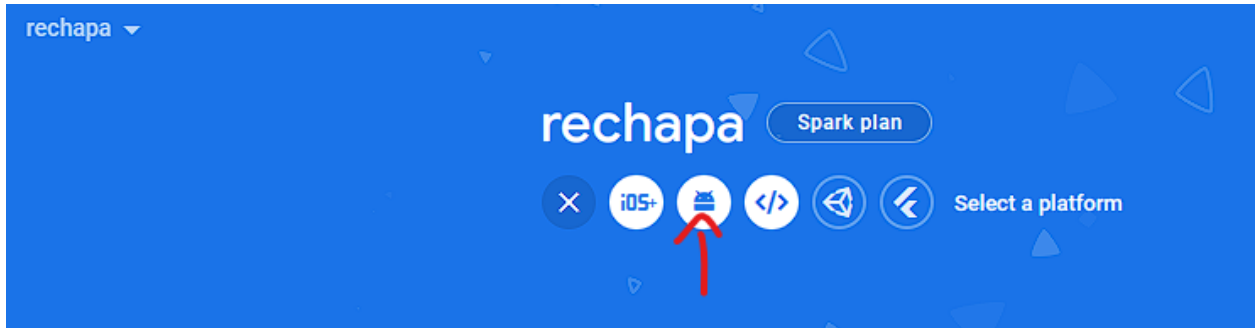


Firestore Phone number auth

- **Create firebase Account**
- **Add project**
- **Add App**



- **Enter package name**
- **Go to vs code**
- **Android\app\build.gradle**

```
}  
  
defaultConfig {  
    // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).  
    applicationId "com.example.phone_opy"  
    // You can update the following values to match your application needs.  
    // For more information, see: https://docs.flutter.dev/deployment/android#reviewing-the-gradle-build-configuration  
    minSdkVersion 33  
    targetSdkVersion flutter.targetSdkVersion  
    versionCode flutterVersionCode.toInteger()  
    versionName flutterVersionName  
}
```

- **Package name**

[illegible]

- Register app
- Download json

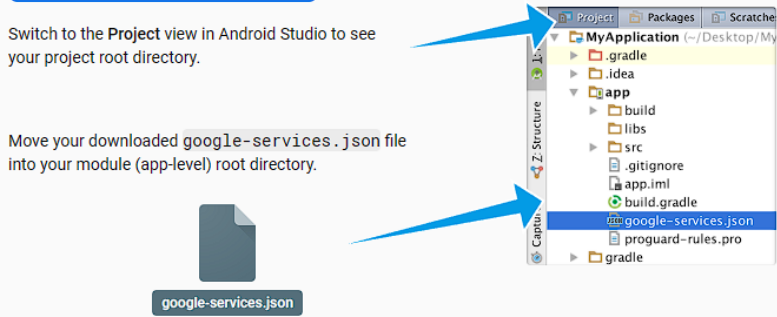
✓ Register app
Android package name: com.example.phone_opy

2 Download and then add config file [Instructions for Android Studio below](#) | [Unity](#) [C++](#)

[Download google-services.json](#)

Switch to the Project view in Android Studio to see your project root directory.

Move your downloaded google-services.json file into your module (app-level) root directory.



google-services.json

Next

3 Add Firebase SDK

4 Next steps

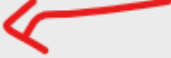


- **Past in project android**
- **Past project build.gradle**


```
buildscript {  
    repositories {  
        // Make sure that you have the following two repositories  
        google() // Google's Maven repository  
        mavenCentral() // Maven Central repository  
    }  
    dependencies {  
        ...  
        // Add the dependency for the Google services Gradle plugin  
        classpath 'com.google.gms:google-services:4.3.15'  
    }  
}  
  
allprojects {  
    ...  
    repositories {  
        // Make sure that you have the following two repositories  
        google() // Google's Maven repository  
        mavenCentral() // Maven Central repository  
    }  
}
```

