Instructions

1 - Create a new project

- Create blank ionic project using the latest ionic-cli

```
ionic start StarterFirebasePackV3 sidemenu --type=angular
```

- Copy the folders `src` , `package.json` and replace them in your new project root.
- Run inside the project folder, install all dependencies added to the new package.json file running:

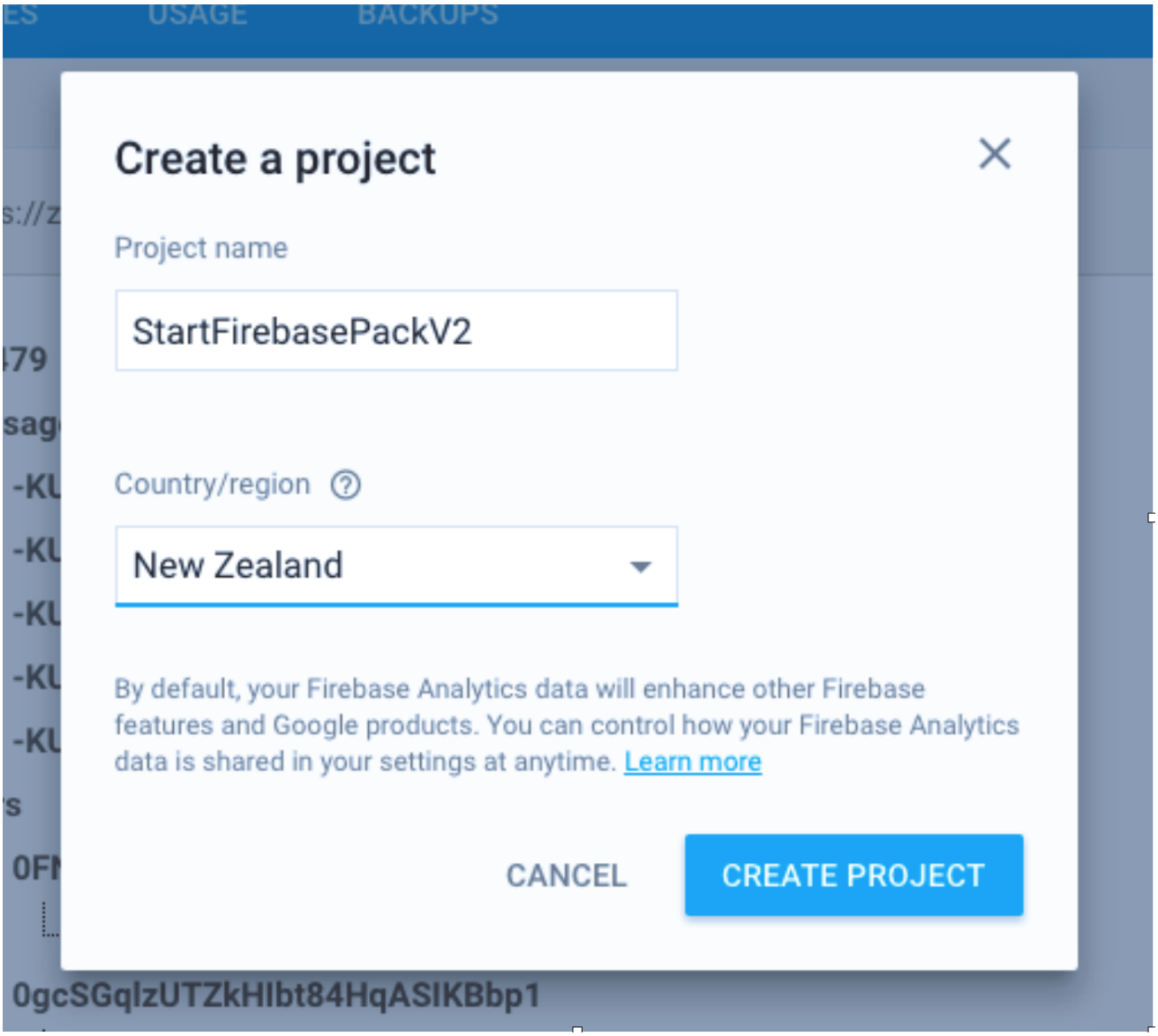
```
npm install
```

- Add all platforms you need:

```
ionic cordova platform add ios  ionic cordova platform add android
```

