Push Notifications
with
Firebase Cloud Messaging (FCM)

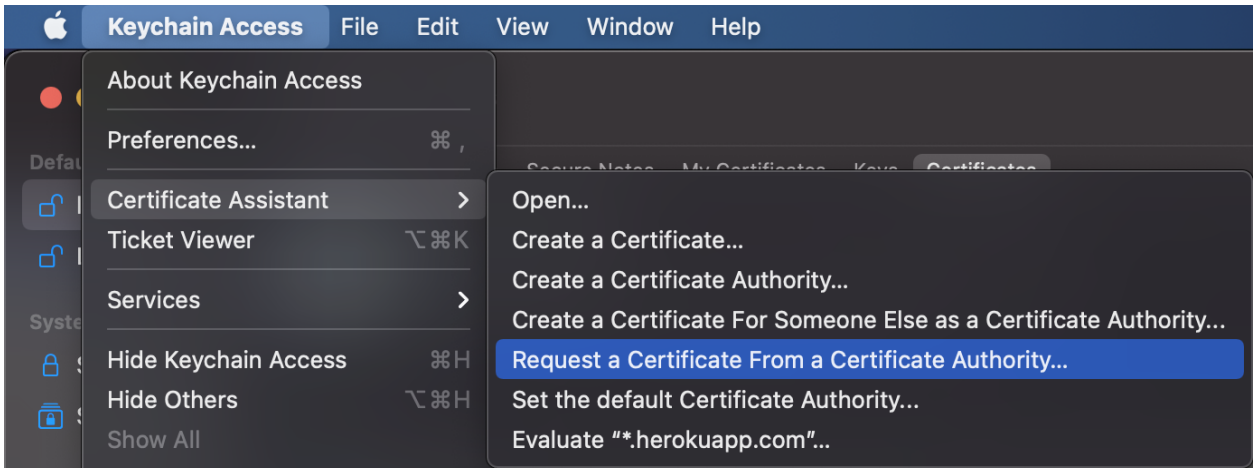
//

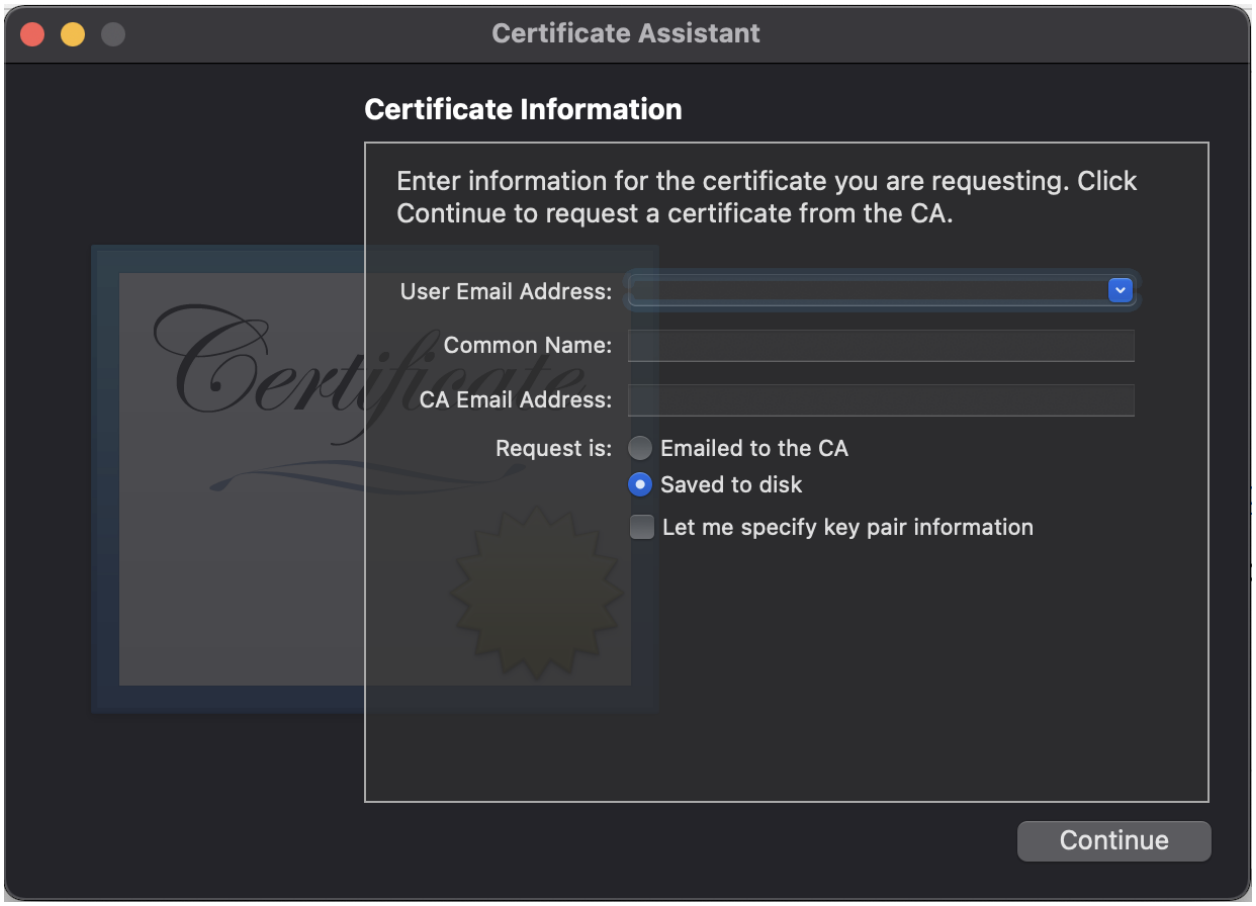
MOBILE APP

//////////////////////////

Apple Push Services Certificate

Prepare “Certificate Signing Request” file





Create “Development SSL Certificate”

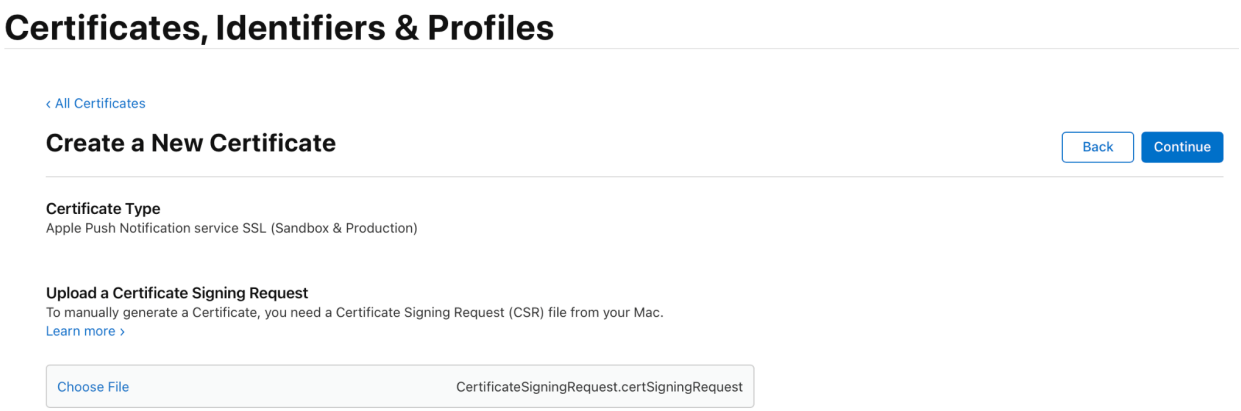
https://developer.apple.com/account/

Certificates, Identifiers & Profiles -> Identifiers -> (Open the app)

Capabilities -> Check “Push Notifications” and Save (top right corner of the page)

Click “Configure” button beside “Push Notifications”

Upload the Certificate Signing Request file prepared just now:



Continue and then Download it (aps_development.cer)

Double-click aps_development.cer to install it to Keychain Access

In Keychain Access find the installed certificate and expand it, there shall be a private key along with the certificate. (If you didn't see the private key, most probably it was because the Certificate Signing Request file is stale and this certificate is useless. You need to delete it from Keychain Access, revoke it from Apple Developer portal and then generate a fresh Certificate Signing Request file and create a new push notification certificate.) Choose both the certificate and its private key in Keychain Access and right-click on them, choose "Export 2 items…" and choose a password, save the p12 file.

Create "Production SSL Certificate"

Repeat the above steps to create Production SSL Certificate and export the certificate and its private key to a p12 file.