- **Past project App-module/build.gradle**

☒ Kotlin ☐ Java

Module (app-level) Gradle file (<project>/<app-module>/build.gradle):

```
plugins {  
    id 'com.android.application'  
    // Add the Google services Gradle plugin  
    id 'com.google.gms.google-services'   
    ...  
}  
  
dependencies {  
    // Import the Firebase BoM  
    implementation platform('com.google.firebase:firebase-bom:32.0.0')   
  
    // TODO: Add the dependencies for Firebase products you want to use  
    // When using the BoM, don't specify versions in Firebase dependencies  
    implementation 'com.google.firebase:firebase-analytics-ktx'   
  
    // Add the dependencies for any other desired Firebase products  
    // https://firebase.google.com/docs/android/setup#available-libraries  
}
```

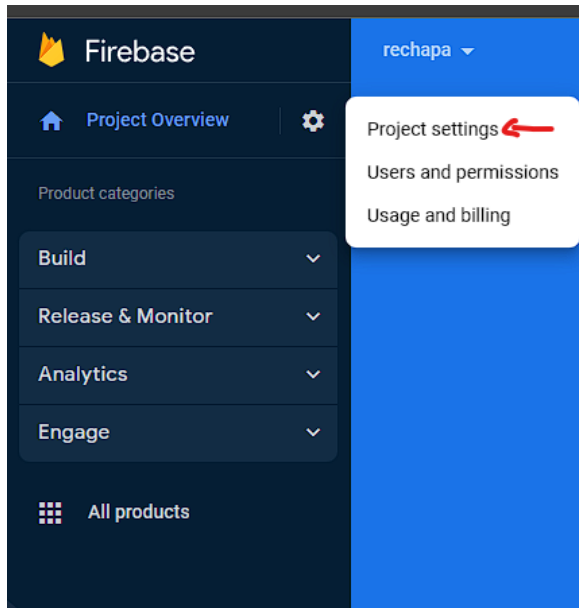
By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#) 

3. After adding the plugin and the desired SDKs, sync your Android project with Gradle files.

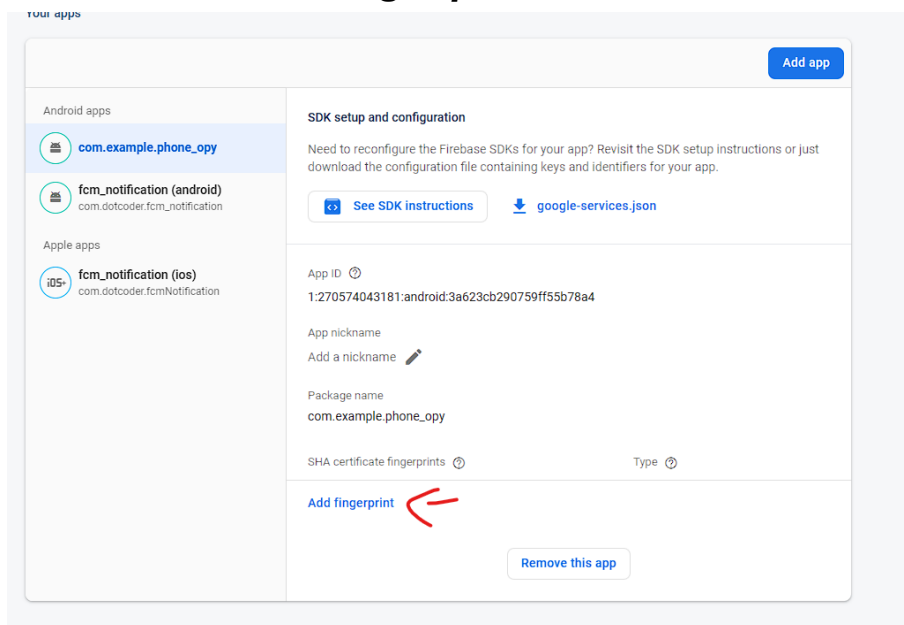
[Previous](#)

[Next](#)