2 - Configuring Firebase project

- Create a Firebase project



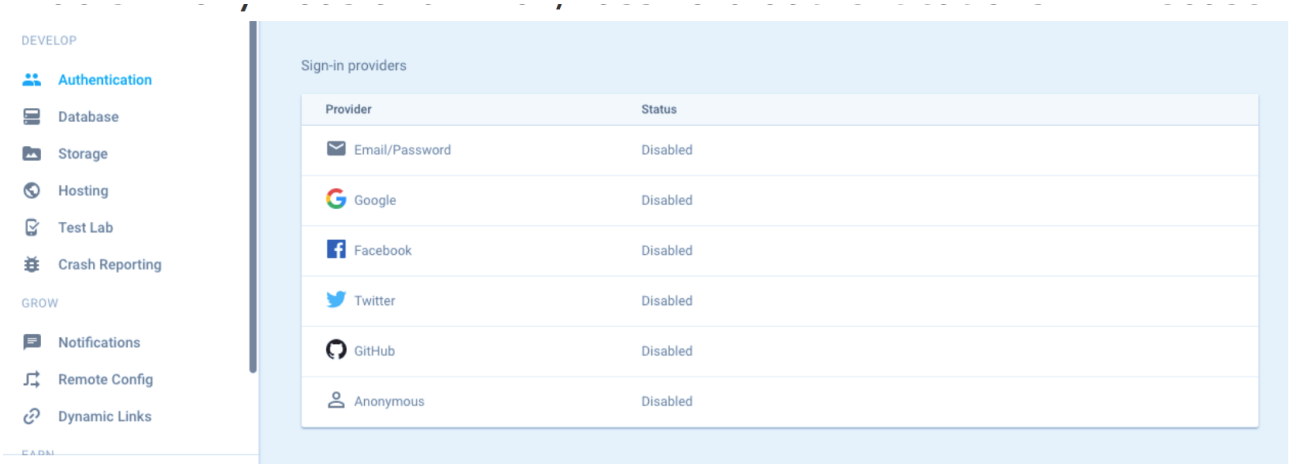
- Add project to Web application.

- Add info inside the `config.ts` file:

```
export const firebaseConfig = {
    apiKey: "",
    authDomain: "",
    databaseURL: "",
    projectId: "",
    storageBucket: "",
    messagingSenderId: ""
};
```

3 - Anonymous and Email/password auth.

- Enable Anonymous and Email/Password authentications in Firebase.

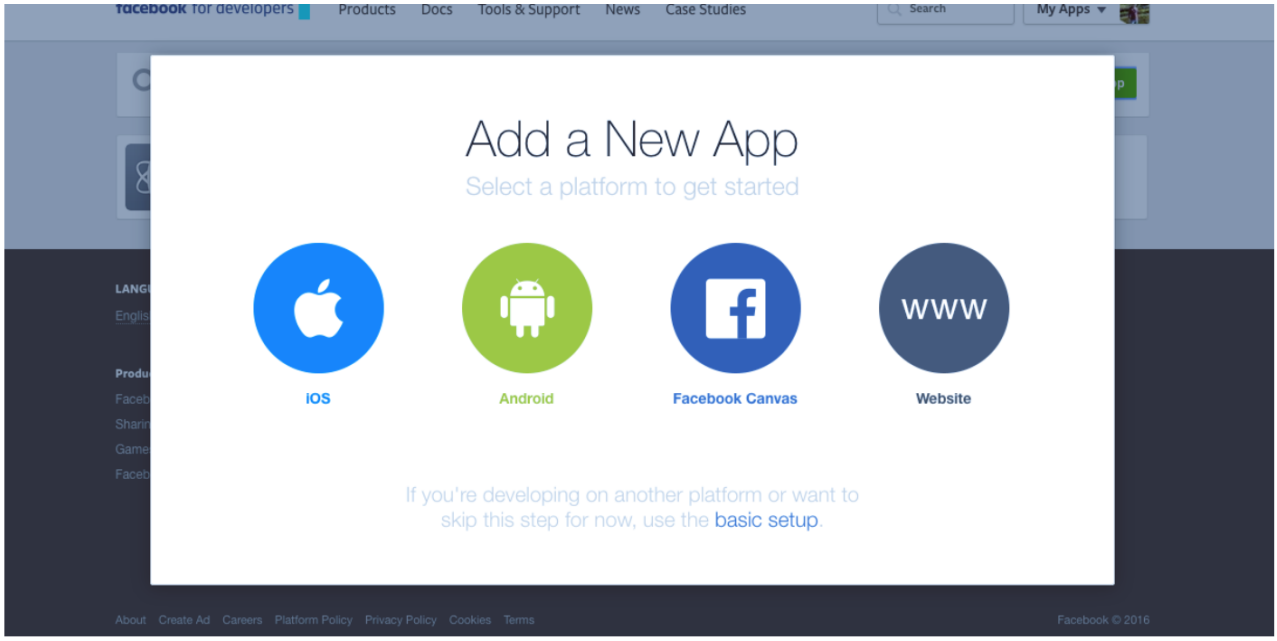


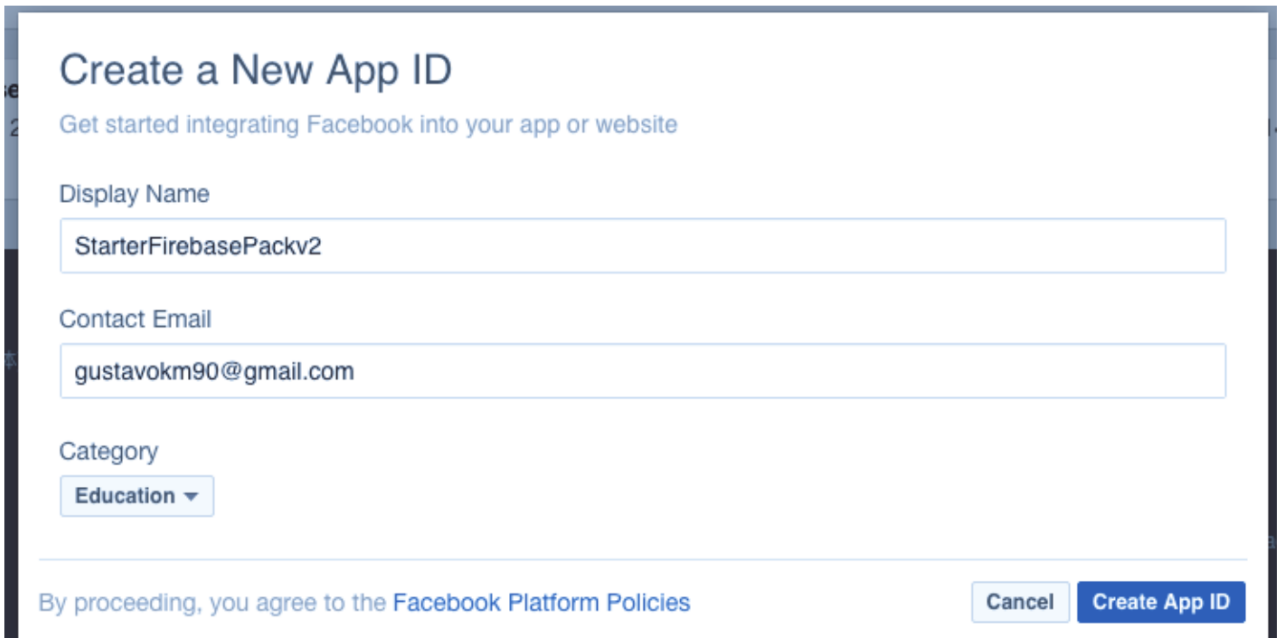
- If you run `ionic serve` in your project root, you should be able to Register and authenticate via Email/password and Anonymous users.

4 - Facebook oAuth v4 SDK

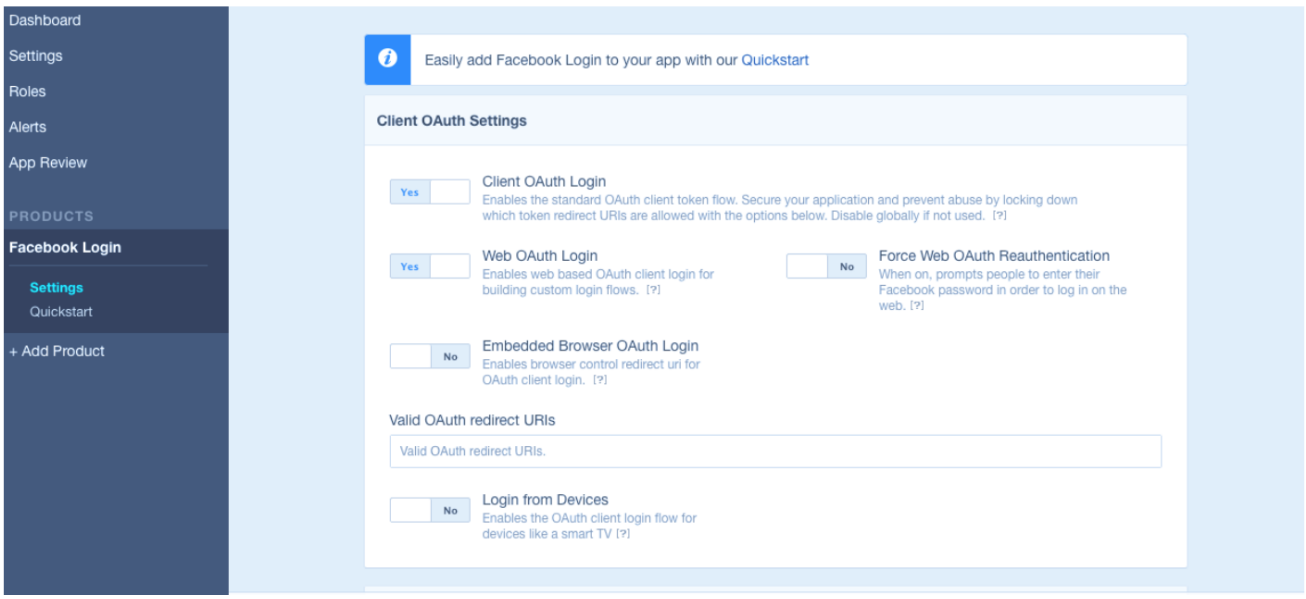
- Enable Facebook login in Firebase – **Copy the redirect URL since we will use it later, leave this page opened for now, we will get the App ID and App secret later.**
- Create Facebook developer app, use the Basic Setup.