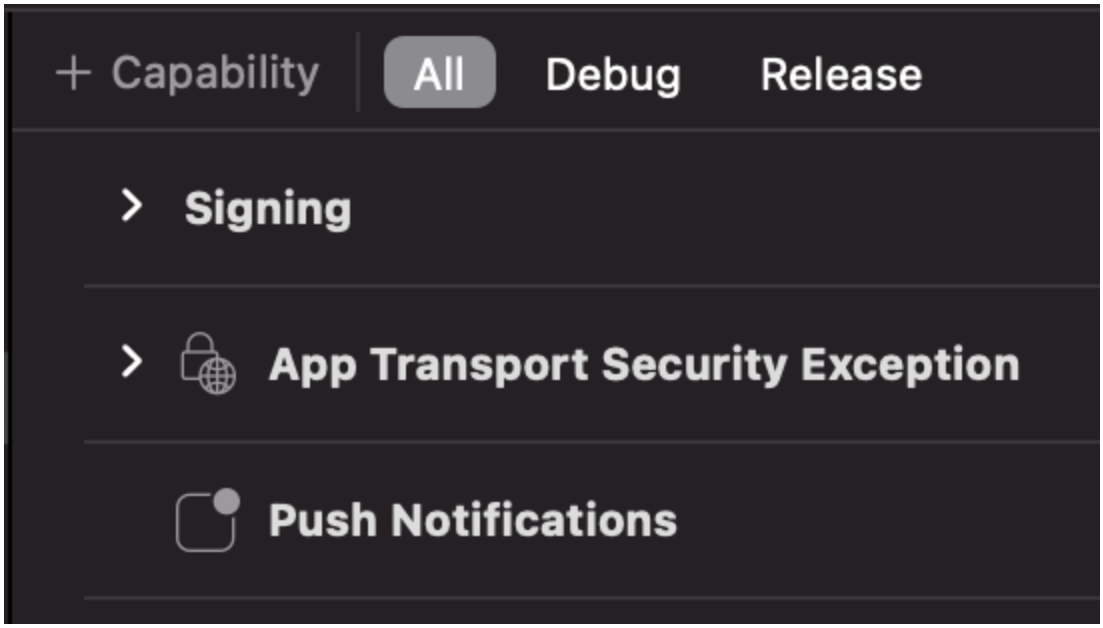
//////////////////////////

iOS App Capabilities

In Xcode,

Project -> target -> "Signing & Capabilities" -> "+ Capability"

Double-click "Push Notifications" to add the capability



//////////////////////////

FCM (Firebase Cloud Messaging)

////////

Go to Firebase console, choose the project where your node server app is added – mobile/web apps must be added to the same project as your node server, otherwise FCM won't work.

