. Then press the run button to start the project (you can still open multi simulator at the same time)

Opt-in to the preview of the embedded view by adding a boolean property to the app's Info.plist file with the key `io.flutter.embedded_views_preview` and the value YES.

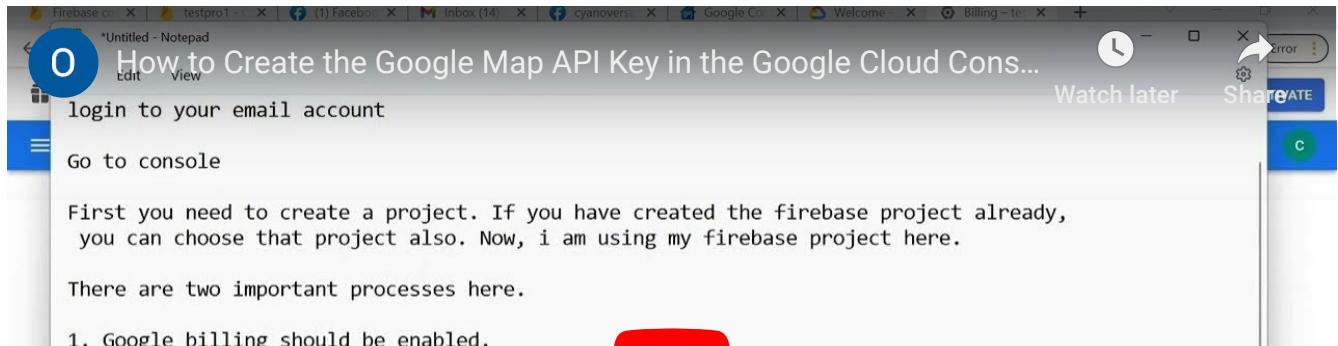
Click the Get dependencies or Packages get to install the libraries from pubspec.yaml file.

Open the simulator to run iOS (as the step above)

Then press the run button to start the project (you can still open multi simulator at the same time)

Google map configuration

Video Link <https://youtu.be/0V8G-mqYkFo>



2. All sdk should be enabled.

If you have a billing account to your logged in email, you can use that.
If not, you need to create the new billing account. Once create the billing account, you need to link that billing account to your google cloud console project.

Give your details to create a billing account. Then enable it.

API key generation process. You should use this key only to access the google map in your android applications

Ln 19, Col 63 90% Windows (CRLF) UTF-8



Step1: You can create the google map API key. Please refer the above video

Note: The google billing account should be enabled to your project in the google cloud console. All api/sdk should be enabled. Then only the google map will work.

The sdk list which should be enabled are listed in the screenshot below,

APIs

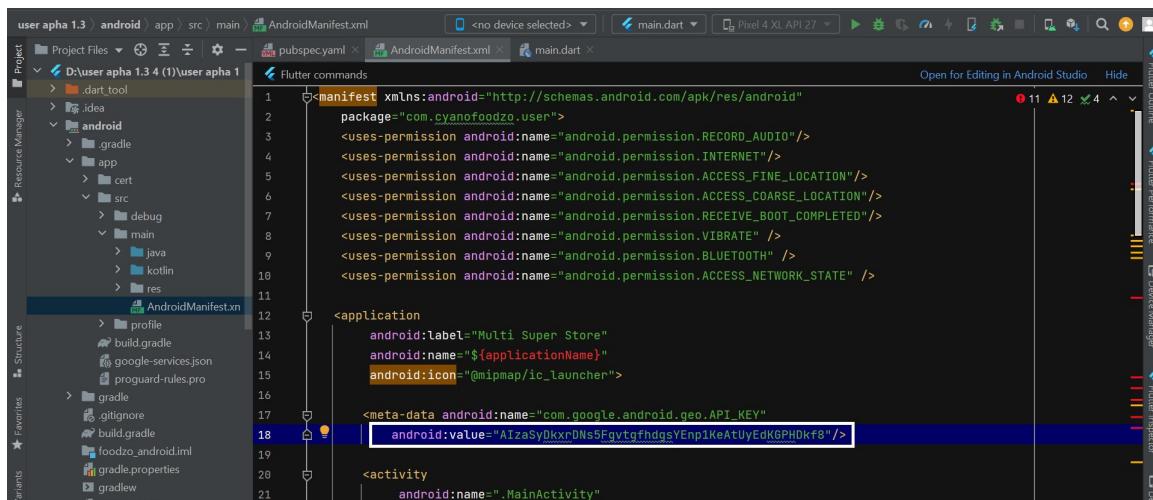
Enabled APIs

Select an API to view details. Figures are for the last 30 days.