https://developers.facebook.com/apps





- Now create a Facebook Login product and paste the OAuth redirect URL provided by Firebase when you enabled the Facebook Authentication.



- Save, grab the App ID and App Secret in your Facebook Dashboard, and add them in the Firebase Facebook Authentication.

- Run the following command:

```
ionic cordova plugin add cordova-plugin-facebook4 --save --variable APP_ID="your_facebook_app_id" --variable APP_NAME="StarterFirebasePackV2"
```

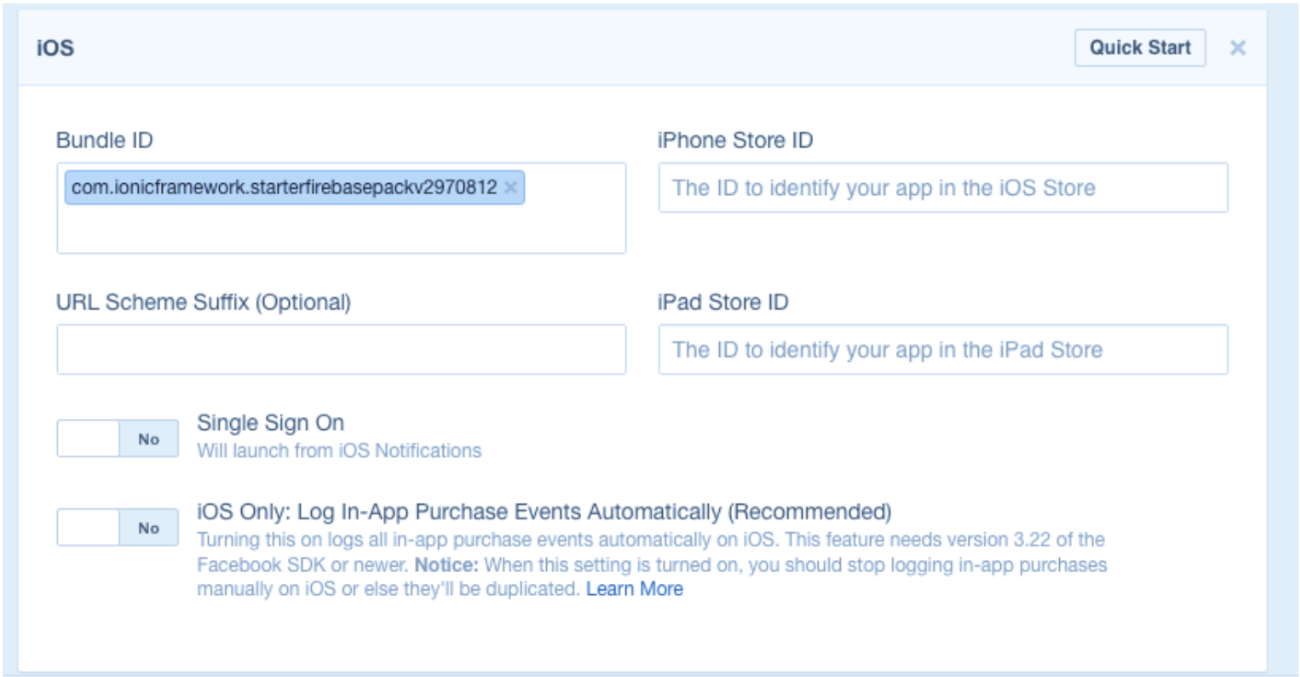
- At this point the Facebook login should work in the Browser Mode only
- However, in order to make the Facebook login work for iOS and Android devices, we need to configure a few more things.

Let's enable for iOS first.

- In the Facebook developer dashboard, go to Settings.
- Add a new platform.
- Choose iOS.
- Now add your Bundle ID – found in the `config.xml` file.

```
<widget id=…….>
```

- Save, and you are done.

