Social Media Logins:  
  
Google:

**Android**  
1- Go to this link  
<https://github.com/react-native-community/google-signin>  
on this page install this library with yarn not npm  
**->**yarn add @react-native-community/google-signin  
**->**then click on AndroidGuid go to this page (<https://github.com/react-native-community/google-signin/blob/master/docs/android-guide.md>

)  
->at next page click on ‘this’  
on clicking on this go to this page(<https://github.com/react-native-community/google-signin/blob/master/docs/get-config-file.md>

)

where go at section without firebase and android   
**->**and click on this [Configure a Google API Project](https://developers.google.com/identity/sign-in/android/start#configure-a-google-api-project)   
on clicking Cofigure a Google API Project come at this page <https://developers.google.com/identity/sign-in/android/start#configure-a-google-api-project>

**->**where click on button configure a project  
 create new project this project was creted on login Gmail  
 write name ,then same name next  
 select android   
 package name from androidmenifest.xml  
 go to cmd past this command and get SHA-1 Key and past here

**keytool -exportcert -keystore ./android/app/debug.keystore -list –v  
 password is android**

after creating click on button Google API Console where button congigure a project button  
at console select a project then copy web client id  
  
past this code at any file and change web client id

//import liraries

import React, { Component } from 'react';

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

import {

GoogleSignin,

GoogleSigninButton,

statusCodes,

} from '@react-native-community/google-signin';

// create a component

class firebaseconnection extends Component {

state = {

userInfo: null

}

componentDidMount() {

GoogleSignin.configure({

scopes: ['https://www.googleapis.com/auth/drive.readonly'], // what API you want to access on behalf of the user, default is email and profile

webClientId: '873298935315-7mgb2rt1aav0jin5s883r1uak652ee22.apps.googleusercontent.com', // client ID of type WEB for your server (needed to verify user ID and offline access)

offlineAccess: true,

// androidClientId: '1031142128277-abejg9uojbd3gv4bccph1qv32uutlp14.apps.googleusercontent.com'

// iosClientId:'1031142128277-sp8hu67m7gp8lij1j5oeuhklpo1mi4rd.apps.googleusercontent.com'

//1031142128277-et8l7gdjsrvrh8msefn4asl6uivoc8e7.apps.googleusercontent.com

//nc76CXBwpsEZ1\_0fbYaRGcb8

});

}

signIn = async () => {

try {

await GoogleSignin.hasPlayServices();

const userInfo = await GoogleSignin.signIn();

//this.setState({ userInfo });

console.log('user data',userInfo)

} catch (error) {

console.log(error.code,error.toString())

}

};

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

alert('signout')

} catch (error) {

console.error(error);

}

};

render() {

return (

<View style={styles.container}>

<Text onPress={() => this.signIn()}>Google Sign IN</Text>

<Text style={{marginTop:100}} onPress={() => this.signOut()}>Google Sign Out</Text>

</View>

);

}

}

// define your styles

const styles = StyleSheet.create({

container: {

flex: 1,

justifyContent: 'center',

alignItems: 'center',

backgroundColor: '#2c3e50',

},

});

//make this component available to the app

export default firebaseconnection;

All codes written at this page:  
<https://github.com/react-native-community/google-signin>

**Google IOS :**After completing the android.into this page go to IOS without firebase <https://github.com/react-native-community/google-signin/blob/master/docs/get-config-file.md>

For ios we have to do 2 steps.

First click on this [Get an OAuth client ID](https://developers.google.com/identity/sign-in/ios/start-integrating#get_an_oauth_client_id).

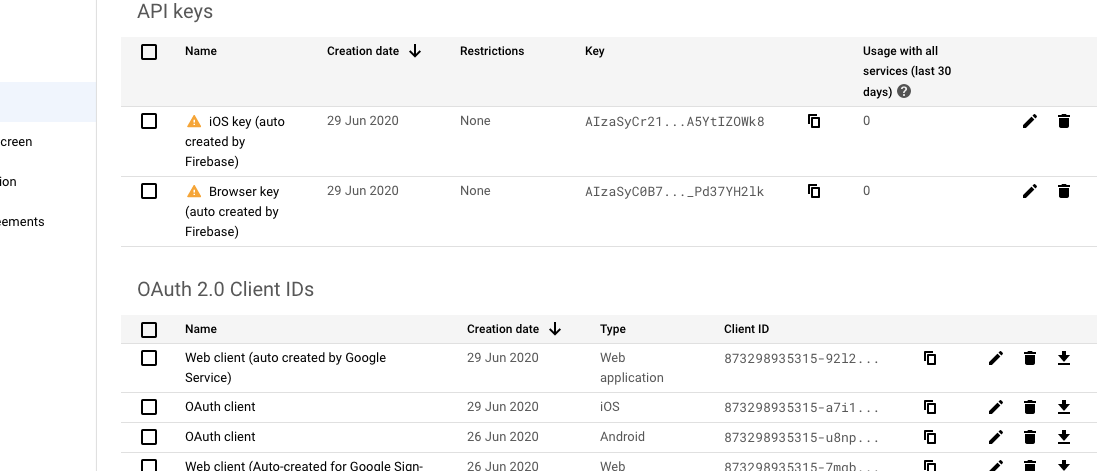
Click on button Create an Oath clint id  
Confire an project for Google Sign-in with IOS Similarly we do fro Android.