API ↑	Requests	Errors	Avg latency (ms)	
Directions API	-	-	-	Details
Distance Matrix API	-	-	-	Details
Geocoding API	-	-	-	Details
Geolocation API	-	-	-	Details
Maps Elevation API	-	-	-	Details
Maps Embed API	-	-	-	Details
Maps JavaScript API	-	-	-	Details
Maps SDK for Android	-	-	-	Details
Maps SDK for iOS	-	-	-	Details
Places API	-	-	-	Details
Roads API	-	-	-	Details

Rows per page: 50 ▾ 1 - 11 of 11 < >

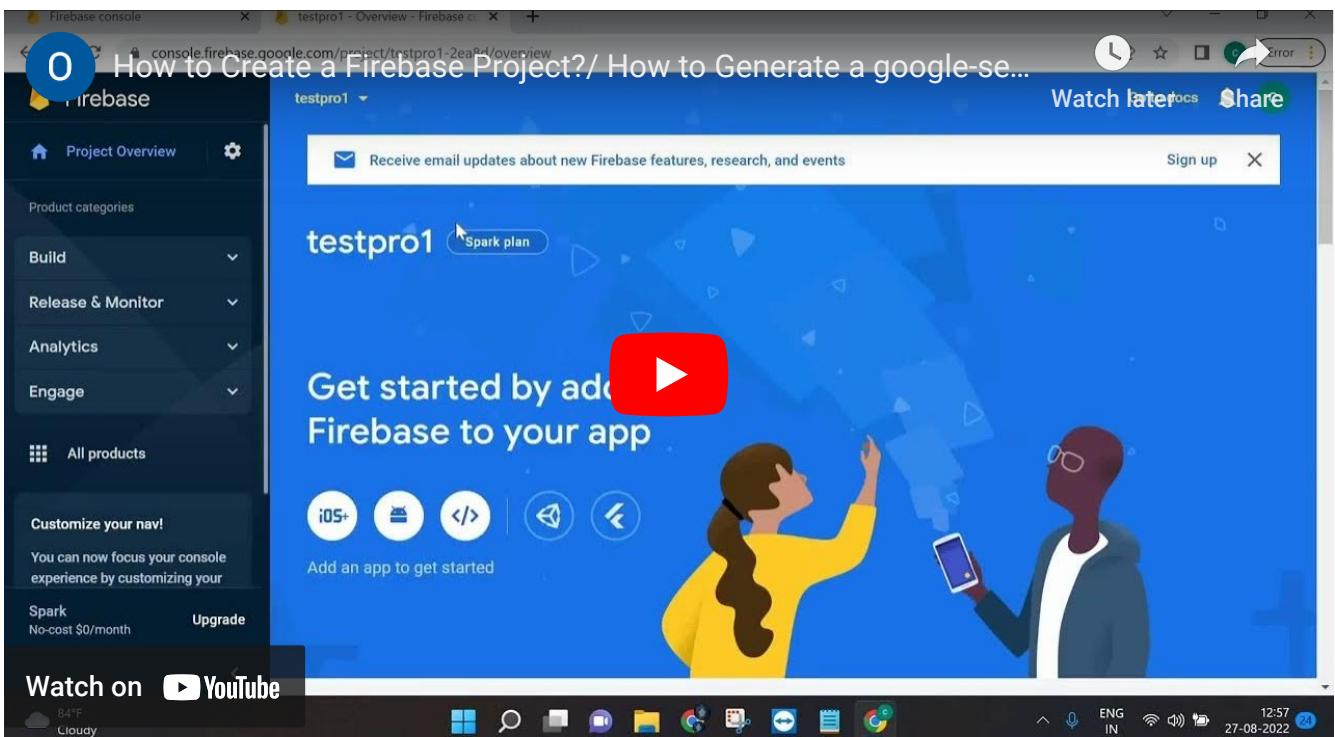
Step2: Once the API key ready, you need to paste it in the particular location which is **android/app/src/main/res/AndroidManifest.xml**. Paste it in the boxed line as it is in the below screenshot.



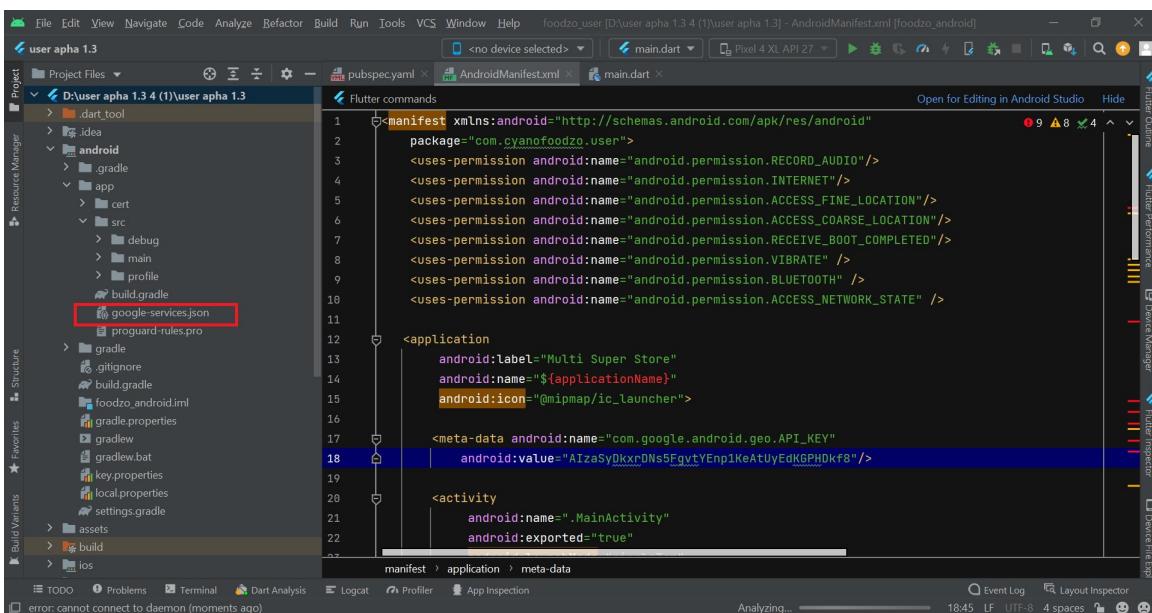


Push notification

Note: To set-up the push notification, the firebase project is need to be created first. I create it, do it now. To create the firebase project and app, verify this tutorial video https://youtu.be/Hi-Zq_ZjJ2w



Copy and paste the generated google-services.json file in android/app folder user

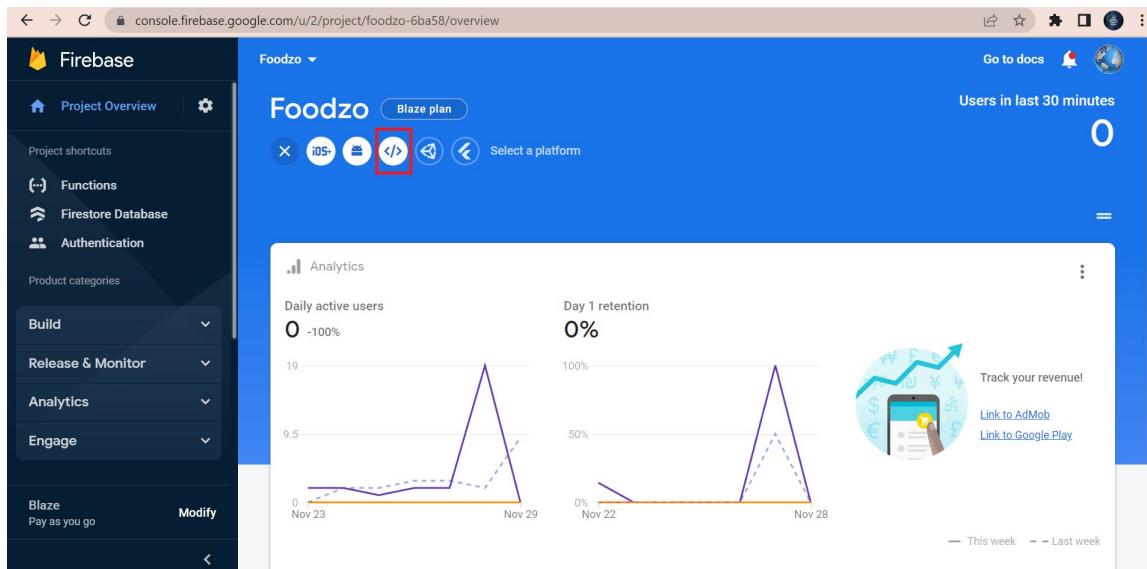


Firebase web configuration

The firebase json code of web application should be pasted in the admin panel.