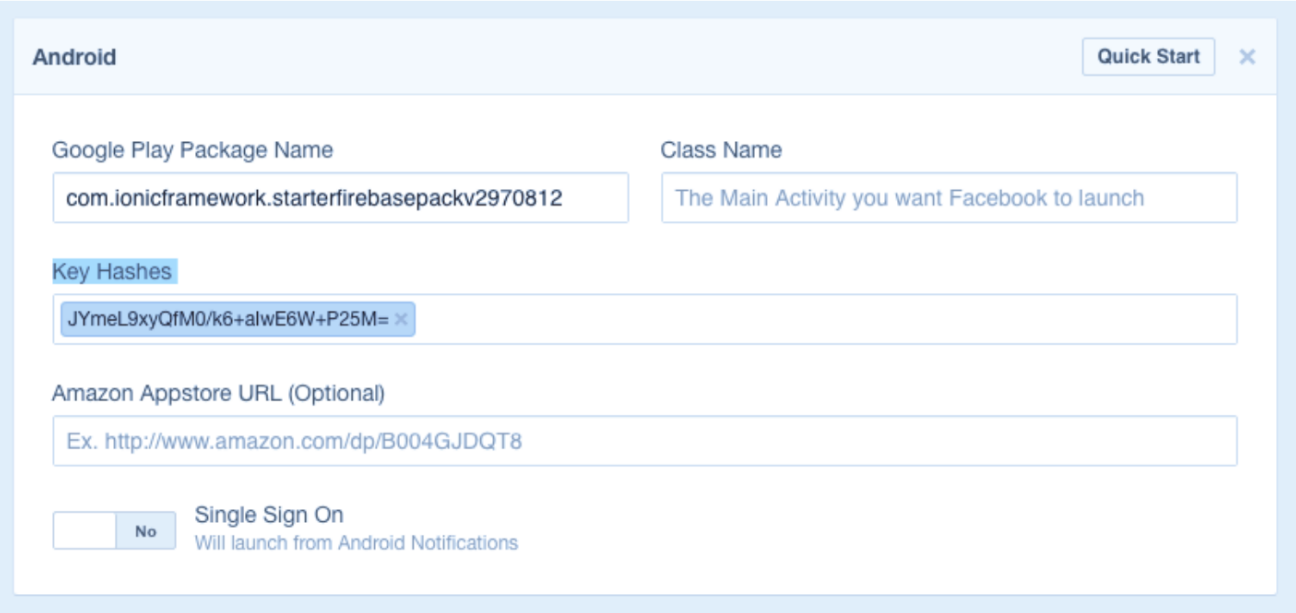


Next up Android.

- In the Facebook developer dashboard, go to Settings.
- Add a new platform.
- Choose Android.
- Add the same value for the Bundle ID in iOS in the field Google Play Package Name.
- Now things gets a little bit nasty, we need to generate a Hash key for the Android apk. **We are going to generate a Hash for the debug apk, so we can deploy our app to an Android device and test.**
- First thing, run the command in your project root folder.

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

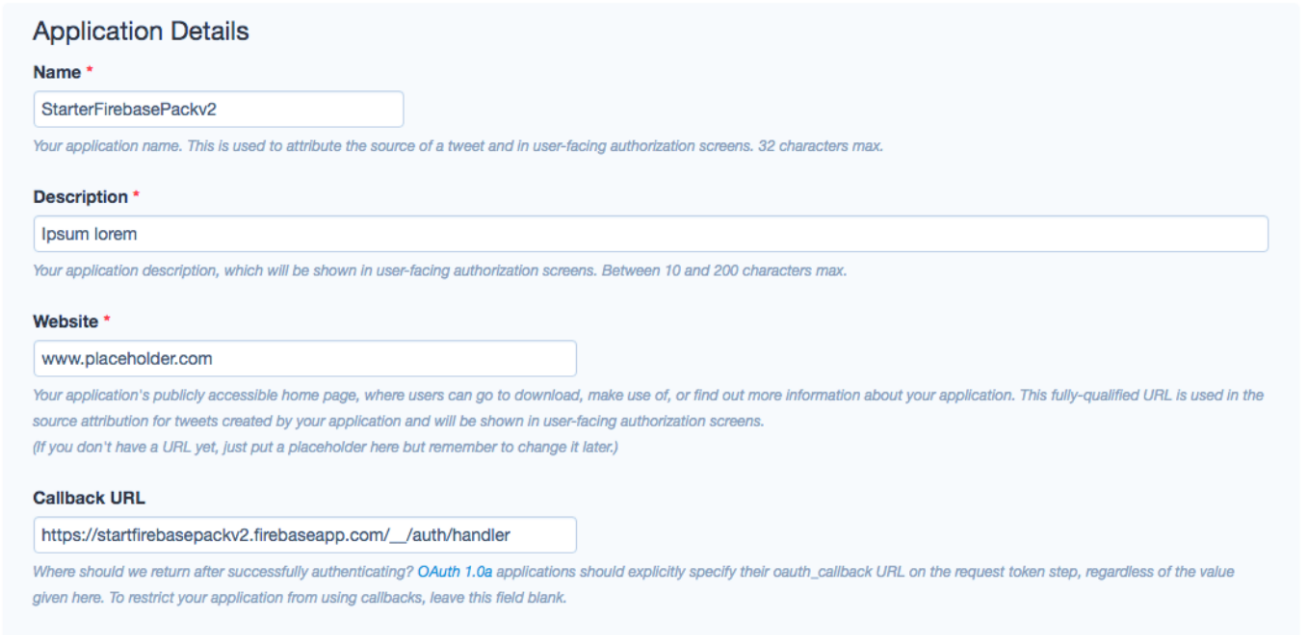
- The password for the `debug.keystore` should be always `android` .
- Now copy the hash generated by this command and paste into the Facebook Android product field Key hashes.



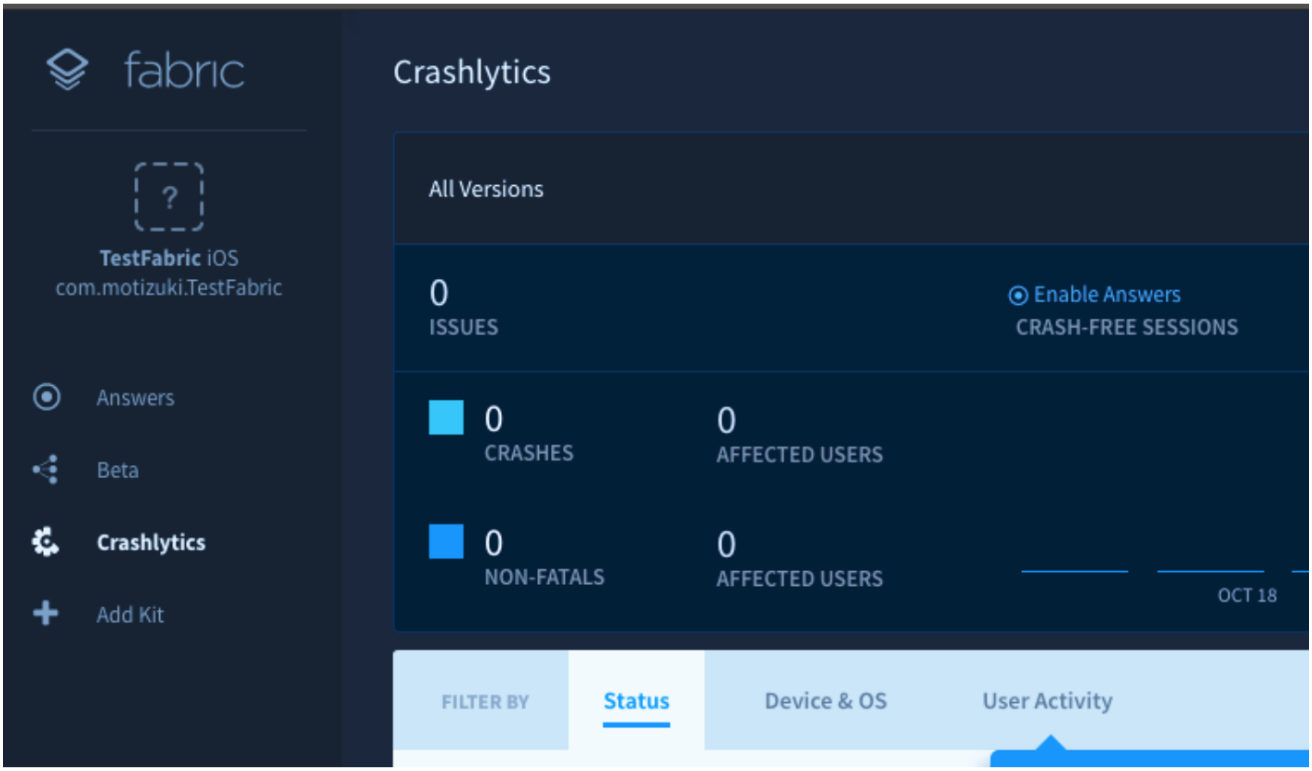
And we are done! You should now be able to Login via Facebook now using devices.

5 - Twitter oAuth

- Enable Twitter authentication if Firebase.
- Copy the callback URL.
- Create a Twitter application at: https://apps.twitter.com/app
- Add the callback URL to the application.



- Go to keys and access token tab and grab your API key and API secret.
- Go back to Firebase website and add them to the Twitter Authentication and save.
- Now we need to create a Fabric application to use the Fabric API in the Twitter Connect Plugin.
- Go to : https://fabric.io
- Create any project, It could be an iOS application for instance.



