

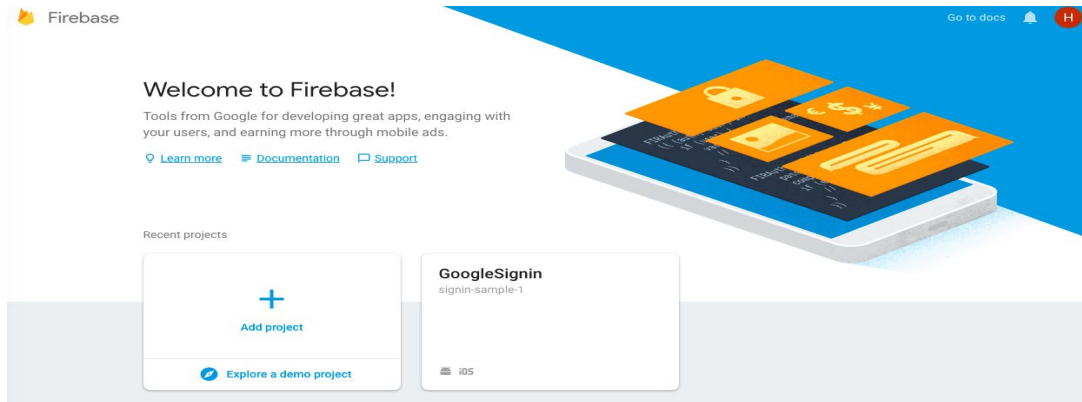
React-Native-Google-SignIn

Installation:-

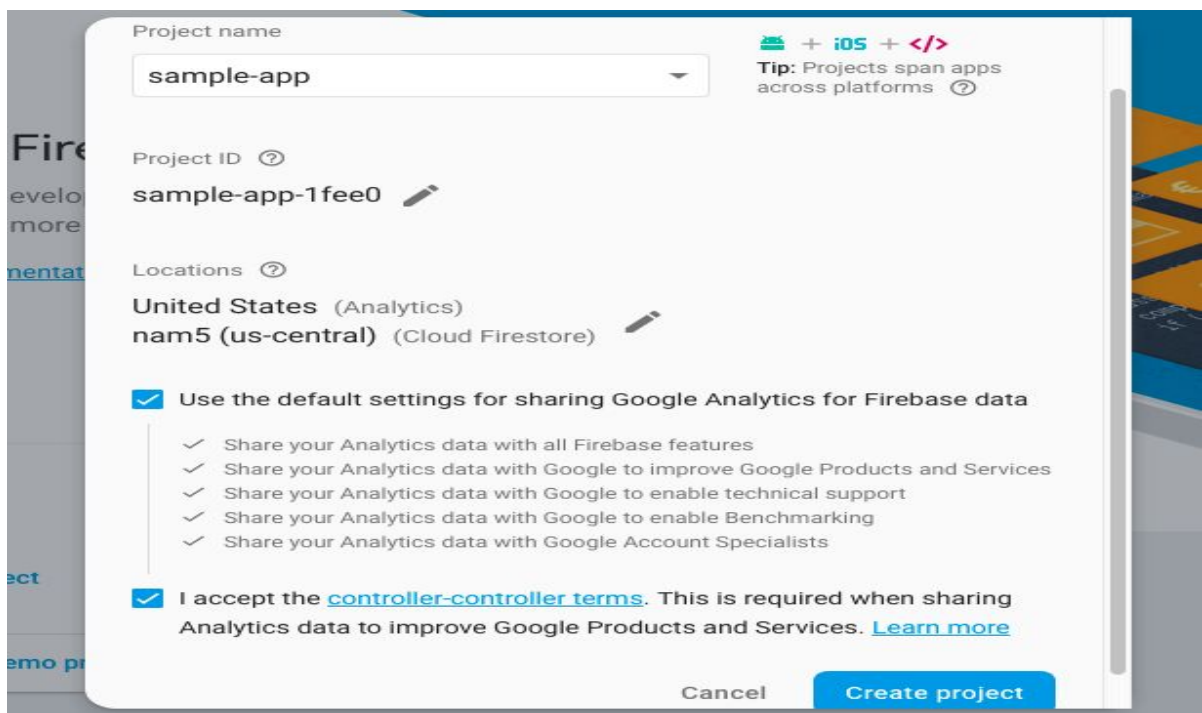
→ npm install react-native-google-signin

Create a GoogleService-Info.plist file (iOS) and google-services.json file (Android):-

- Open firebase using [this](#) link.
- To click add button for creating a new project in below mentioned image.