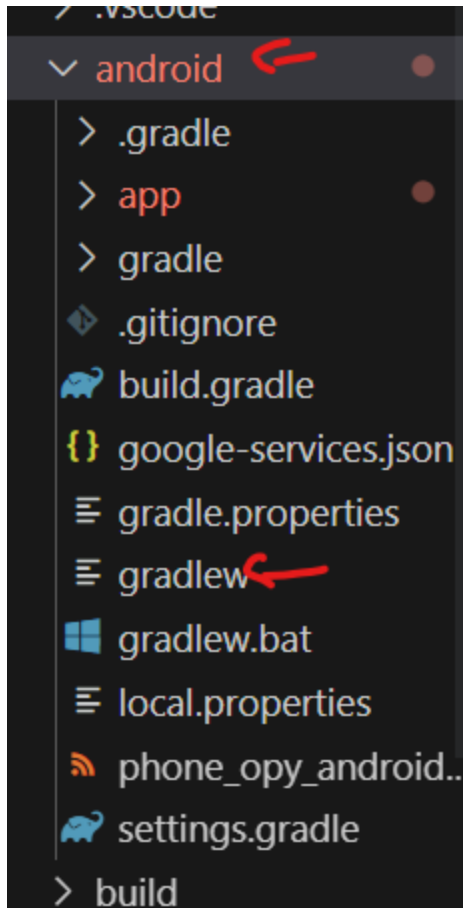
- **Open project setting**



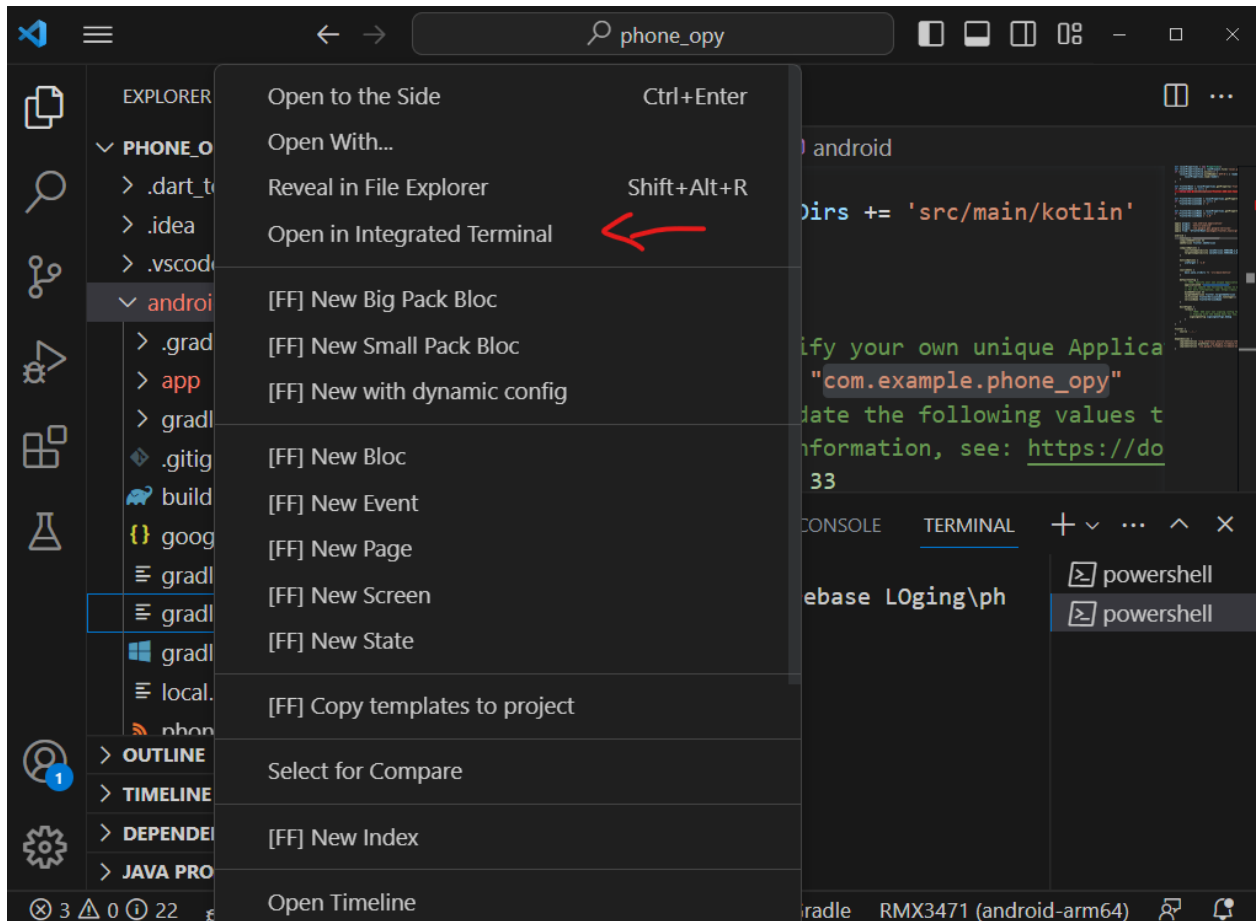
-
- **Scroll down Add finger print**



-
- **Go vs code**
- **Android / gradlew**



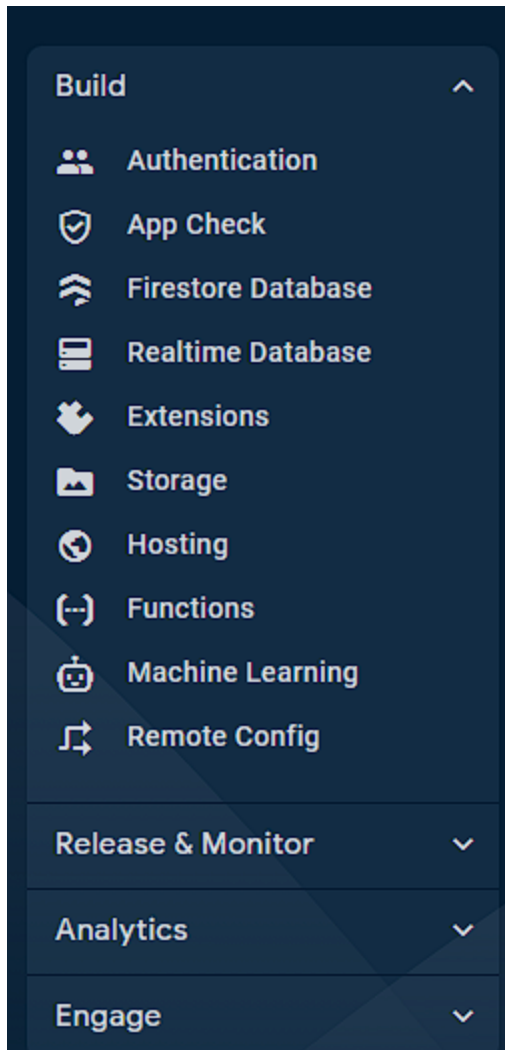
-
- ***Click right button in gradlew***



- Open terminal
- Run command
- ./gradlew signingReport
- Generate code

```
> Task :fluttertoast:signingReport
Variant: debugAndroidTest
Config: debug
Store: C:\Users\Asus\.android\debug.keystore
Alias: AndroidDebugKey
MD5: 46:0A:6B:75:3F:FB:C9:A1:4E:82:11:0A:B4:16:5E:61
SHA1: 28:05:83:87:6D:06:9F:20:23:12:7A:59:FF:DC:C1:77:F8:FD:C4:35
SHA-256: 36:35:00:16:29:0C:D2:C3:BE:53:93:9A:2B:3D:D9:43:0F:D2:18:61:06:C7:49:2F:27:4D:54:D5:F9:EF:A4:4B
Valid until: Saturday, 15 March, 2053
-----
```