- The only thing we really need is the API key. Getting the API key is fairly tricky, but this process seems to work:
  - Login to Fabric account and open https://fabric.io/kits/android/crashlytics/install
  - Find the meta-data code block in AndroidManifest.xml
  - Find your API Key pre filled in the code.
- Now run the following command with your Fabric API KEY in your project root:

```
ionic cordova plugin add twitter-connect-plugin --variable FABRIC_KEY=<Fabric API Key>
```

- The last thing you need to do now is to open config.xml (in your project's root) and add these two lines before the closing `</widget>` tag:

```
<preference name="TwitterConsumerKey" value="<Twitter Consumer Key>" />
<preference name="TwitterConsumerSecret" value="<Twitter Consumer Secret>" />
```

If you get stuck in one of those steps, you can checkout the plugin github page for more information: https://github.com/ManifestWebDesign/twitter-connect-plugin.

6 - GooglePlus oAuth

Enable Google authentication in Firebase.

iOS

- To get your iOS `REVERSED_CLIENT_ID` , generate a configuration file here. This `GoogleService-Info.plist` file contains the `REVERSED_CLIENT_ID` you'll need during installation.
- Go to: https://console.firebase.google.com/u/0/
- Select your Firebase Project
- Click on `Add app` in the main dashboard.
- Select iOS
- Create a new application by passing a name and adding the `BundleID` .
- Enable Google Sign in.
- Download the plist file.
- You will find your reverse client ID inside this file.
- Run the following command using your reverse client id:

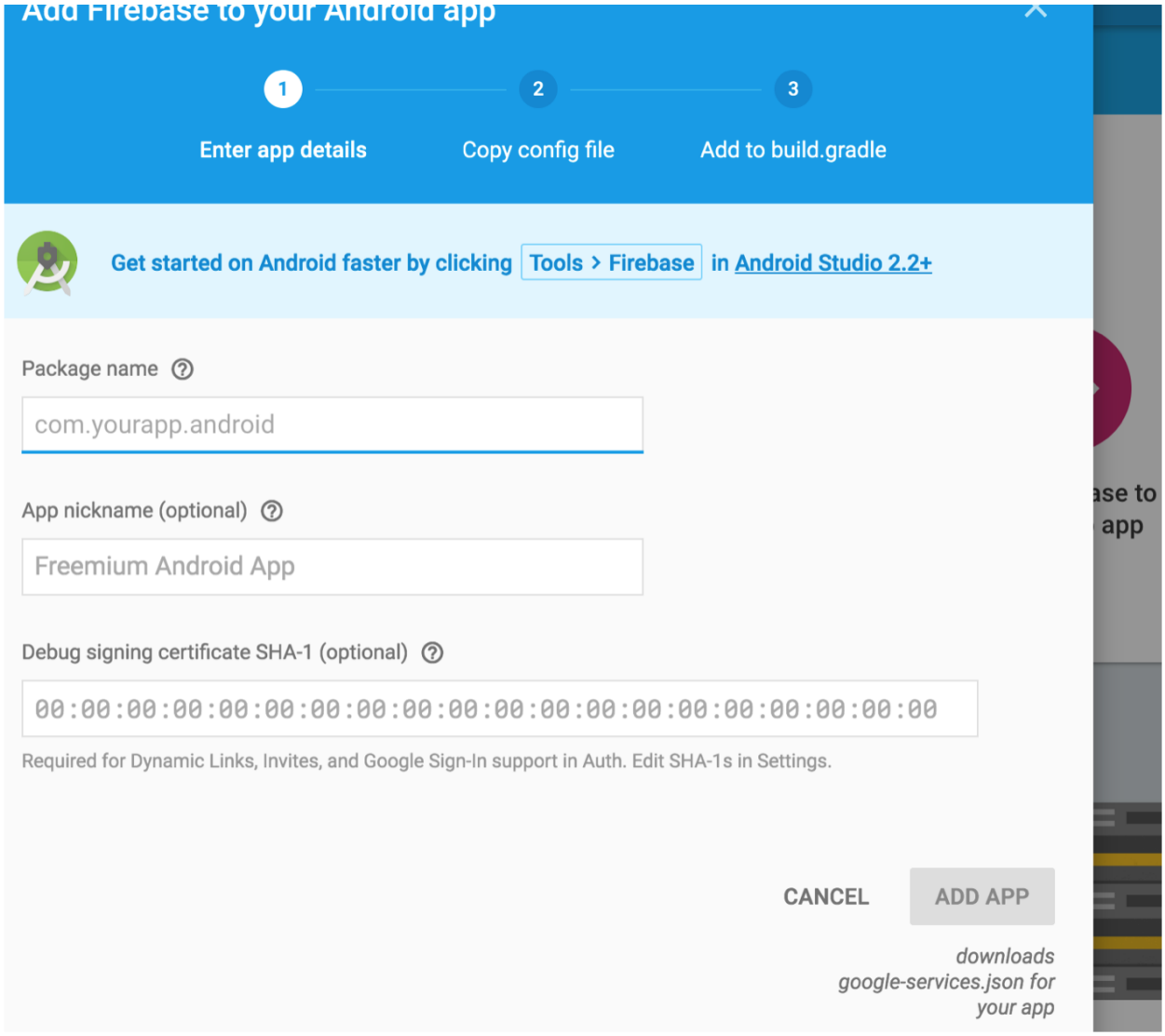
```
ionic cordova plugin add cordova-plugin-googleplus --save --variable REVERSED_CLIENT_ID=myreversedclientid
```