- After clicking add button the image should be like this:-



To fill the project details in above pop-up image.




- After creating a project

Your project

Project name	sample 
Project ID 	sample-67adf
Cloud Firestore location	nam5 (us-central)
Web API Key	AlzaSyBBSxGNIU1-hCB7gVFAYL6qmFqL-GZi6JY

Public settings





These settings control instances of your project shown to the public

Public-facing name 	project-264785984347 
Support email 	<input type="text" value="harika.n@stellentsoft.com"/>

Your apps


There are no apps in your project


Select a platform to get started




- When clicks on iOS button:-

1 Register app

iOS bundle ID 

App nickname (optional) 

App Store ID (optional) 

2 Download config file

3 Add Firebase SDK

4 Add initialization code

5 Run your app to verify installation

➤ iOS bundle ID:- To give your bundle identifier which is used in ios app.

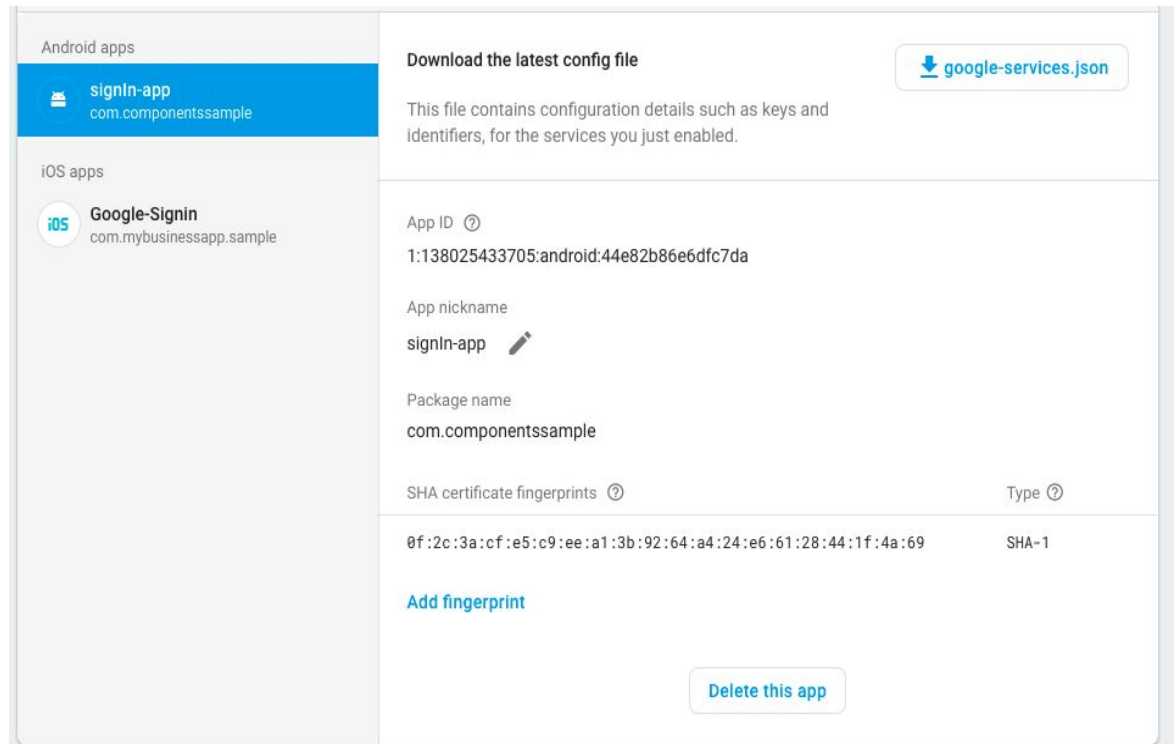
- After filling the details **GoogleService-Info.plist** file is generated like this

The screenshot shows the Firebase console interface for an app named 'Google-Signin' (com.mybusinessapp.sample). On the left, a sidebar lists 'Android apps' (signIn-app) and 'iOS apps' (Google-Signin). The main panel is titled 'Download the latest config file' and contains a 'Download GoogleService-Info.plist' button. Below this, it states: 'This file contains configuration details such as keys and identifiers, for the services you just enabled.' The configuration details listed are: App ID (1:138025433705:ios:bf939ff2feb6f67c), App nickname (Google-Signin), Bundle ID (com.mybusinessapp.sample), App Store ID, Add an App Store ID, Team ID, and Add a Team ID. A 'Delete this app' button is at the bottom right.