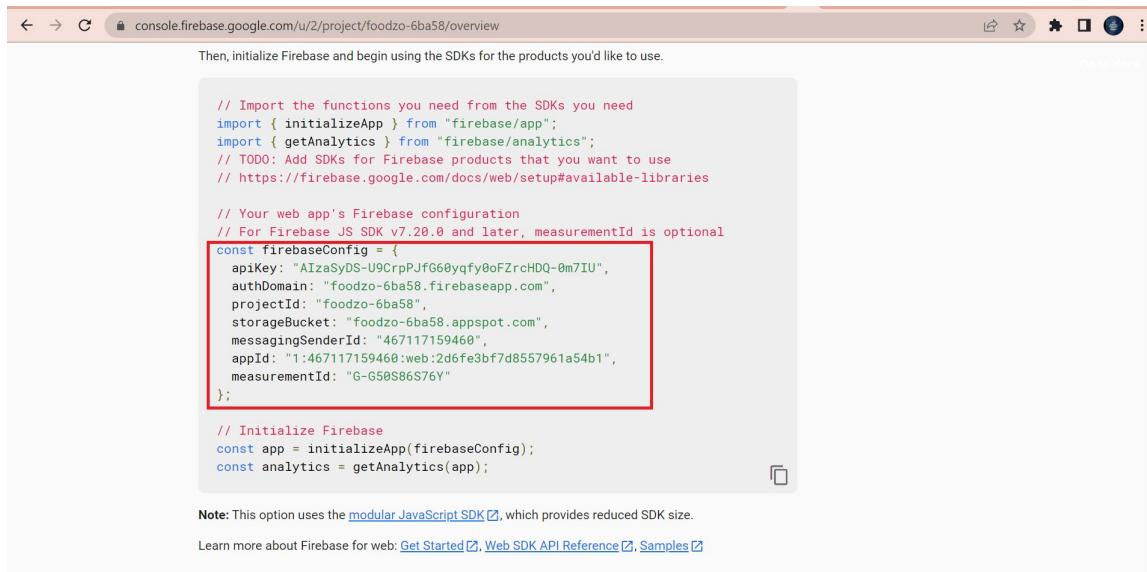
The json code will be generated while creating the web app in the firebase.

See the below screenshot and create the web app in the firebase.



The screenshot shows the Firebase console interface. On the left, there's a sidebar with 'Project Overview', 'Functions', 'Firestore Database', and 'Authentication'. Under 'Build', 'Release & Monitor', and 'Engage', there are tabs like 'Blaze plan', 'Blaze', and 'Modify'. In the main area, it says 'Foodzo' and 'Blaze plan'. Below that, there are icons for 'iOS', 'Android', 'Web', and 'React Native', with 'Web' being the one highlighted by a red box. To the right, there's a chart titled 'Analytics' showing 'Daily active users' and 'Day 1 retention'. A callout bubble says 'Track your revenue!' with links to 'Link to AdMob' and 'Link to Google Play'. At the top right, it says 'Users in last 30 minutes: 0'.

Then the json code will be generated as below screenshot.



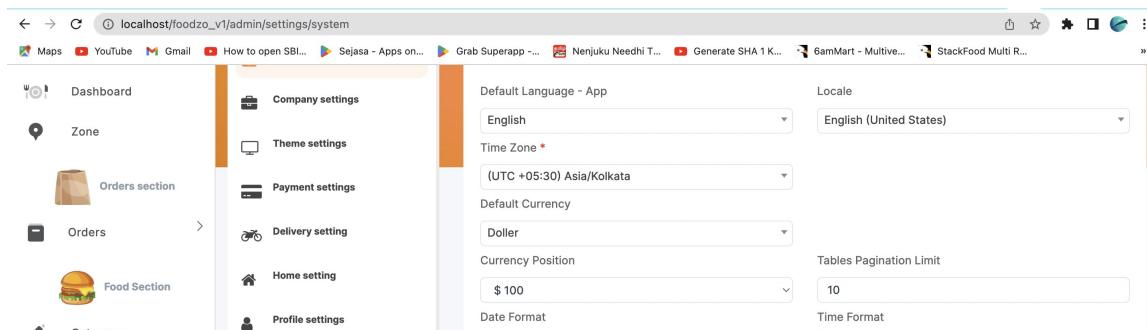
The screenshot shows the Firebase console with the message 'Then, initialize Firebase and begin using the SDKs for the products you'd like to use.' Below this, a large code block is displayed, which is the generated JSON configuration for the web app. The code includes imports for Firebase functions and analytics, and defines a 'firebaseConfig' object with various project settings like apiKey, authDomain, projectId, storageBucket, messagingSenderId, appId, and measurementId. This entire code block is highlighted with a red box. Below the code, there's a note about using the modular JavaScript SDK and links to 'Get Started', 'Web SDK API Reference', and 'Samples'.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyDS-U9CrPjfG60qfy0oFZrcHDQ-0m7IU",
  authDomain: "foodzo-6ba58.firebaseioapp.com",
  projectId: "foodzo-6ba58",
  storageBucket: "Foodzo-6ba58.appspot.com",
  messagingSenderId: "467117159460",
  appId: "1:467117159460:web:2d6fe3bf7d8557961a54b1",
  measurementId: "G-G50S86S76Y"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

Now, paste the code in the admin panel as below screenshot.