Android

- Go to: https://console.firebase.google.com/u/0/
- Select your Firebase Project
- Click on `Add app` in the main dashboard.
- Select Android
- Create a new application by passing a name and adding the `BundleID`.
- Enable Google Sign in.
- We need to get the SHA1 Hash now.
- Run the command:

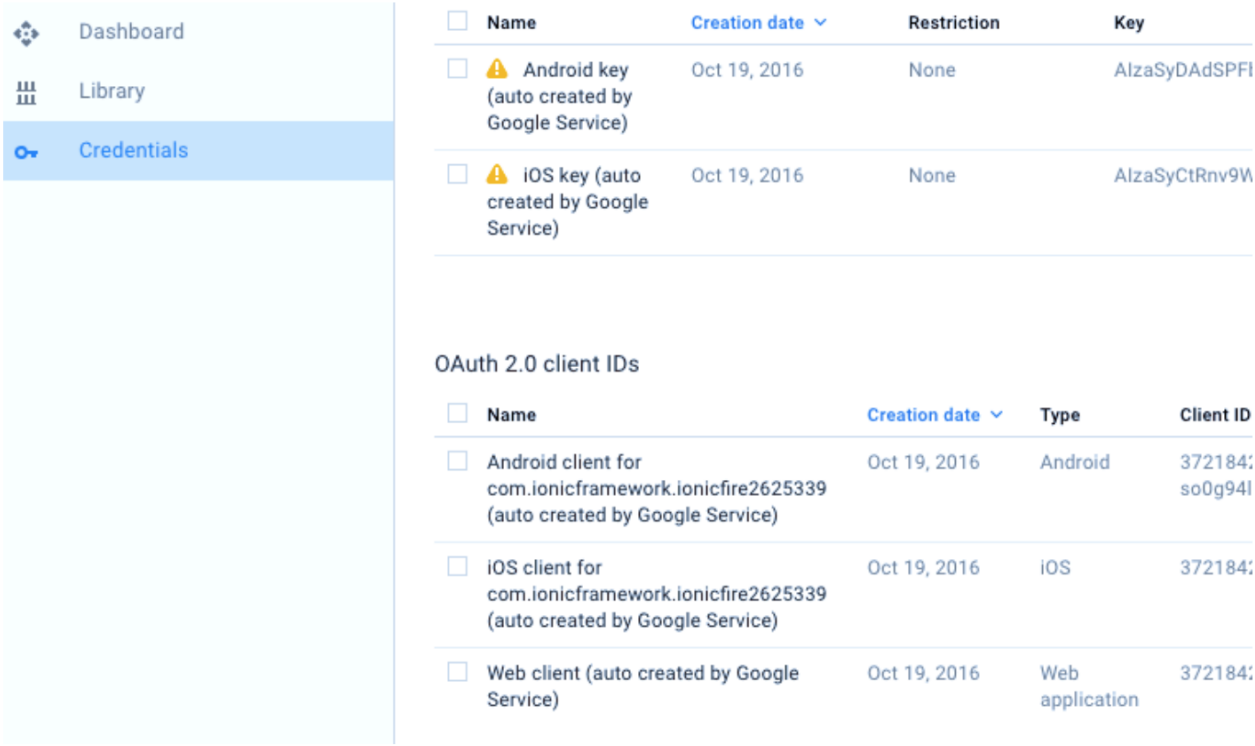
```
keytool -exportcert -list -v -alias androiddebugkey -keystore ~/.android/debug.keystore
```

- Get the `SHA-1` and add to the enable the Google Sign in.
- Add in Firebase an Android app



- Add the `SHA-1` into it
- Lastly, get your web client ID at: https://console.developers.google.com/apis/credentials
- Add it to the `/src/app/config.ts`

- Make sure all client ids were correctly added, and they should all be present in the console dashboard.



If you had any trouble running those steps, checkout the github website for the plugin:
https://github.com/EddyVerbruggen/cordova-plugin-googleplus

Happy coding