- When clicks on Android

The screenshot shows the '1 Register app' step in the Firebase console. It includes a vertical progress bar on the left with steps 1 through 4. The main form has three input fields: 'Android package name' (containing 'com.company.appname'), 'App nickname (optional)' (containing 'Freemium Android App'), and 'Debug signing certificate SHA-1 (optional)' (containing a long hexadecimal string). Below the fields is a 'Register app' button. At the bottom, a note states: 'Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.'

- Android package name:- To give your package name which is placed in **ProjectFolder/android/app/src/main/AndroidManifest.xml** file
ex:- `package="com.componentssample"`
- To generate keystore file please follow [this](#)
- To generate a SHA-1 key by using this command:-
keystores folder:- `keytool -J-Duser.language=en -list -v -keystore [keystore_name]`
- After filling the details **google-services.json** file is generated

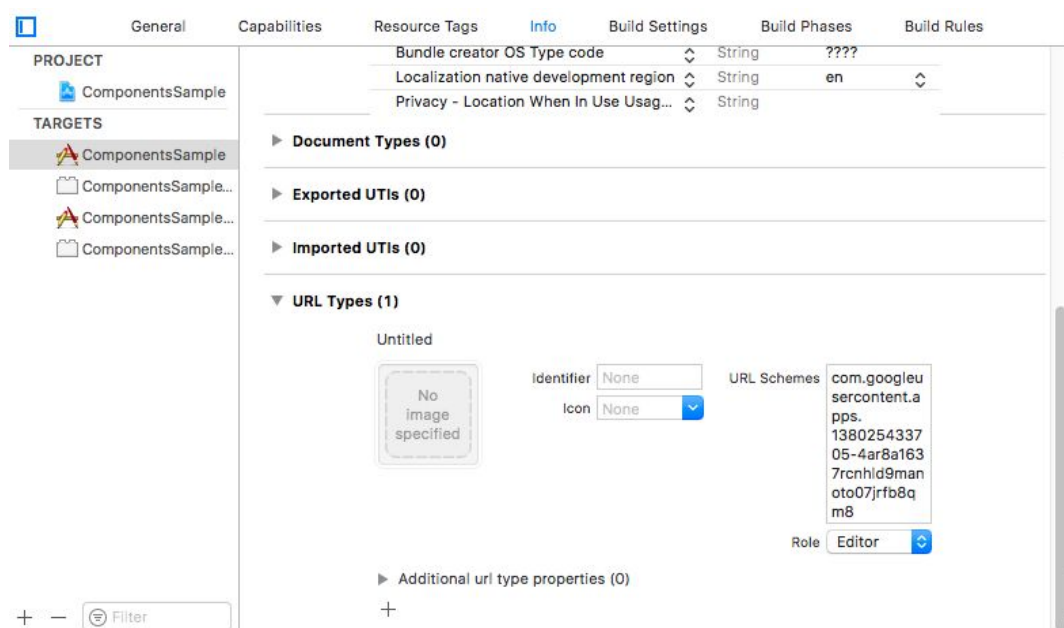


Project setup and initialization:-

iOS:-

- **Manual (Without Cocoapods):-**
 - In XCode, in the project navigator, right click Libraries → Add Files to [your project's name]
 - Go to node_modules → react-native-google-signin and add RNGoogleSignin.xcodeproj
 - In XCode, in the project navigator, select your project. Add libRNGoogleSignin.a to your project's Build Phases → Link Binary With Libraries

- Download the GoogleSignIn SDK from [here](#) and unzip it. Drag and drop the unzipped .framework files into the Frameworks group in Xcode and copy GoogleSignIn.bundle to your project. During copying, check copy items if needed.
- Make sure GoogleSignIn.bundle is added in your Xcode project's Copy Bundle Resources build phase.
- According to google sign in docs you may also need to do this: In -> Build Phases -> Link binary with libraries step, add libRNGoogleSignIn.a, AddressBook.framework, SafariServices.framework, SystemConfiguration.framework and libz.tbd. We have found it not to be necessary for a successful build, but we recommend to follow Google's installation instructions!
- Follow [this](#) guide to get the configuration file
If you don't already have a project in Firebase you need to create one in order to generate credentials for an iOS and Android application. Please follow [this](#) to create a **GoogleService-Info.plist** file in iOS and **google-services.json** file in Android.
- Download the GoogleService-Info.plist file at the end of the process.
- Configure URL types in the **Info** panel. Add a URL with scheme set to your **REVERSED_CLIENT_ID** (found inside GoogleService-Info.plist). Please follow below image.