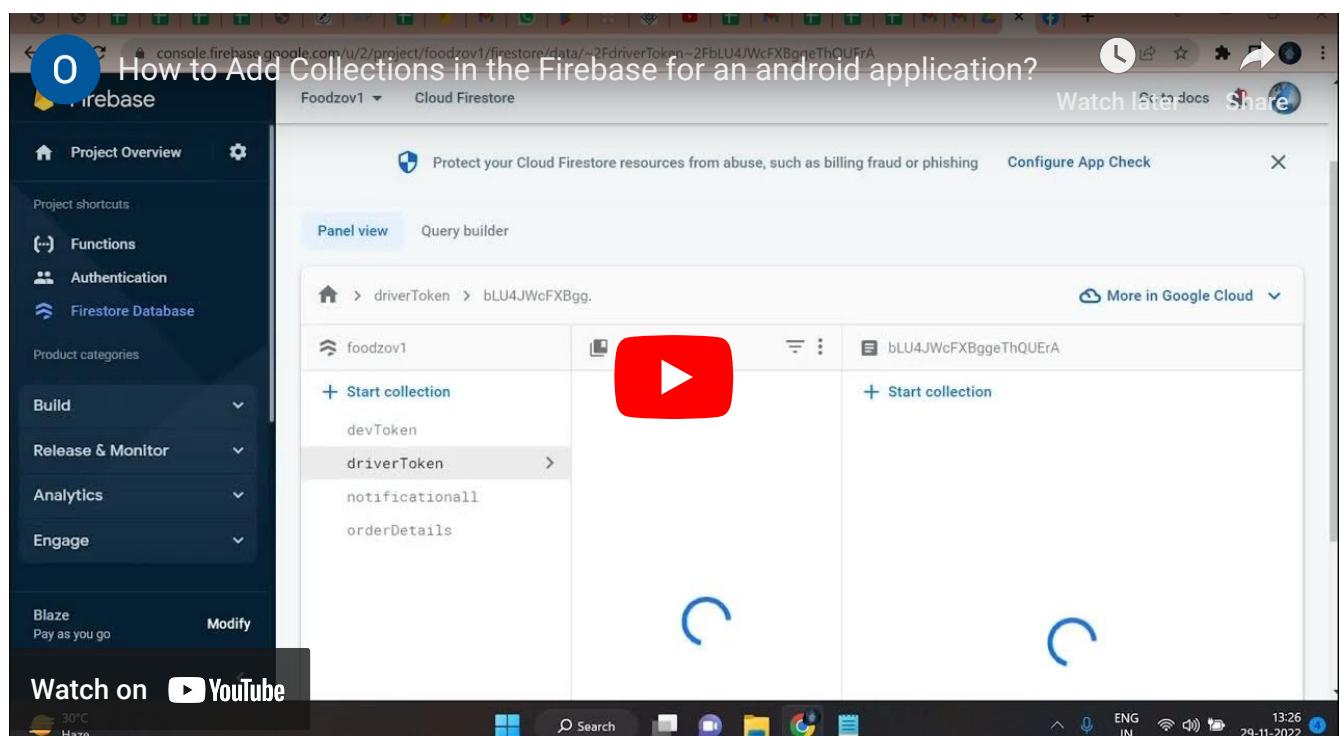


The screenshot shows the admin panel at 'localhost/foodzo_v1/admin/settings/system'. The left sidebar has sections like 'Dashboard', 'Zone', 'Orders section', 'Orders', 'Food Section', and 'Category'. The main area shows 'Company settings' with fields for 'Default Language - App' (English), 'Locale' (English (United States)), 'Time Zone' (UTC +05:30) Asia/Kolkata, 'Default Currency' (Dollar), 'Currency Position' (\$ 100), 'Tables Pagination Limit' (10), and 'Time Format'. The 'Company settings' section is highlighted with an orange box.

The screenshot shows the FoodZo Admin panel under the 'SMTP settings' section. On the left, there's a sidebar with icons for Subcategory, Menu, Item, Addon, Custom Addon group, Custom Variant group, Customer Section, and Customer list. The main area displays 'Nov 30, 2022' and '11:22 AM'. Under 'Allowed Files', it lists 'Money Format', '1,234.00', 'Max File size (MB)', '5000', and 'Allowed file types: gif|png|jpeg|jpg|pdf|doc|txt|docx|xls|zip|rar|xls|mp4|ic'. Below this is a 'Google API KEY' field with the value 'AlzaSyDgnKfYmvV-G-WYMJEC7nX2GyFCWCzTc'. A red box highlights the 'Firebase Key' field, which contains the following JSON:

```
{
  "apiKey": "AlzaSyDS-U9CrpPJG60yqfy0oFZrcHDQ-0m7IU",
  "authDomain": "foodzo-6ba58.firebaseioapp.com",
  "projectId": "foodzo-6ba58",
  "storageBucket": "foodzo-6ba58.appspot.com",
  "messagingSenderId": "105404400000"
}
```

Note: If you have already done the below process, you don't need to do this again.



You need to add the below collections to your firebase project.

1. devToken, 2. notificationall, 3. orderDetails, 4. driverToken

Login procedures

G-mail and phone login

To enable the gmail and phone login, you need to copy the sha keys and paste them in the firebase.

You can login to the local app by generating and pasting the sha keys in the firebase.

Note: To generate the sha keys for your local system, you can verify the below video.

