

To integrate Google Sign-In functionality into your React Native app, follow these steps:

Step 1: Install Dependencies

Run the following command in your project directory:

```
npm install @react-native-google-signin/google-signin
```

Step 2: Configure Google Sign-In

Modify your **LoginScreen.tsx** to include Google authentication logic.

1. Import Required Modules

At the top of **LoginScreen.tsx**, add:

```
import React, { useEffect } from 'react';
import { View, Text, TouchableOpacity, Alert } from 'react-native';
import { GoogleSignin, statusCodes } from '@react-native-google-signin/google-signin';
import auth from '@react-native-firebase/auth';

const LoginScreen = ({ navigation }) => {

  useEffect(() => {
    GoogleSignin.configure({
      webClientId: 'YOUR_WEB_CLIENT_ID.apps.googleusercontent.com', // Replace with your
      actual web client ID
    });
  }, []);

  const signInWithGoogle = async () => {
    try {
      await GoogleSignin.hasPlayServices();
      const { idToken } = await GoogleSignin.signIn();
      const googleCredential = auth.GoogleAuthProvider.credential(idToken);
      await auth().signInWithCredential(googleCredential);

      Alert.alert('Login Successful', 'You have signed in with Google!');
```

```

        navigation.navigate('Home'); // Redirect user to home after login
    } catch (error) {
        console.log(error);
        if (error.code === statusCodes.SIGN_IN_CANCELLED) {
            Alert.alert('Cancelled', 'Google Sign-In was cancelled');
        } else if (error.code === statusCodes.IN_PROGRESS) {
            Alert.alert('Error', 'Google Sign-In in progress');
        } else if (error.code === statusCodes.PLAY_SERVICES_NOT_AVAILABLE) {
            Alert.alert('Error', 'Google Play Services not available');
        } else {
            Alert.alert('Error', 'Something went wrong');
        }
    }
};

return (
    <View>
        <Text>Login Screen</Text>
        <TouchableOpacity onPress={signInWithGoogle} style={{ backgroundColor: 'blue', padding:
10, marginTop: 20 }}>
            <Text style={{ color: 'white', textAlign: 'center' }}>Sign in with Google</Text>
        </TouchableOpacity>
    </View>
);
};

export default LoginScreen;

```

Step 3: Add Firebase Setup

1. **Set up Firebase in your project** (if not done already) by going to [Firebase Console](#).
2. **Enable Google Sign-In** in Firebase Authentication.
3. Download **google-services.json** (for Android) and place it in **android/app/**.

Step 4: Configure Android Setup

Edit your **android/app/build.gradle**:

```
dependencies {
```

```
implementation 'com.google.firebase:firebase-auth'
implementation 'com.google.android.gms:play-services-auth:20.4.0'
}
```

Then, add this to **android/app/src/main/AndroidManifest.xml**:


```
<activity android:name="com.google.android.gms.auth.api.signin.internal.SignInHubActivity" />
```

Step 5: Run Your App

Run the project with:

```
npx react-native run-android
```

Now, when you click the **Sign in with Google** button, it will authenticate users and navigate to the **Home** screen.

Would you like assistance with setting up Firebase as well? 

DeepSeek –

To implement "Sign in with Google" functionality in a React Native CLI project, you can use the `@react-native-google-signin/google-signin` library. Below is a step-by-step guide to set it up:

Step 1: Install Dependencies

1. Install the `@react-native-google-signin/google-signin` package:

```
```bash
```

```
npm install @react-native-google-signin/google-signin
```

```
...
```

or

```
```bash
```

```
yarn add @react-native-google-signin/google-signin
```

```
...
```

2. For iOS, install the necessary pods:

```
```bash
```

```
cd ios && pod install && cd ..
```

```
...
```

```

```

### ### Step 2: Configure Google API Console

1. Go to the [Google API Console](<https://console.developers.google.com/>).
2. Create a new project (or use an existing one).
3. Enable the **Google Sign-In API** for your project.
4. Create OAuth 2.0 credentials:
  - Add your app's package name (e.g., `com.yourapp`).
  - Add your SHA-1 fingerprint (for Android) and bundle ID (for iOS).
  - Download the `google-services.json` file (for Android) and `GoogleService-Info.plist` file (for iOS).

```

```

### ### Step 3: Configure Android

1. Add the `google-services.json` file to your `android/app` directory.
2. Update `android/build.gradle`:

```
```gradle
buildscript {
    dependencies {
        classpath 'com.google.gms:google-services:4.3.15' // or latest version
    }
}
```
```

3. Update `android/app/build.gradle`:

```
```gradle
apply plugin: 'com.android.application'

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

dependencies {
    implementation project(':@react-native-google-signin_google-signin') // Add this line
}
```
```

---

### ### Step 4: Configure iOS

1. Add the `GoogleService-Info.plist` file to your iOS project in Xcode.

2. Update `ios/Podfile`:

```
``ruby

pod 'GoogleSignIn', '~> 7.0' // Add this line

...
```

3. Run `pod install` in the `ios` directory.

---

### ### Step 5: Implement Google Sign-In in React Native

1. Initialize the Google Sign-In library in your app (e.g., in `App.js` or a separate file):

```
``javascript

import React, { useEffect } from 'react';

import { Button, View, Text } from 'react-native';

import { GoogleSignin, GoogleSigninButton, statusCodes } from
'@react-native-google-signin/google-signin';

const App = () => {

 useEffect(() => {

 GoogleSignin.configure({

 scopes: ['email'], // optional, for additional scopes

 webClientId: 'YOUR_WEB_CLIENT_ID', // from Firebase console

 offlineAccess: true, // optional, for server-side access

 });

 }, []);

}
```

```
const signIn = async () => {
 try {
 await GoogleSignin.hasPlayServices();
 const userInfo = await GoogleSignin.signIn();
 console.log('User Info:', userInfo);
 } catch (error) {
 if (error.code === statusCodes.SIGN_IN_CANCELLED) {
 console.log('User cancelled the login flow');
 } else if (error.code === statusCodes.IN_PROGRESS) {
 console.log('Sign in is in progress already');
 } else if (error.code === statusCodes.PLAY_SERVICES_NOT_AVAILABLE) {
 console.log('Play services not available or outdated');
 } else {
 console.log('Error:', error);
 }
 }
};
```

```
const signOut = async () => {
 try {
 await GoogleSignin.signOut();
 console.log('User signed out');
 } catch (error) {
 console.log('Error:', error);
 }
};
```

```

 }
};

return (
 <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
 <GoogleSignInButton
 style={{ width: 192, height: 48 }}
 size={GoogleSignInButton.Size.Wide}
 color={GoogleSignInButton.Color.Dark}
 onPress={signIn}
 />
 <Button title="Sign Out" onPress={signOut} />
 </View>
);
};

export default App;
...

```

### ### Step 6: Test the Implementation

1. Run your app on an Android or iOS device/emulator:

```
```bash
```



```
npx react-native run-android
```

```
...
```

```
or
```

```
```bash
```

```
npx react-native run-ios
```

```
...
```

2. Test the Google Sign-In button and ensure it works as expected.

---

#### ### Notes

- Replace `YOUR\_WEB\_CLIENT\_ID` with your actual web client ID from the Google API Console.
- Ensure you have enabled the Google Sign-In API in the Google API Console.
- For iOS, make sure your bundle ID matches the one in the `GoogleService-Info.plist` file.

---

This setup should allow you to integrate Google Sign-In into your React Native CLI project. Let me know if you encounter any issues!