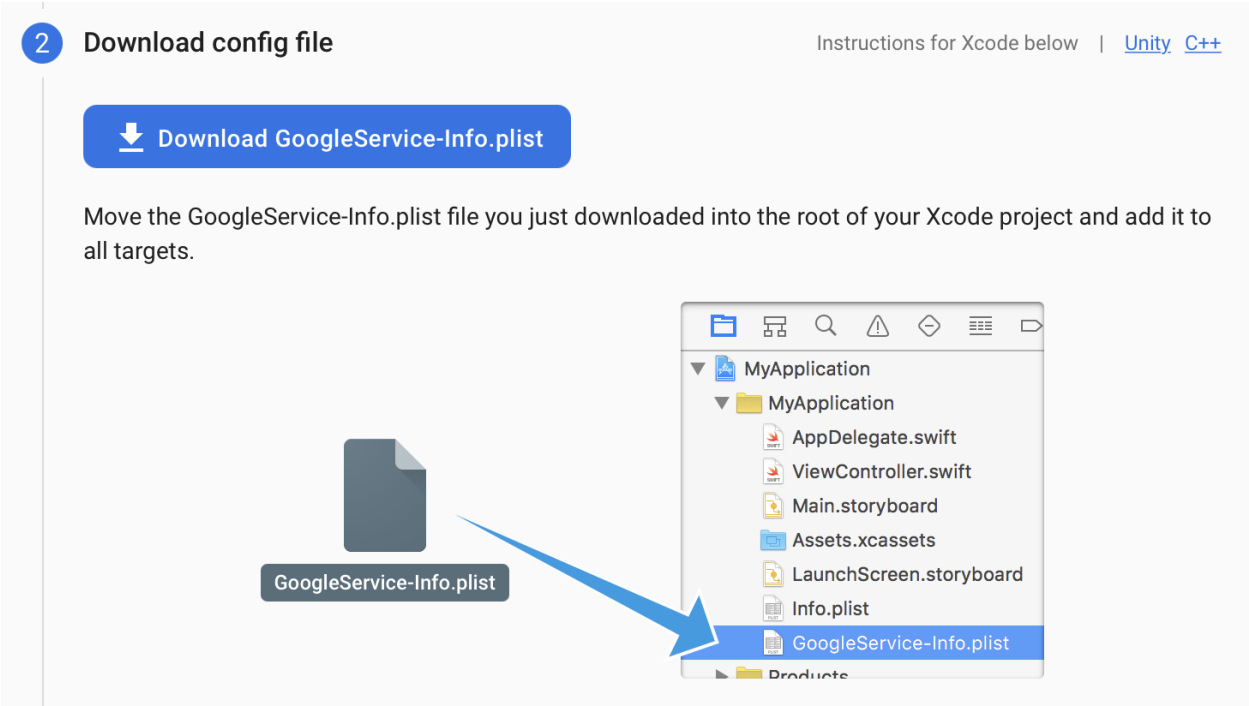
Project Settings -> "General" tab

Add iOS App to Firebase Project

"Add app" button, choose "iOS"

Provide Apple bundle ID, and app nick name -> Register App

Download GoogleService-Info.plist and add to the iOS project



Skip the next step of “Add Firebase SDK” because we don’t need this for React Native app’s push notification service. (We have already installed the minimum set of React Native packages by `yarn add @react-native-firebase/app @react-native-firebase/messaging` and `yarn pod` in previous step.)

GoogleService-Info.plist contains credentials and shall not be uploaded to Git. Make sure you add
ios/<project_name>/GoogleService-Info.plist
to .gitignore

Add initialization code

4 **Add initialization code**

To connect Firebase when your app starts up, add the initialization code below to your main **`AppDelegate`** class.

○ Swift ◉ Objective-C

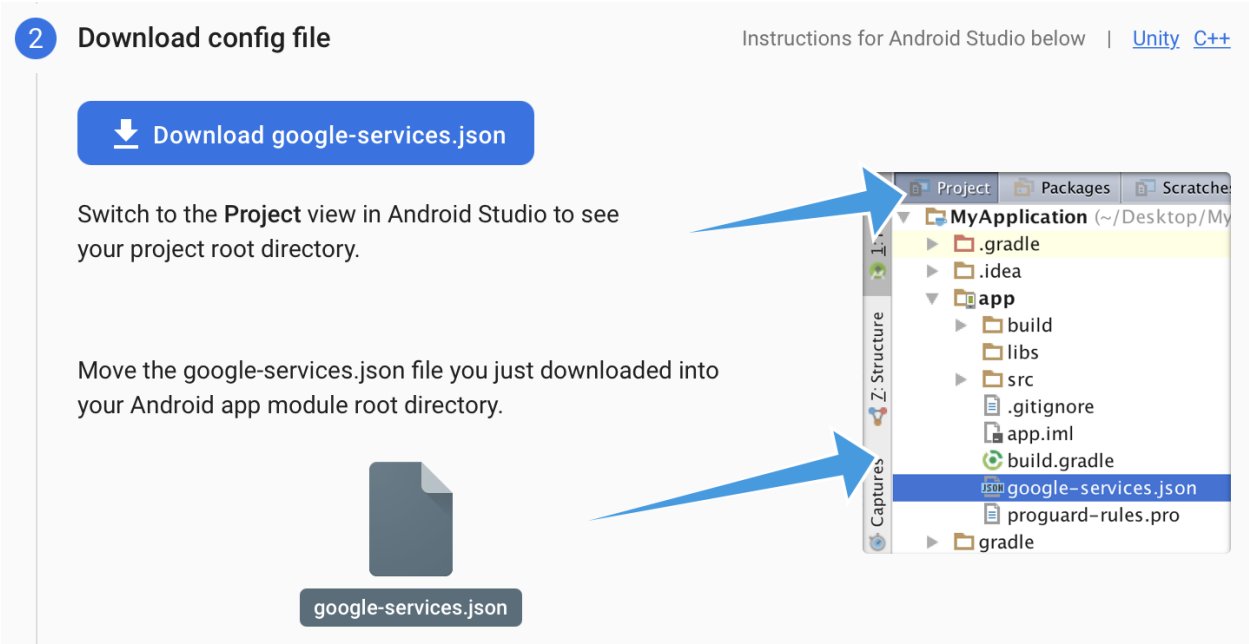
```
@import UIKit;
@import Firebase;

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application
    didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
  [FIRApp configure];
  return YES;
}
```

Add Android App to Firebase Project

“Add app” button, choose “Android”

Provide Android package name, and app nick name -> Register App

Download GoogleService-Info.plist and add to the Android app’s folder



google-services.json contains credentials and shall not be uploaded to Git. Make sure you add
android/app/google-services.json
to .gitignore

Add Firebase SDK

3 Add Firebase SDK

Instructions for Gradle | Unity C++

The Google services plugin for Gradle loads the `google-services.json` file you just downloaded. Modify your build.gradle files to use the plugin.