0 How to Create Sha Keys for the local system? and How to Enable...

```
cd android
gradlew signinReport
```

Now the sha keys generated to your local system.

```
Sha1: 2F:EC:DE:B1:97:B4:3B:DA:0A:30:E8:93:DF:A0:99:AE:49
Sha2: |
```

Ln 16, Col 7 100% Windows (CRLF) UTF-8

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.

Watch on YouTube

If you need the sha keys of the published app, go to the play console developer account. Go to the **All apps** in the side navigation then open your user app.

Then go to the **App integrity** option of the **Setup** in the side navigation. You will be having four sha keys in that page. Just look the below screenshots.

Google Play Console

Reach and devices

Device catalog

App bundle explorer

Setup

App integrity

Internal app sharing

Advanced settings

Grow

Store presence

Main store listing

Custom store listings

Store listing experiments

App integrity

Integrity API

App signing

App signing key certificate

This is the public certificate for the app signing key that Google uses to sign each of your releases. Use it to register your key with API providers. The app signing key itself is not accessible, and is kept on a secure Google server.

MD5 certificate fingerprint	BE:1B:A2:86:0A:FF:CD:26:C2:D6:D8:21:37:5F:7A:B0	Download certificate
SHA-1 certificate fingerprint	3D:14:FF:FD:B8:38:A2:8E:4C:E9:30:CB:35:47:56:C5:28:59:FB:86	Download certificate
SHA-256 certificate fingerprint	06:FD:C0:04:42:A2:B7:0E:80:7E:96:83:32:49:40:32:86:16:66:A8:3D:65:B4:EB:48:21	Download certificate

Upgrade your app signing key

Google Play Console

Reach and devices

Device catalog

App bundle explorer

Setup

App integrity

Internal app sharing

Advanced settings

Once annually, you can upgrade your app signing key to move your users to a new key. [Learn more](#)

Request key upgrade

Upload key certificate

This is the public certificate for your private upload key. Use your upload key to sign each release so that Google knows updates are from you. Use the certificate below to register your upload key with API providers.

Grow	MD5 certificate fingerprint	93:3C:B9:8C:78:3B:0A:CA:CD:FC:C8:5D:4B:2C:C3:6B	<input type="button" value="Copy"/>
Store presence	SHA-1 certificate fingerprint	83:C3:FD:6D:71:22:0D:D9:0B:9C:67:3D:A3:A8:9D:D3:79:8E:67:7D	<input type="button" value="Copy"/>
Main store listing	SHA-256 certificate fingerprint	7E:25:1A:A9:4B:8E:BB:44:D3:99:91:29:23:7A:AB:1D:7D:09:B7:62:41:02:F9:29:87:3	<input type="button" value="Copy"/>
Custom store listings			
Store listing experiments		If you have lost your upload key, please contact our support team.	

Then go to your firebase project and open your user app setting page. Scroll down to the bottom, you can see the **Add fingerprint** option. Click that and paste your four sha keys (sha 1, sha 256) one by one.

The top screenshot shows the 'Project Overview' page for the 'testpro1' project. On the left sidebar, 'Authentication' is selected. In the main area, there's a card for 'com.testpro1.user' with a gear icon. A red arrow points to this gear icon. The bottom screenshot shows the 'Project settings' page for the same app. Under the 'General' tab, the 'SHA certificate fingerprints' section is expanded, showing the SHA-1 and SHA-256 values from the previous table. Below this, there's a blue 'Add fingerprint' button.

Now, go to the **Authentication** option of the firebase. It will be there in the side navigation. Then click **sign-in method** and give **Add new provider**.

You need to enable the phone and gmail options.

The screenshot shows the 'Authentication' page in the Firebase console. The 'Sign-in method' tab is active. Below this, the 'Sign-in providers' section is shown, which is currently empty. The left sidebar shows 'Authentication' is selected.

Get started with Firebase Auth by adding your first sign-in method

Native providers	Additional providers	Custom providers
Email/Password	Google, Facebook, Play Games	OpenID Connect
Phone	Game Center, Apple, GitHub	SAML
Anonymous	Microsoft, Twitter, Yahoo	

If you done all these steps correctly, the g-mail and phone login will work fine.

Facebook login



To set-up the facebook login, verify this tutorial video link <https://youtu.be/S2EqUjc7My4> and follow each steps carefully. If you done all the steps correctly, the facebook login will work fine.

Note: To set-up the facebook login, verify this tutorial video link <https://www.youtube.com/watch?v=CjbmvgbiuJQ> and follow each steps carefully. If you done all the steps correctly, login will work fine.