- Open **AppDelegate.m** file in **Project/ios/ProjectName** and please follow below code.

Import this line:- `#import <RNGoogleSignin/RNGoogleSignin.h>`

Placed this code before **@end** in **AppDelegate.m** file

```
- (BOOL)application:(UIApplication *)application openURL:(nonnull
NSURL *)url options:(nonnull NSDictionary<NSString *,id> *)options {
    return [RNGoogleSignin application:application
                                openURL:url
                                sourceApplication:options[UIApplicationOpenURLOptionsSourceApplicationKey]
                                annotation:options[UIApplicationOpenURLOptionsAnnotationKey]];
}
```

- **With Cocoapods:-**

- Go to project folder → **cd ios**
- To create a pod file in ios folder:- **pod init**
- Open the pod file and add this line :- **pod 'GoogleSignIn'**
- Install the pods using this command:- **pod install**
- Make sure after installing the pods please open the **.xcworkspace** file in iOS folder.

Android:-

- Update **android/build.gradle** with

```
buildscript {
    ext {
        buildToolsVersion = "27.0.3"
        minSdkVersion = 16
        compileSdkVersion = 27
        targetSdkVersion = 26
        supportLibVersion = "27.1.1"
        googlePlayServicesAuthVersion = "15.0.1" // <--- use this version or newer
    }
    ...
    dependencies {
        classpath 'com.android.tools.build:gradle:3.1.2' // <--- use this version or newer
        classpath 'com.google.gms:google-services:3.2.1' // <--- use this version or newer
    }
    ...
    allprojects {
        repositories {
            mavenLocal()
            google() // <--- make sure this is included
            jcenter()
            maven {
                // All of React Native (JS, Obj-C sources, Android binaries) is installed from npm
                url "$rootDir/../node_modules/react-native/android"
            }
        }
    }
}
```

- Update **android/app/build.gradle** with

```
dependencies {
    implementation fileTree(dir: "libs", include: ["*.jar"])
    implementation "com.android.support:appcompat-v7:23.0.1"
    implementation "com.facebook.react:react-native:+"
    implementation(project(":react-native-google-signin")) → this line
}
```

- In **android/settings.gradle** you should add

```
include ':react-native-google-signin', ':app'
project(':react-native-google-signin').projectDir = new File(rootProject.projectDir,
'../node_modules/react-native-google-signin/android')
```

- In **MainApplication.java** you should add

```
import co.apptailor.googlelogin.RNGoogleSignInPackage; // <--- import
public class MainApplication extends Application implements ReactApplication {
    .....
    @Override
    protected List<ReactPackage> getPackages() {
        return Arrays.<ReactPackage>asList(
            new MainReactPackage(),
            new RNGoogleSignInPackage() // <-- this needs to be in the list
        );
    }
    .....
}
```

Code to execute:-

Google Signin Button:-

```
import { GoogleSignIn, GoogleSignInButton } from 'react-native-google-signin';

render() {
    <GoogleSignInButton
        style={{ width: 192, height: 48 }}
        size={GoogleSignInButton.Size.Wide}
        color={GoogleSignInButton.Color.Dark}
        onPress={this._signIn}
        disabled={this.state.isSignInInProgress} />
}
```


Placed the Web-Client-Id and iOS-Client-Id:-

```
GoogleSignin.configure({
  iosClientId:"<FROM DEVELOPER CONSOLE>",
  webClientId:"<FROM DEVELOPER CONSOLE>"
});
```

Getting the signin user details:-

```
// import statusCodes along with GoogleSignin
import { GoogleSignin, statusCodes } from 'react-native-google-signin';

// Somewhere in your code
signIn = async () => {
  try {
    await GoogleSignin.hasPlayServices();
    const userInfo = await GoogleSignin.signIn();
    this.setState({ userInfo });
  } catch (error) {
    if (error.code === statusCodes.SIGN_IN_CANCELLED) {
      // user cancelled the login flow
    } else if (error.code === statusCodes.IN_PROGRESS) {
      // operation (f.e. sign in) is in progress already
    } else if (error.code === statusCodes.PLAY_SERVICES_NOT_AVAILABLE) {
      // play services not available or outdated
    } else {
      // some other error happened
    }
  }
};
```

For signOut:-

```
signOut = async () => {
  try {
    await GoogleSignin.revokeAccess();
    await GoogleSignin.signOut();
    this.setState({ user: null }); // Remember to remove the user from your app's state as well
  } catch (error) {
    console.error(error);
  }
};
```

Referred Link:- <https://github.com/react-native-community/react-native-google-signin>