- Go fire base
- Go to build



-
- Authenticatio
- Signing method

Authentication

Users [Sign-in method](#) Templates Usage Settings [Extensions](#) **NEW**

Sign-in providers

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

Native providers

Email/Password

Phone

Anonymous

Additional providers

Google

Game Center

Microsoft

Facebook

Apple

Twitter

Play Games

GitHub

Yahoo

Custom providers

OpenID Connect

SAML

Advanced

-
- Select phone

Authentication

Users [Sign-in method](#) Templates Usage Settings [Extensions](#) **NEW**

Sign-in providers

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

Native providers

Email/Password

Phone

Anonymous

Additional providers

Google

Game Center

Microsoft

Facebook

Apple

Twitter

Play Games

GitHub

Yahoo

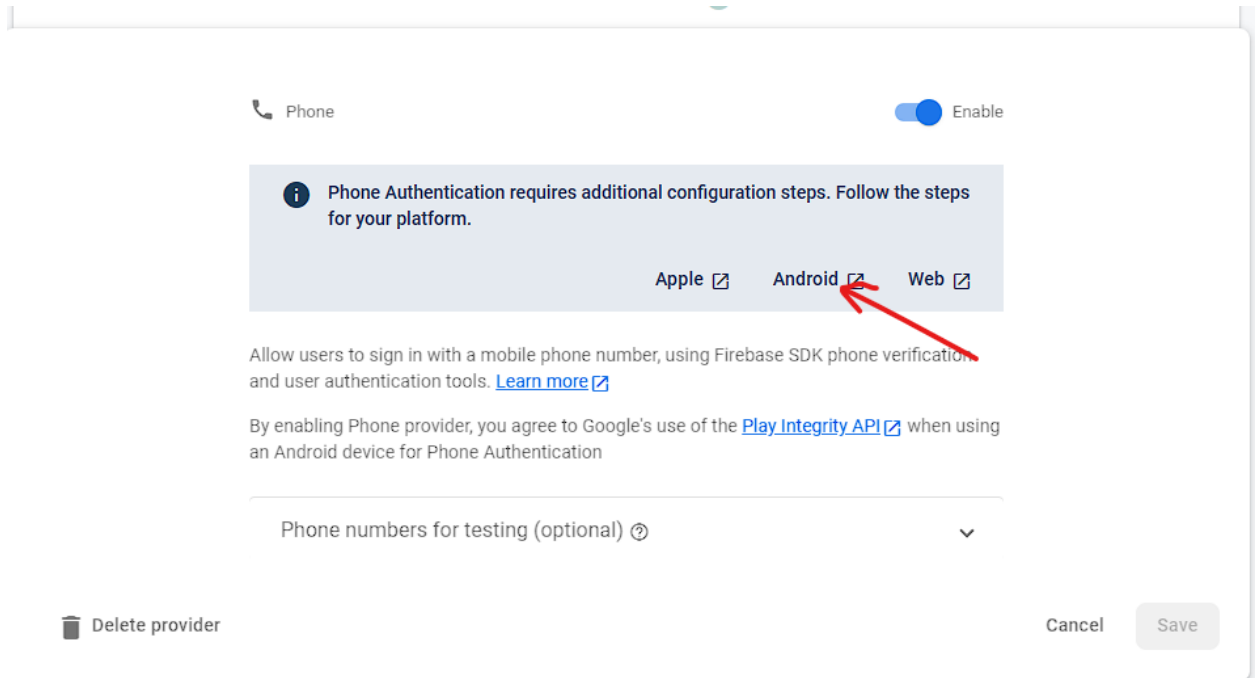
Custom providers

OpenID Connect

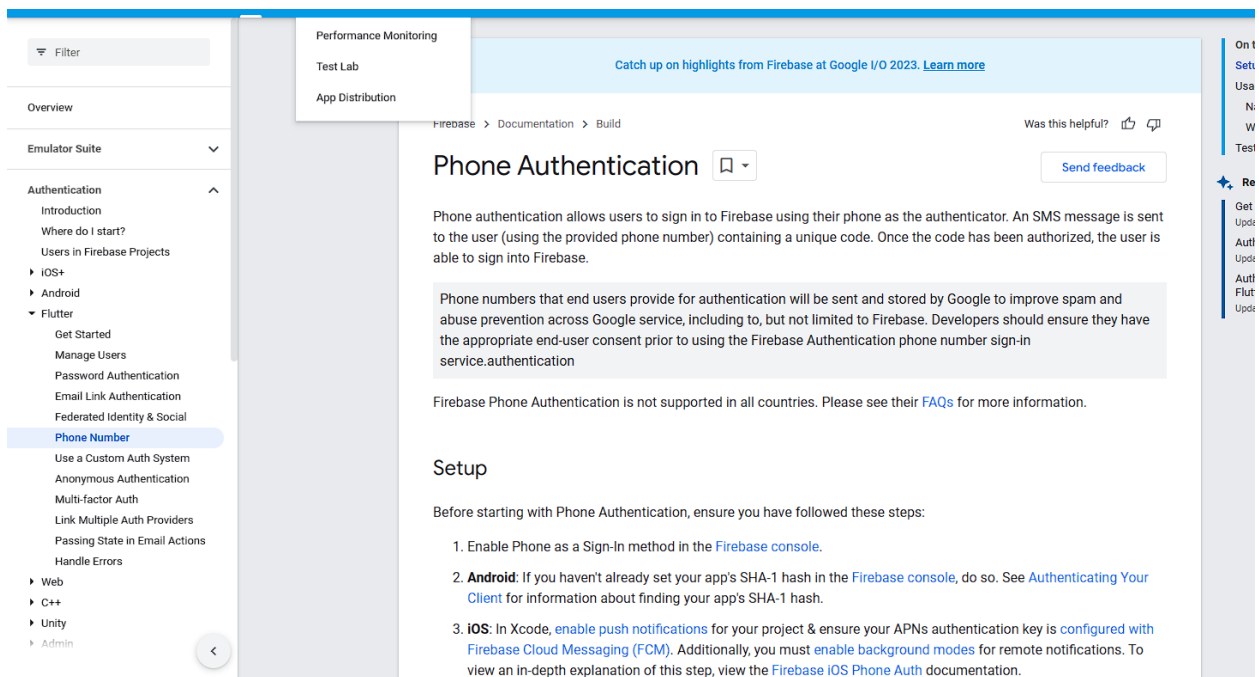
SAML

Advanced

-
- Click android



-
- *Go to this page*



-
- *Click flutter phone number*

Performance Monitoring
Test Lab
App Distribution

Catch up on highlights from Firebase at Google I/O 2023. [Learn more](#)

Was this helpful? [Send feedback](#)

Phone Authentication

Phone authentication allows users to sign in to Firebase using their phone as the authenticator. An SMS message is sent to the user (using the provided phone number) containing a unique code. Once the code has been authorized, the user is able to sign into Firebase.

Phone numbers that end users provide for authentication will be sent and stored by Google to improve spam and abuse prevention across Google service, including to, but not limited to Firebase. Developers should ensure they have the appropriate end-user consent prior to using the Firebase Authentication phone number sign-in service.authentication

