# Step-by-Step Guide: Google Sign-In with Firebase in Android Studio Using Kotlin

Securely implement Google Sign-In with Firebase in Android Studio using Kotlin with our step-by-step guide for seamless user authentication.

**CONTENTS PAGE**

Heading Page

[**MAIN HEADING** 2](#_Toc167151920)

[**Sub-Heading** 2](#_Toc167151921)

[**Sub-Heading** 2](#_Toc167151922)

[**MAIN HEADING** 2](#_Toc167151923)

[**Sub-Heading** 2](#_Toc167151924)

[**Sub-Heading** 2](#_Toc167151925)

[REFERENCES 3](#_Toc167151926)

[**Bibliography** 3](#_Toc167151927)

**Introduction:**

Google Sign-In is a quick and secure way to allow users to sign in to your Android app using their Google accounts. When combined with Firebase Authentication, it provides a simple and powerful solution for handling user authentication in your app. In this tutorial, we will explore how to implement Google Sign-In with Firebase in Android Studio using Kotlin.

**Step 1: Set Up Firebase Authentication**

The first step is to set up Firebase Authentication in your Android Studio project. Create a new Firebase project and add Firebase Authentication to it. Once you have done that, download and add the Firebase SDK to your project.

**Step 2: Configure Google Sign-In in Firebase Console**

The next step is to configure Google Sign-In in the Firebase Console. Go to the Firebase Console, select your project, and navigate to the "Authentication" section. Under the "Sign-in method" tab, enable the Google sign-in provider and add your app's SHA-1 key. Add the google-services.json file to your project by downloading it from the Firebase Console and placing it in the app directory of your project.

**Step 3: Add the following dependency to the build.gradle file.**

**build.gradle**

# **MAIN HEADING**

### STEP 1:

Da

### STEP 2:

Da

### STEP 3:

**build.gradle**

implementation platform('com.google.firebase:firebase-bom:31.5.0')  
implementation 'com.google.firebase:firebase-auth-ktx'  
implementation 'com.google.android.gms:play-services-auth:20.5.0'

### STEP 4:

Da

### STEP 5:

Da **Step 5**: Add the following code in the **activity\_main**.**xml** file.

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Loading.."  
 android:textSize="30dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.212" />  
  
 <Button  
 android:id="@+id/logout\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="132dp"  
 android:text="logout"  
 android:background="@color/red"  
 android:textColor="@color/white"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/name" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

### STEP 6:

**Step 6**: Create new empty activity, name it **SignInActivity.kt,**and Add the following code to it**.**

**activity\_sign\_in.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp"  
 tools:context=".SignInActivity">  
  
  
 <LinearLayout  
 android:layout\_centerInParent="true"  
 android:layout\_marginTop="50dp"  
 android:background="@drawable/border"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:layout\_width="28dp"  
 android:layout\_height="28dp"  
 android:layout\_alignParentLeft="true"  
 android:layout\_centerVertical="true"  
 android:layout\_gravity="center"  
 android:layout\_marginLeft="11dp"  
 android:layout\_marginRight="11dp"  
 android:background="@color/white"  
 android:src="@drawable/ic\_google" />  
  
 <Button  
 android:id="@+id/signInButton"  
 android:textAllCaps="false"  
 android:layout\_width="200dp"  
 android:layout\_height="50dp"  
 android:background="@color/blue"  
 android:text="Sign up with Google"  
 android:textColor="@color/white"  
 android:textSize="20sp" />  
  
 </LinearLayout>  
  
  
  
</RelativeLayout>

### STEP 7:

import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.Button  
import android.widget.TextView  
import com.google.android.gms.auth.api.signin.GoogleSignIn  
import com.google.android.gms.auth.api.signin.GoogleSignInClient  
import com.google.android.gms.auth.api.signin.GoogleSignInOptions  
import com.google.firebase.auth.FirebaseAuth  
import com.google.firebase.auth.ktx.auth  
import com.google.firebase.ktx.Firebase  
  
class MainActivity : AppCompatActivity() {  
  
 private lateinit var mGoogleSignInClient: GoogleSignInClient  
 private lateinit var mAuth: FirebaseAuth  
  
  
  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 mAuth = FirebaseAuth.getInstance()  
  
  
 val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.*DEFAULT\_SIGN\_IN*)  
 .requestIdToken(getString(R.string.*default\_web\_client\_id*))  
 .requestEmail()  
 .build()  
  
 mGoogleSignInClient = GoogleSignIn.getClient(this, gso)  
  