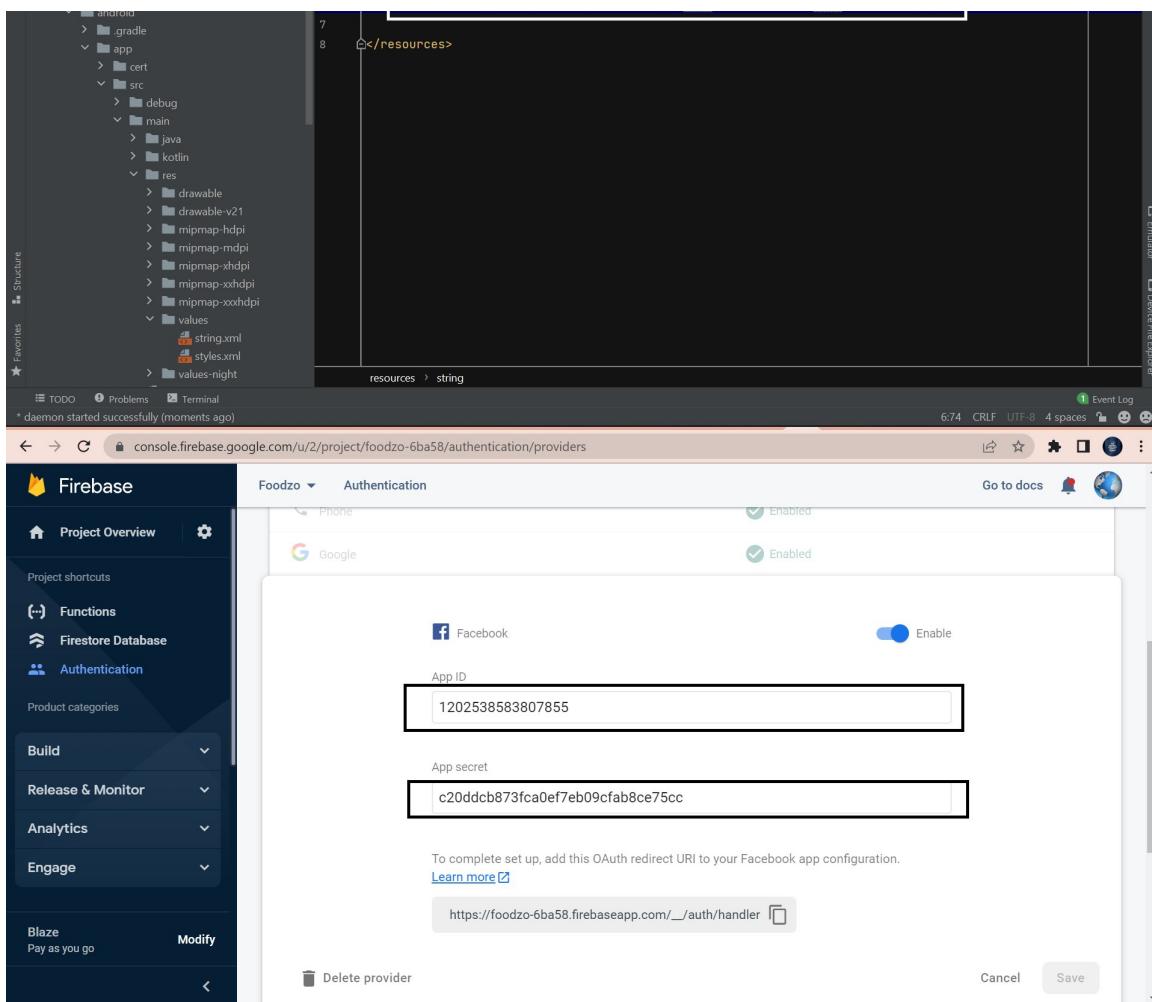
You need to enable the facebook option too in the firebase authentication option.

The **app id** and the **secret key** should be pasted in the firebase and the user app of the particular location. Please see the screenshots below.