Select project from dropdown which is already used for android.  
selet for ios then done.  
google concole and check now Added API Keys section in which IOS Key

Now configure this project with Firebase. At firebase select the same name which are already in google . google developer console and firebase must use same project mean not write new name while we will selet already existing name this name will be from google console. Download GooGleService.Info.Plist and place into ios->project name near .plist file with XCODE.

In second step [Add a URL scheme to your project](https://developers.google.com/identity/sign-in/ios/start-integrating#add_a_url_scheme_to_your_project) click on this at page <https://github.com/react-native-community/google-signin/blob/master/docs/get-config-file.md>  
And add Add a URL scheme to your project.

For this open XCODE then project name and then at lower project name and then select Info Tab then url Type and here click on + button and add new url schema here copy and past only reverse id from GoogleServiceInfoPlist.

****

**Facebook:  
  
Android facebook:  
1-**You have to facebook developer option on

2- Start from this site <https://github.com/facebook/react-native-fbsdk>

-> here install npm install --save react-native-fbsdk

got to android section and click on [Getting Started Guide](https://developers.facebook.com/docs/android/getting-started/)

after clicking on this reach at this page <https://developers.facebook.com/docs/android/getting-started/>

where click on button Quick start for Android   
click on start over and select android   
here we are at facebook developer mood  
select any existing app or crate new one  
Now Follow below guide  
skip this **Add Facebook SDK to Your Project** and his 7 steps  
follow **Add Facebook App ID** guide  
package name from Androidmenifest.xml i.e com.island\_taste  
default class name below i.e com.island\_taste.MainActivity  
generate hashkey past this command into project cmd code with android

keytool -exportcert -alias androiddebugkey -keystore ~/.android/debug.keystore | openssl sha1 -binary | openssl base64

generated hash key place into intput field click on next and finish

code but also all code is available at <https://github.com/facebook/react-native-fbsdk>

import React, { Component } from 'react';

import { View ,StatusBar,TouchableWithoutFeedback} from 'react-native';

import { LoginButton, AccessToken,LoginManager } from 'react-native-fbsdk';

import { Text } from 'native-base';

export default class Login extends Component {

logout=()=>{

LoginManager.logOut()

alert('logout')

}

login=()=>{

LoginManager.logInWithPermissions(["public\_profile"]).then(

function(result) {

if (result.isCancelled) {

console.log("Login cancelled");

} else {

console.log(

"Login success with permissions: " +

result.grantedPermissions.toString()

);

console.log(result)

AccessToken.getCurrentAccessToken().then((accessToken) => console.log(accessToken))

}

},

function(error) {

console.log("Login fail with error: " + error);

}

);

}

render() {

return (

<View>

<View style={{width:'100%'}}>

{/\* <LoginButton

onLoginFinished={

(error, result) => {

if (error) {

console.log("login has error: " + result.error);

} else if (result.isCancelled) {

console.log("login is cancelled.");

} else {

AccessToken.getCurrentAccessToken().then(

(data) => {

console.log(data.accessToken.toString())

}

)

}

}

}

onLogoutFinished={() => console.log("logout.")}/> \*/}

{/\* custom login \*/}

<TouchableWithoutFeedback onPress={()=>this.login()}>

<Text style={{marginTop:50}}>

Login

</Text>

</TouchableWithoutFeedback>

<TouchableWithoutFeedback onPress={()=>this.logout()}>

<Text style={{marginTop:100}}>

Logout

</Text>

</TouchableWithoutFeedback>

</View>

</View>

);

}

};

**ios configuration :**After configuration with android and successfully run click on start over and select ios  
  
1-**Configure your info.plist and project settings**

only copy and past code code into .plist  
2-**Supply us with your Bundle Identifier**

only copy and past bundleridentifier from xcode