Firebase Phone Authentication is not supported in all countries. Please see their [FAQs](#) for more information.

Setup

Before starting with Phone Authentication, ensure you have followed these steps:

1. Enable Phone as a Sign-In method in the [Firebase console](#).
2. **Android:** If you haven't already set your app's SHA-1 hash in the [Firebase console](#), do so. See [Authenticating Your Client](#) for information about finding your app's SHA-1 hash.
3. **iOS:** In Xcode, [enable push notifications](#) for your project & ensure your APNs authentication key is [configured with Firebase Cloud Messaging \(FCM\)](#). Additionally, you must [enable background modes](#) for remote notifications. To view an in-depth explanation of this step, view the [Firebase iOS Phone Auth](#) documentation.

-
- Add this line

```
await FirebaseAuth.instance.verifyPhoneNumber(
  phoneNumber: '+44 7123 123 456',
  verificationCompleted: (PhoneAuthCredential credential) {},
  verificationFailed: (FirebaseAuthException e) {},
  codeSent: (String verificationId, int? resendToken) {},
  codeAutoRetrievalTimeout: (String verificationId) {},
);
```

-
- Change phone number

```
await FirebaseAuth.instance.verifyPhoneNumber(
  → phoneNumber: '+44 7123 123 456',
  verificationCompleted: (PhoneAuthCredential credential) {},
  verificationFailed: (FirebaseAuthException e) {},
  codeSent: (String verificationId, int? resendToken) {},
  codeAutoRetrievalTimeout: (String verificationId) {},
);
```

-
- phoneNumber: '\${countrycontroller.text} + phone}'