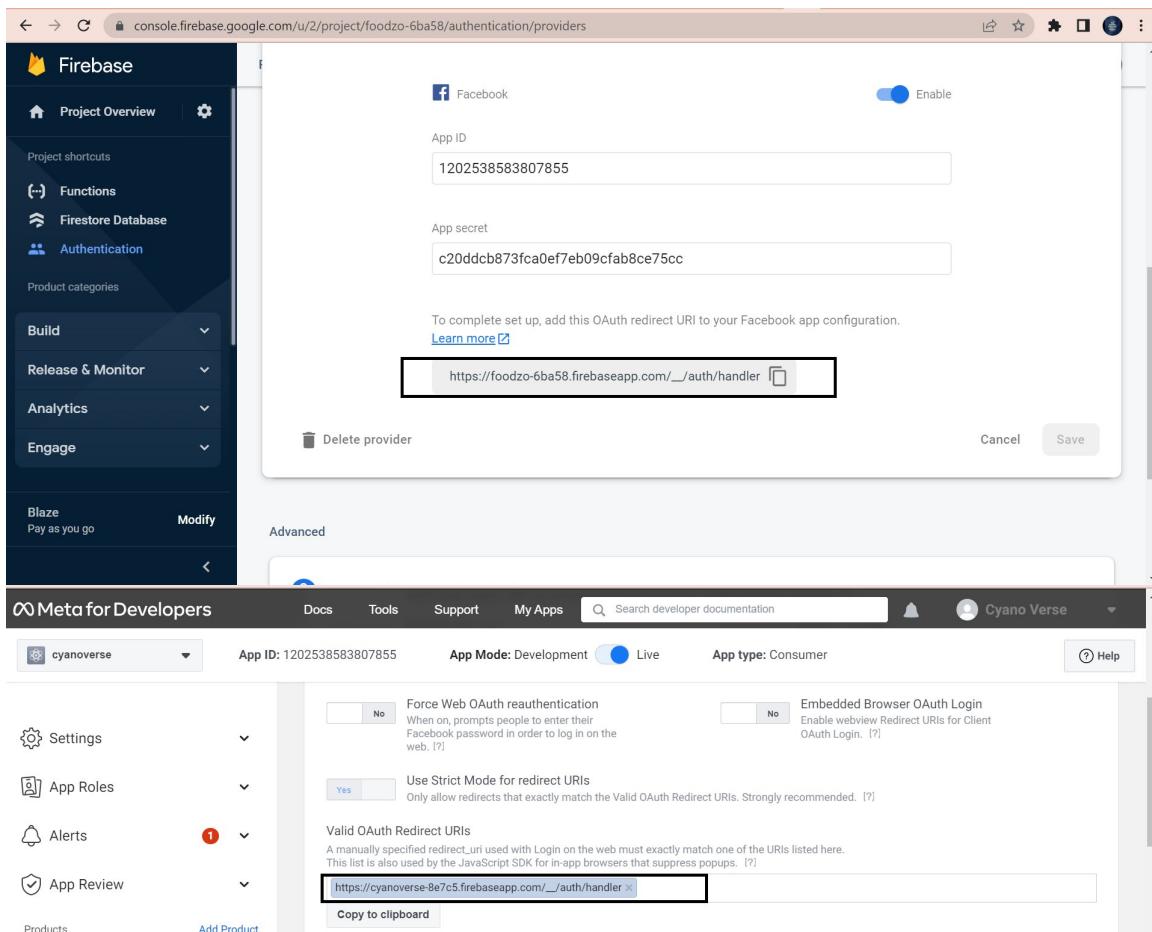
```

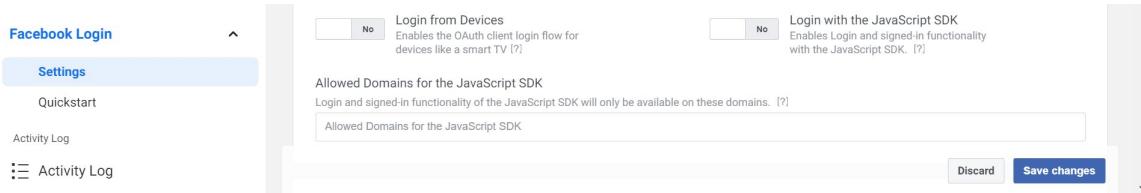
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">fblogin</string>
    <string name="facebook_app_id">1202538583807855</string>
    <string name="fb_login_protocol_scheme">fb1202538583807855</string>
    <string name="facebook_client_token">c20ddc0873fca0ef7eb09cfab8ce75cc</string>

```



The **auth url** of the firebase should be pasted in the facebook app. Please see the below screenshots.





Run application

In the target selector, select an Android device for running the app. If none are listed as available, select Tools > Android > AVD Manager and create one there.

Once selected the device, run the application by using the command **flutter run**

If you need to build the application, you can use the command **flutter build apk**

If you need to build the aab file to upload the app to the play store, you can use the command **flutter build appbundle**

You can increase the version code in the pubspec.yaml file.

Source code details

Structure of the project

The explanation for the directories of the **project/lib/src**.

The design developed in MVC pattern on both user

The lib folder is divided into eight sub folders. Those are animation, components, controllers, elements, helpers, models, pages and repository.

Animation: Contains files used for animation.

Components: Contains files used for components.

Elements: Contains files used for elements.

Helpers: Contains files used for calculate logic helpers.

Models: Contains files used for dummy data to show.

Pages: Contains files used for multiple other UI files

Translation

Go to the file lication **flutter_application/lib/l10n** and duplicate the file **intl_en.arb**. Then rename it to

your local language code example: (intl_fr.arb for french) and translate it.

Run this command into the flutter application folder

```
flutter pub run intl_utils:generate
```

Main files

The main file which is located in **project file/lib/main.dart** contains global configuration (Tilte/Themes/Font family/Colors...) of the app.

Was this article helpful?

 Yes  No

Have more questions? Submit a request

Copyright 2022 Powered by Optimaprotech