Project-level build.gradle (`<project>/build.gradle`):

```
buildscript {
  repositories {
    // Check that you have the following line (if not, add it):
    google()  // Google's Maven repository
  }
  dependencies {
    ...
    // Add this line
    classpath 'com.google.gms:google-services:4.3.10'
  }
}

allprojects {
  ...
  repositories {
    // Check that you have the following line (if not, add it):
    google()  // Google's Maven repository
    ...
  }
}
```

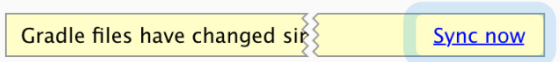
(●) Java ( ) Kotlin

App-level build.gradle (`<project>/<app-module>/build.gradle`):

```
apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'

dependencies {
  // Import the Firebase BoM
  implementation platform('com.google.firebase:firebase-bom:29.0.4')

  // Add the dependency for the Firebase SDK for Google Analytics
  // When using the BoM, don't specify versions in Firebase dependencies
  implementation 'com.google.firebase:firebase-analytics'

  // Add the dependencies for any other desired Firebase products
  // https://firebase.google.com/docs/android/setup#available-libraries
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. Learn more

Finally, press "Sync now" in the bar that appears in the IDE:

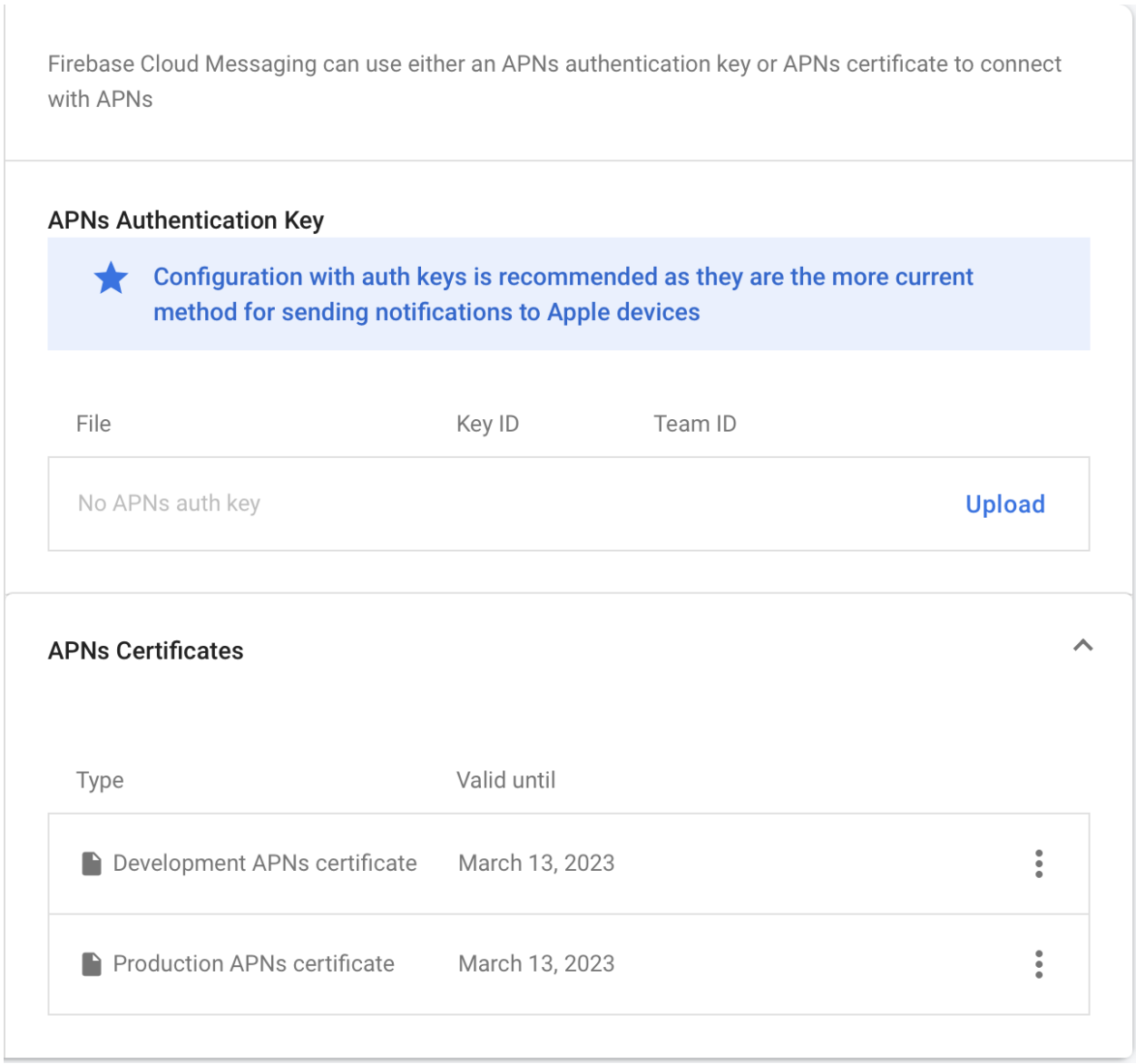


////////

Upload APNs certificates to FCM iOS app

Firebase console, choose the project where the iOS app was added

Project settings -> Cloud Messaging tab -> Apple app configuration, choose the iOS app just added -> upload APNs certificates (p12 files) for both production and development



////////

React Native App

In Terminal,

% yarn add @react-native-firebase/app @react-native-firebase/messaging

% yarn pod

Note that `yarn pod` won't be recognized by default, you need to add a yarn command line script to your React Native app's package.json:

```
  "scripts": {
    ...
    "pod": "pod install --project-directory=ios",
    ...
  },
```

Add the following code to your React Native app to (1) initialize Firebase, (2) request user permission, (3) get FCM token

```
import Firebase from '@react-native-firebase/app';
import messaging, {
  FirebaseMessagingTypes,
} from '@react-native-firebase/messaging';

export const initializeFirebase = async () => {
  if (!Firebase.app()) {
    const apiKey = '<apiKey>'; // For ios, GoogleService-Info.plist, API_KEY field; For Android, google-services.json, api_key.current_key field -- be careful to choose the correct client when multiple Android apps were added to your Firebase project
    const appId = '<appId>'; // For ios, GoogleService-Info.plist, GOOGLE_APP_ID; For Android, google-services.json, client_info.mobilesdk_app_id -- be careful to choose the correct client when multiple Android apps were added to your Firebase project
    const messagingSenderId = '<messagingSenderId>'; // Firebase console, Project number, a 12-digit number
    const projectId = '<projectId>'; // Firebase console, Project ID, "<nameoftheproject>"
    await Firebase.initializeApp({
      apiKey,
      appId,
      databaseURL: '',
      messagingSenderId,
      projectId,
      storageBucket: '',
    });
  }
};
```

```
export function requestUserPermission(): Promise<
  [boolean, FirebaseMessagingTypes.AuthorizationStatus]
> {
  return new Promise((resolve, reject) => {
    messaging()
      .requestPermission()
      .then(authorizationStatus => {
        console.log(
          'messaging().requestPermission() status:',
          authorizationStatus,
        );
        resolve([
          authorizationStatus ===
            FirebaseMessagingTypes.AuthorizationStatus.AUTHORIZED,
          authorizationStatus,
        ]);
      })
      .catch(reason => {
        console.log(
          'messaging().requestPermission() failed. reason:',
          JSON.stringify(reason),
        );
        reject(reason);
      });
  });
}

export function getCloudMessagingToken(): Promise<string> {
  return new Promise((resolve, reject) => {
    messaging()
      .getToken()
      .then(token => {
        console.log('messaging().getToken()', token);
        resolve(token);
      })
      .catch(reason => {
        console.log(
          'messaging().getToken() failed. reason:',
          JSON.stringify(reason),
        );
        reject(reason);
      });
  });
}
```

Run the app on a device and get the token (got by getCloudMessagingToken from above code) and send it to your Node server. Your Node server can use this token to send push notifications via FCM to the device.

//

NODE SERVER

//////////////////////////

Follow the doc in subfolder “ref”
  Sending Firebase Cloud Messages from a Node.js Server - Techotopia.pdf

//////////////////////////
//