  
  
 val textView = findViewById<TextView>(R.id.*name*)  
  
 val auth = Firebase.*auth* val user = auth.*currentUser* if (user != null) {  
 val userName = user.*displayName* textView.*text* = "Welcome, " + userName  
 } else {  
 // Handle the case where the user is not signed in  
 }  
  
  
  
// Inside onCreate() method  
 val sign\_out\_button = findViewById<Button>(R.id.*logout\_button*)  
 sign\_out\_button.setOnClickListener **{** signOutAndStartSignInActivity()  
 **}** }  
  
  
 private fun signOutAndStartSignInActivity() {  
 mAuth.signOut()  
  
 mGoogleSignInClient.signOut().addOnCompleteListener(this) **{** // Optional: Update UI or show a message to the user  
 val intent = Intent(this@MainActivity, SignInActivity::class.*java*)  
 startActivity(intent)  
 finish()  
 **}** }  
}

### STEP 8:

**Step 8**: Add the following code in the **SignInActivity.kt** file.

**SignInActivity.kt**

import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.Button  
import android.widget.LinearLayout  
import android.widget.Toast  
import com.google.android.gms.auth.api.signin.GoogleSignIn  
import com.google.android.gms.auth.api.signin.GoogleSignInOptions  
import com.google.android.gms.common.api.ApiException  
import com.google.firebase.auth.FirebaseAuth  
import com.google.firebase.auth.GoogleAuthProvider  
  
class SignInActivity : AppCompatActivity() {  
  
  
 companion object {  
 private const val RC\_SIGN\_IN = 9001  
 }  
  
 private lateinit var auth: FirebaseAuth  
  
  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_sign\_in*)  
  
 auth = FirebaseAuth.getInstance()  
  
  
  
 val currentUser = auth.*currentUser* if (currentUser != null) {  
 // The user is already signed in, navigate to MainActivity  
 val intent = Intent(this, MainActivity::class.*java*)  
 startActivity(intent)  
 finish() // finish the current activity to prevent the user from coming back to the SignInActivity using the back button  
 }  
  
  
  
  
 val signInButton = findViewById<Button>(R.id.*signInButton*)  
 signInButton.setOnClickListener **{** signIn()  
 **}** }  
  
 private fun signIn() {  
 val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.*DEFAULT\_SIGN\_IN*)  
 .requestIdToken(getString(R.string.*default\_web\_client\_id*))  
 .requestEmail()  
 .build()  
  
 val googleSignInClient = GoogleSignIn.getClient(this, gso)  
 val signInIntent = googleSignInClient.*signInIntent* startActivityForResult(signInIntent, RC\_SIGN\_IN)  
 }  
  
 override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
 super.onActivityResult(requestCode, resultCode, data)  
  
 if (requestCode == RC\_SIGN\_IN) {  
 val task = GoogleSignIn.getSignedInAccountFromIntent(data)  
 try {  
 val account = task.getResult(ApiException::class.*java*)  
 firebaseAuthWithGoogle(account.*idToken*!!)  
 } catch (e: ApiException) {  
 Toast.makeText(this, "Google sign in failed: ${e.message}", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 }  
  
 private fun firebaseAuthWithGoogle(idToken: String) {  
 val credential = GoogleAuthProvider.getCredential(idToken, null)  
 auth.signInWithCredential(credential)  
 .addOnCompleteListener(this) **{** task **->** if (task.*isSuccessful*) {  
 val user = auth.*currentUser* Toast.makeText(this, "Signed in as ${user?.*displayName*}", Toast.*LENGTH\_SHORT*).show()  
 startActivity(Intent(this, MainActivity::class.*java*))  
 finish()  
 } else {  
 Toast.makeText(this, "Authentication failed", Toast.*LENGTH\_SHORT*).show()  
 }  
 **}** }  
}

### STEP 9:

**Step 9**: Add the following theme to the **themes.xml** file.

**themes.xml**

 Theme.AppCompat.DayNight.NoActionBar

### STEP 10:

**Step 10**: In the **Manifest**file, replace **MainActivity**with **SigninActivity**and replace **SigninActivity**with **MainAcivity**. see the following example code.

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MyApplication648"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="false" />  
 <activity  
 android:name=".SignInActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

### STEP 10:

### STEP 10:

### STEP 10:

### STEP 10: