Aula 43

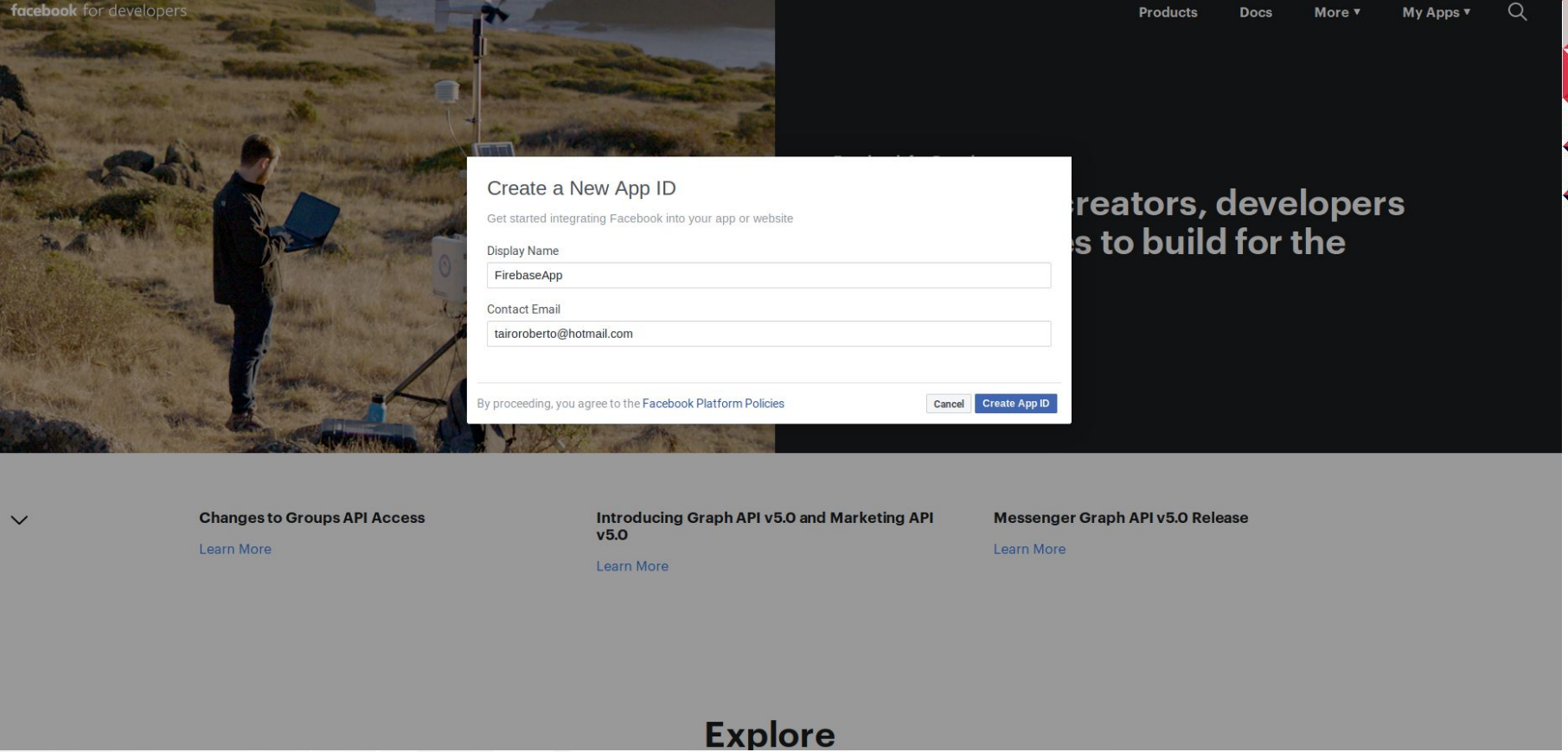
Facebook

DigitalHouse >
Coding School

SDK do Facebook

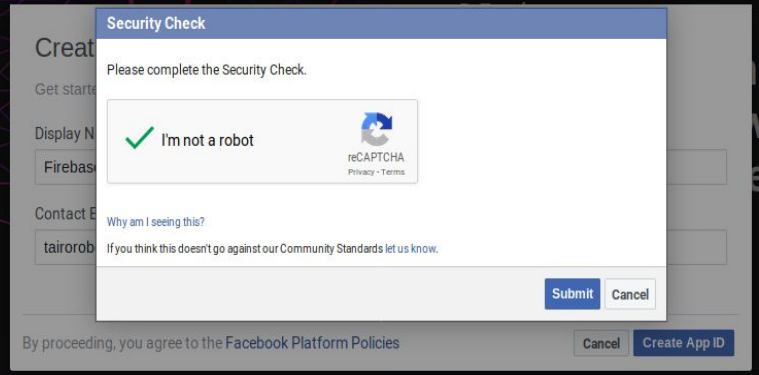
O SDK do Facebook para Android é a forma mais simples de integrar seu aplicativo móvel ao Facebook

https://developers.facebook.com/



Acesse o Painel do facebook e crie um novo aplicativo

DigitalHouse > Coding School



Changes to Groups API Access

Learn More

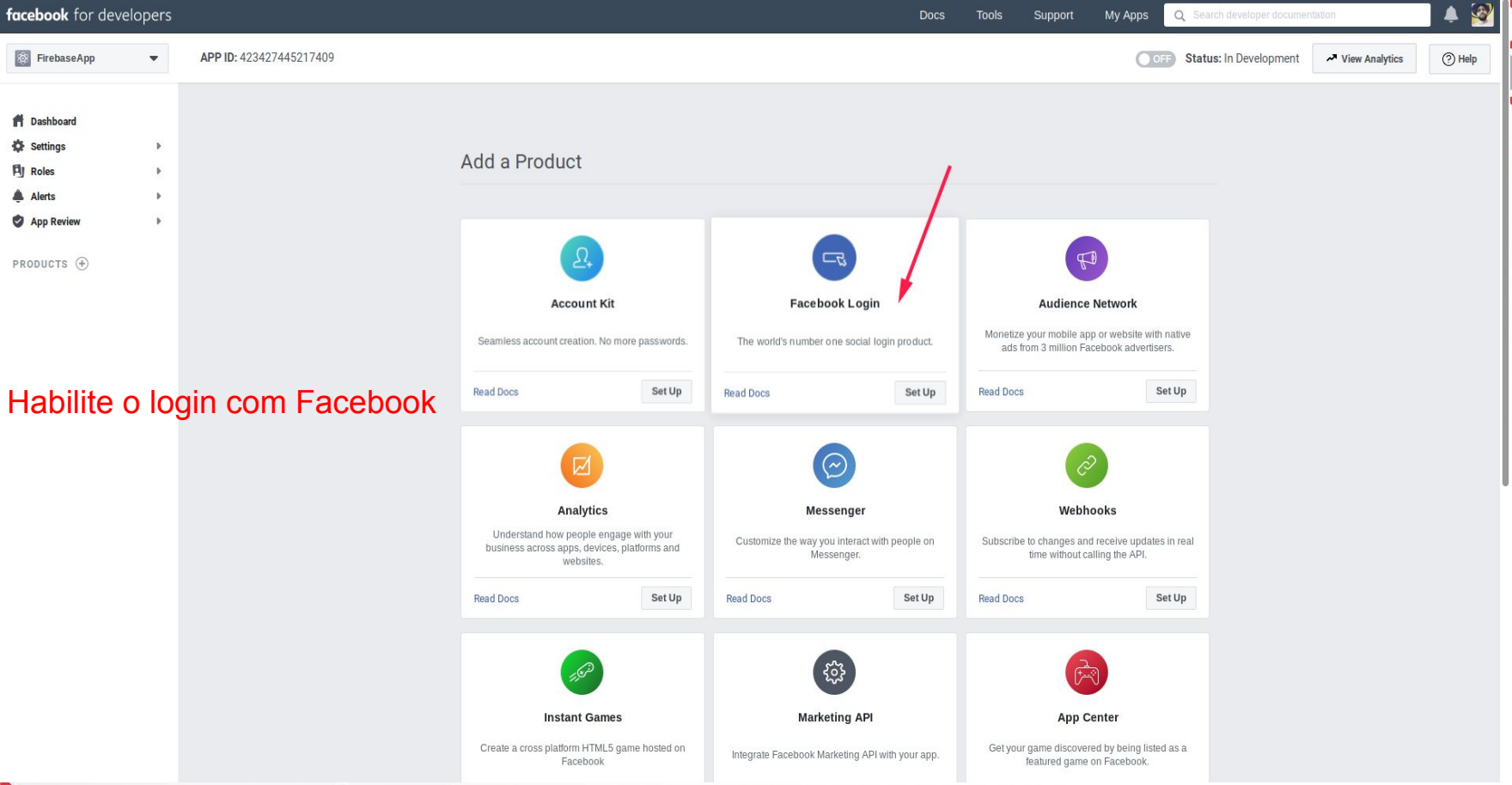
Introducing Graph API v5.0 and Marketing API v5.0

Learn More

Messenger Graph API v5.0 Release

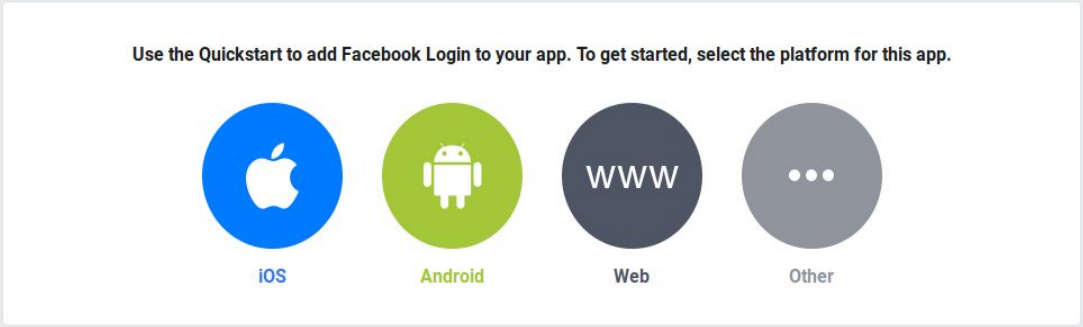
Learn More

DigitalHouse > Coding School



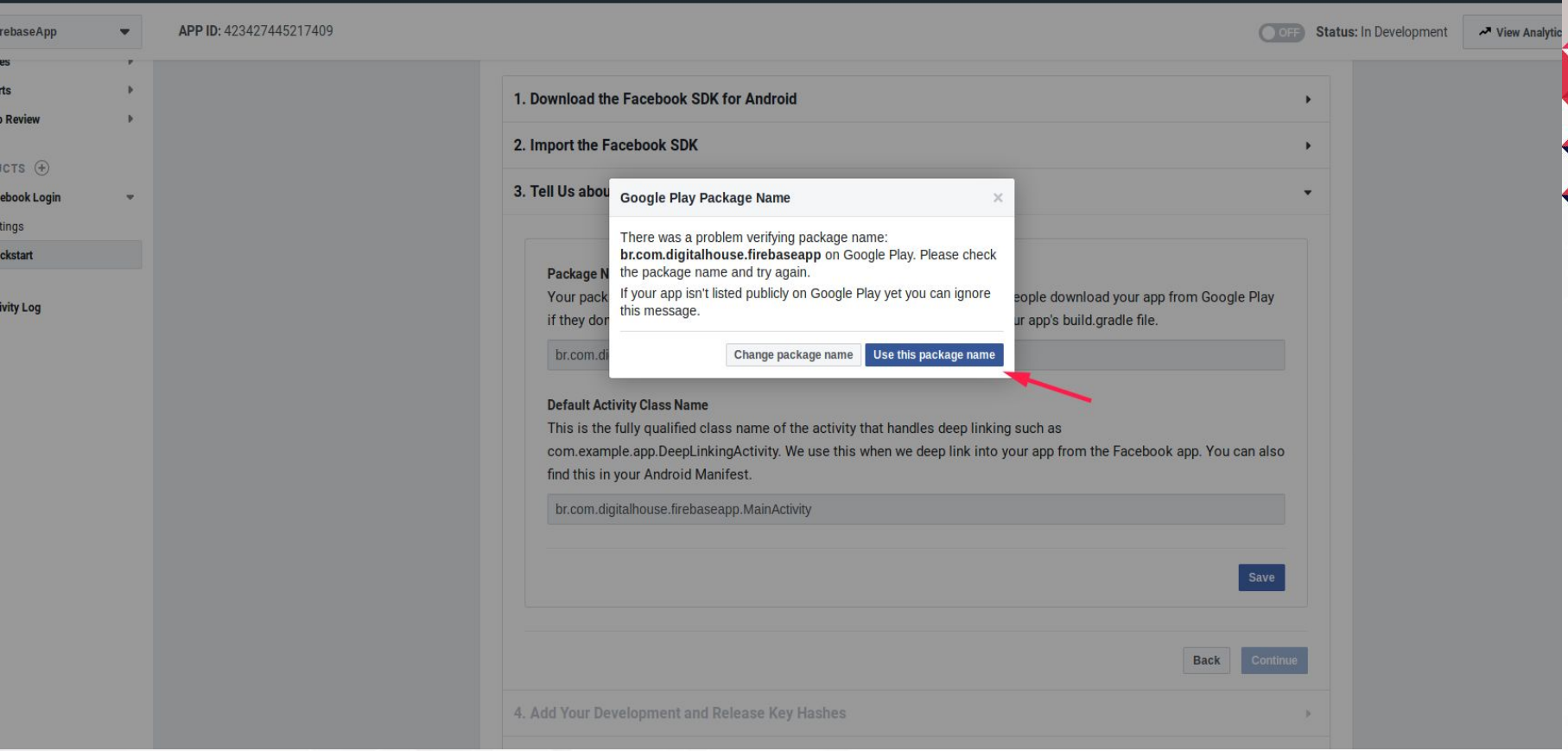
Habilite o login com Facebook

DigitalHouse > Coding School



Selecione o sistema que irá usar para a conexão com o login

DigitalHouse >
Coding School



Informe o application ID e nome completo da activity que será responsável pelo login. E confirme no modal que aparecerá.

DigitalHouse > Coding School

4. Add Your Development and Release Key Hashes

To ensure the authenticity of the interactions between your app and Facebook, you need to supply us with the Android key hash for your development environment. If your app has already been published, you should add your release key hash too.

Generating a Development Key Hash
You'll have a unique development key hash for each Android development environment.

Mac OS
To generate a development key hash, open a terminal window and run the following command:

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

Copy Code

Copiar e colar no terminal do git bash

Windows
You will need the following:

- Key and Certificate Management Tool (`keytool`) from the Java Development Kit
- `openssl-for-windows` openssl library for Windows from the Google Code Archive

To generate a development key hash, run the following command in a command prompt in the Java SDK folder:

```
keytool -exportcert -alias androiddebugkey -keystore "C:\Users\USERNAME\.android\debug.keystore" | "PATH_T
```

Copy Code

This command will generate a 28-character key hash unique to your development environment. Copy and paste it into the field below. You will need to provide a development key hash for the development environment of each person who works on your app.

DigitalHouse > Coding School

This command will generate a 28-character key hash unique to your development environment. Copy and paste it into the field below. You will need to provide a development key hash for the development environment of each person who works on your app.

Generating a Release Key Hash

Android apps must be digitally signed with a release key before you can upload them to the store. To generate a hash of your release key, run the following command on Mac or Windows substituting your release key alias and the path to your keystore:

Copy Code

```
keytool -exportcert -alias YOUR_RELEASE_KEY_ALIAS -keystore YOUR_RELEASE_KEY_PATH | openssl sha1 -binary |
```

This will generate a 28-character string that you should copy and paste into the field below. Also, see the Android documentation for signing your apps.

Key Hashes

HRWUGSvAEyfbhBAtipqFbuH1qTA= ×

Colar hash gerado pelo comando acima e salvar

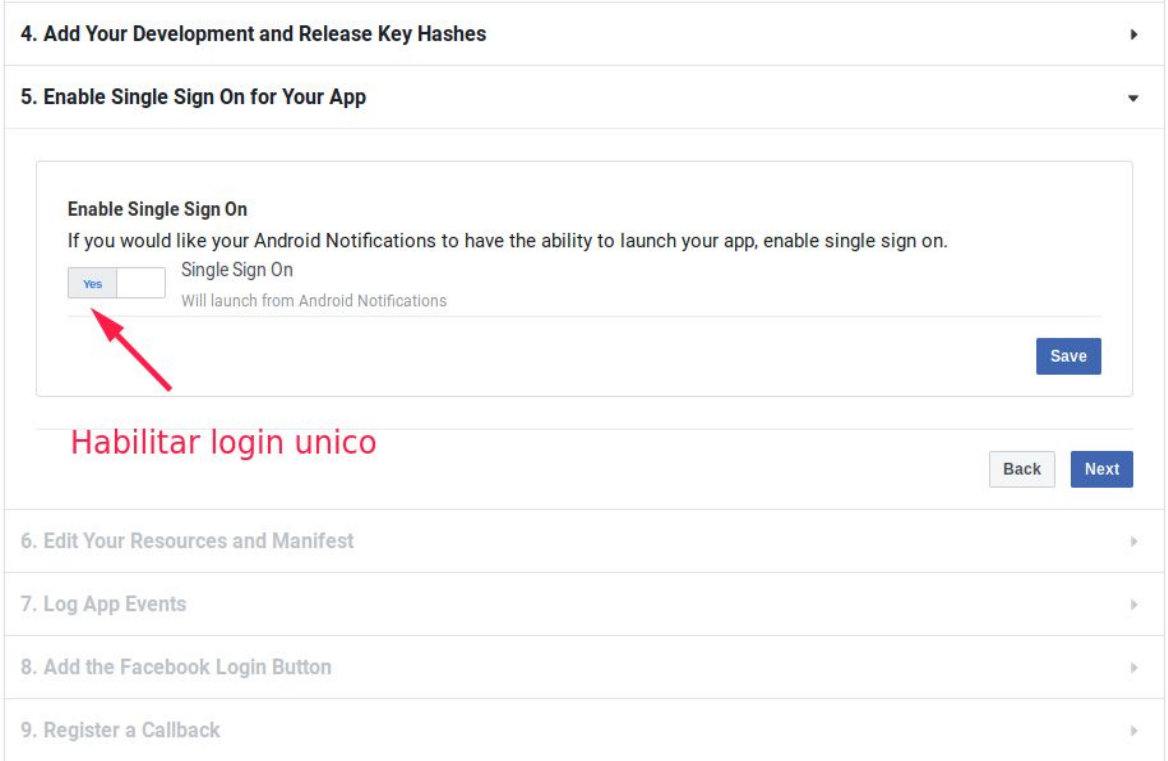
Save

Back Continue

5. Enable Single Sign On for Your App

6. Edit Your Resources and Manifest

7. Log App Events


4. Add Your Development and Release Key Hashes
5. Enable Single Sign On for Your App
Enable Single Sign On
If you would like your Android Notifications to have the ability to launch your app, enable single sign on.
Yes
Single Sign On
Will launch from Android Notifications
Save
Habilitar login unico
Back
Next
6. Edit Your Resources and Manifest
7. Log App Events
8. Add the Facebook Login Button
9. Register a Callback

6. Edit Your Resources and Manifest

Create strings for your Facebook app ID and for those needed to enable Chrome Custom Tabs. Also, add `FacebookActivity` to your Android manifest.

1. Open your `/app/res/values/strings.xml` file.
2. Add the following:

```
<string name="facebook_app_id">423427445217409</string>
<string name="fb_login_protocol_scheme">fb423427445217409</string>
```

Copy Code

Colar no arquivo strings.xml

3. Open the `/app/manifest/AndroidManifest.xml` file.
4. Add the following `uses-permission` element after the `application` element:

```
<uses-permission android:name="android.permission.INTERNET"/>
```

Copy Code

5. Add the following `meta-data` element, an activity for Facebook, and an activity and intent filter for Chrome Custom Tabs inside your `application` element:

```
<meta-data android:name="com.facebook.sdk.ApplicationId"
        android:value="@string/facebook_app_id"/>

    <activity android:name="com.facebook.FacebookActivity"
        android:configChanges=
                "keyboard|keyboardHidden|screenLayout|screenSize|orientation"
        android:label="@string/app_name" />
```

Copy Code

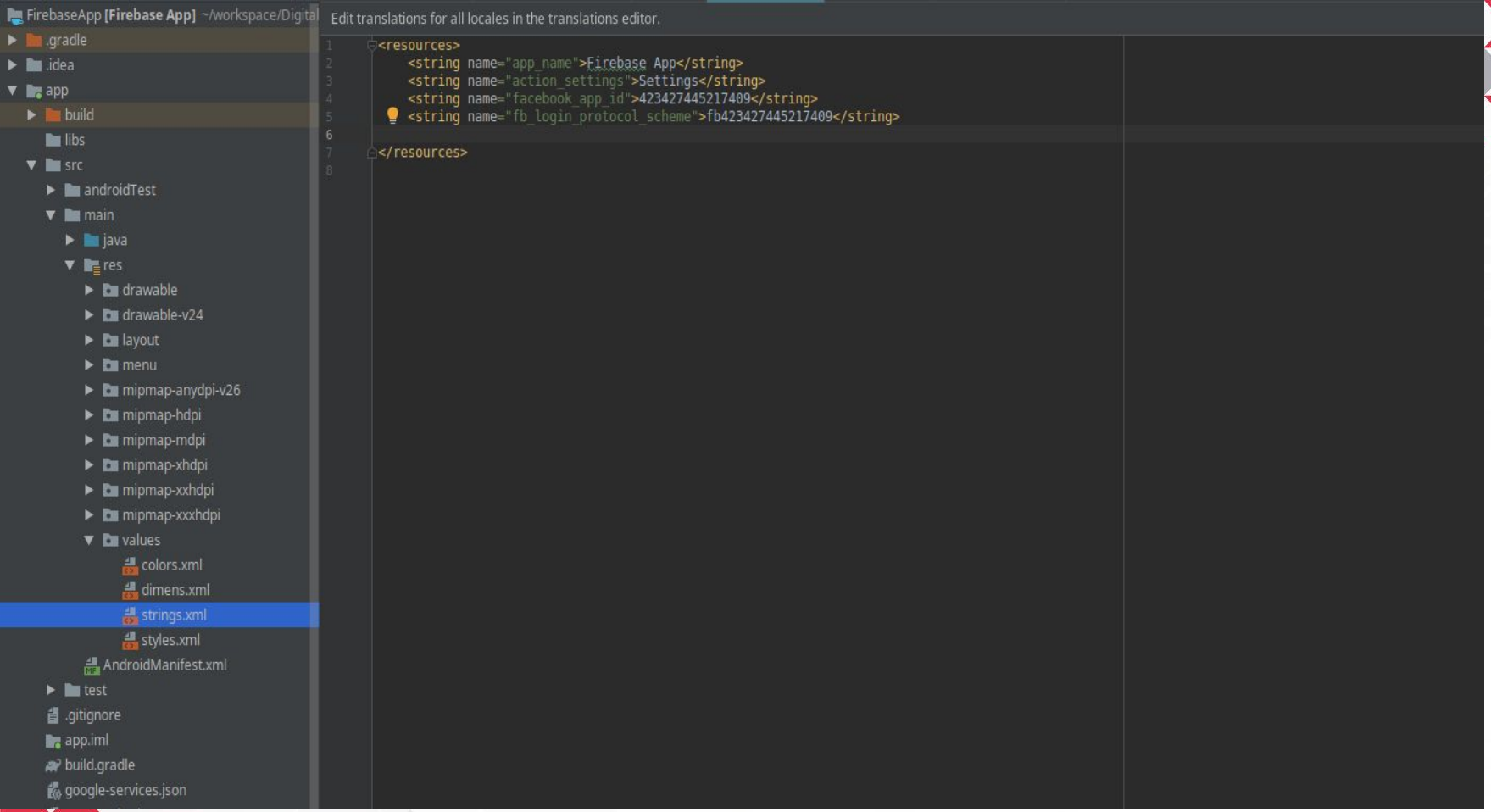
Colar no arquivo AndroidManifest.xml

Back Next

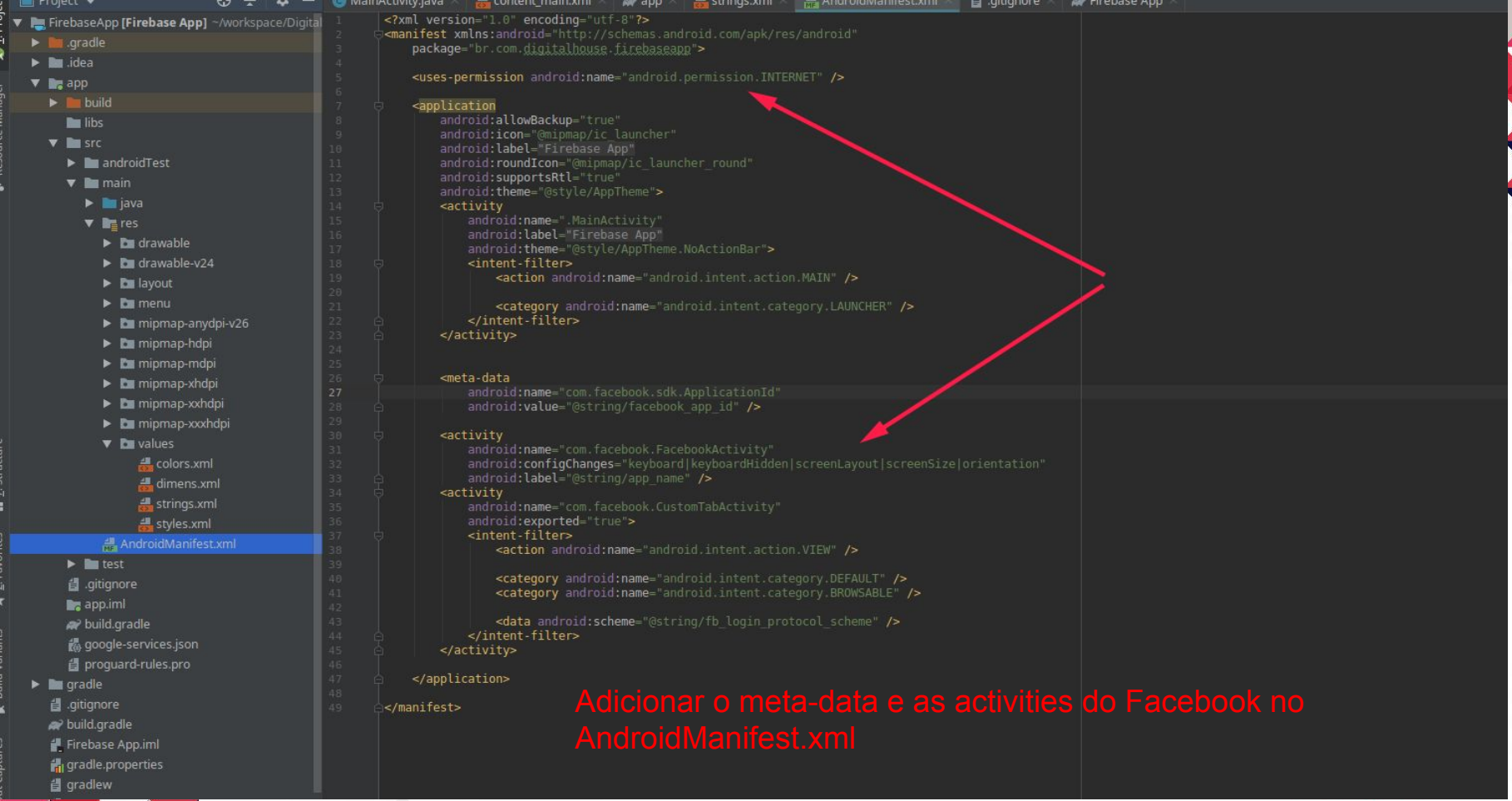
7. Log App Events

8. Add the Facebook Login Button

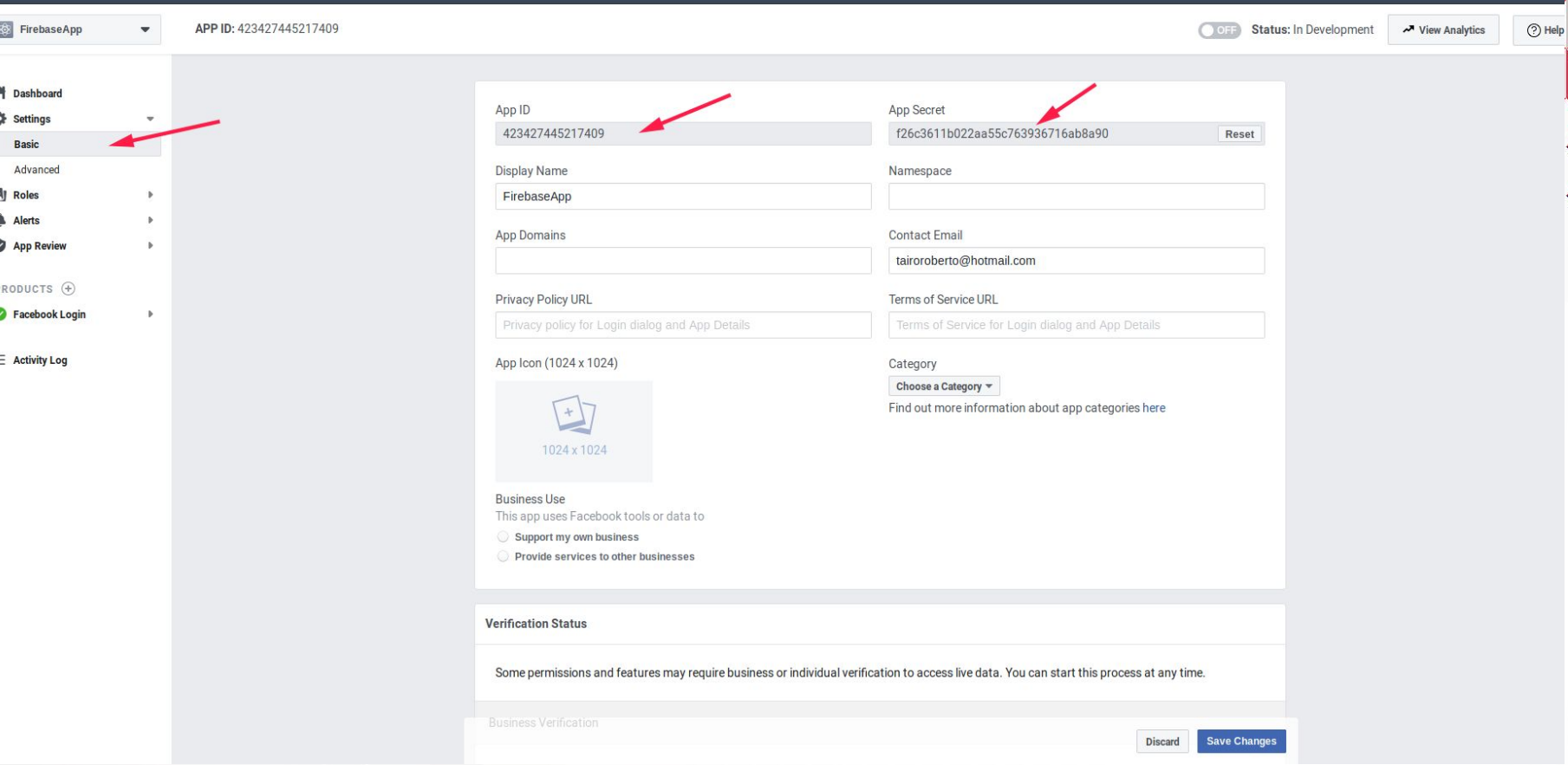
DigitalHouse > Coding School



DigitalHouse > Coding School

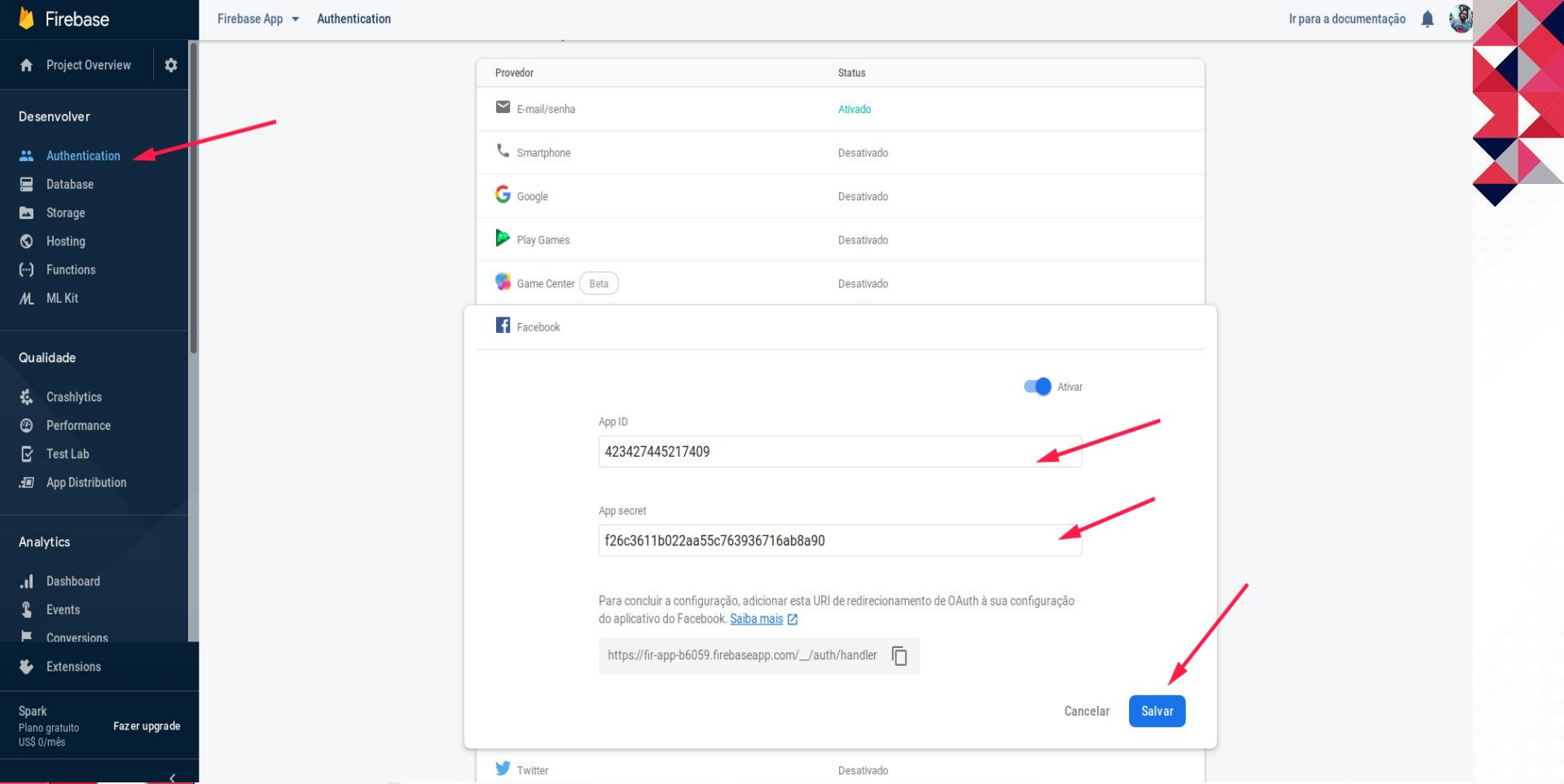


DigitalHouse > Coding School



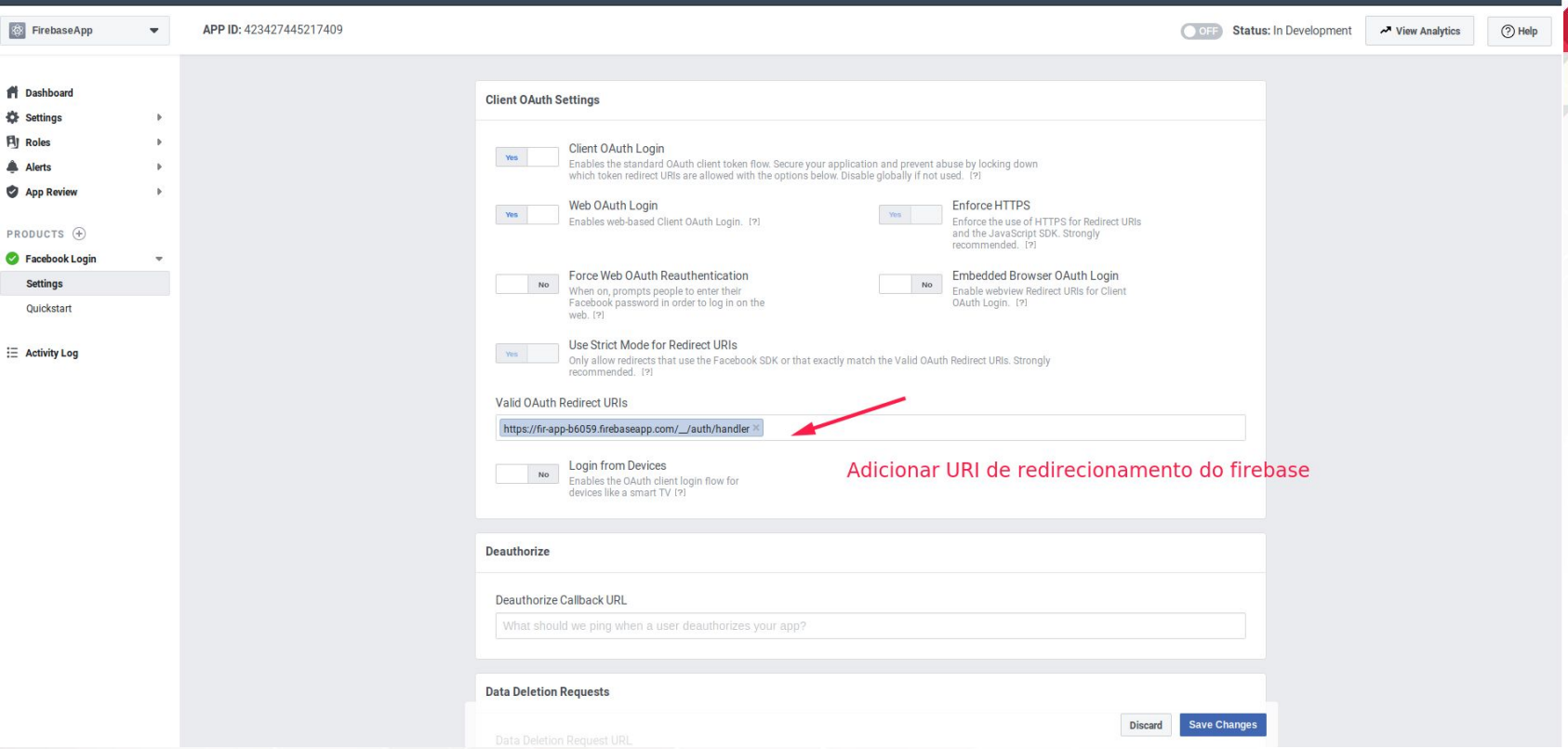
Ir no Painel do facebook copiar App ID e App Secret habilitar o login com facebook no Painel de Authentication no Firebase.

DigitalHouse > Coding School



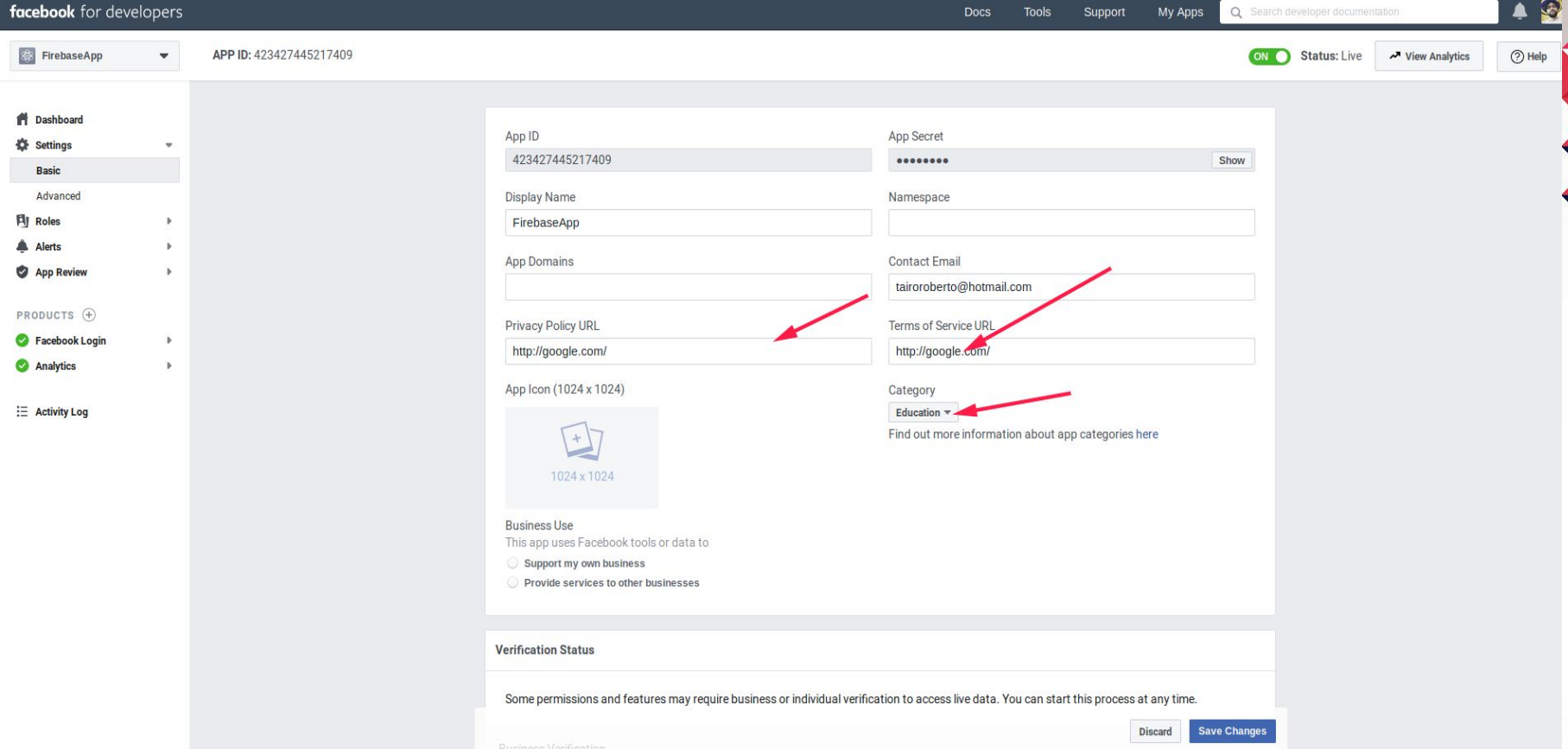
Copie a url de redirecionamento do OAuth

DigitalHouse > Coding School



Voltamos ao facebook em facebook login -> settings e adicionamos a Url de redirecionamento do OAuth

DigitalHouse > Coding School



Em Settings adicione uma url para termos de privacidade, uma url para termos de serviços e uma categoria de aplicativo. Assim podemos habilitar nossa aplicação para produção em vez de desenvolvimento.

DigitalHouse > Coding School

Docs Tools Support My Apps Search developer documentation

427445217409

OFF **Status:** In Development View Analytics

Mudar para produção

App ID
423427445217409

App Secret
•••••••• Show

Display Name
FirebaseApp

Namespace

App Domains

Contact Email
tairoroberto@hotmail.com

Privacy Policy URL
http://google.com/

Terms of Service URL
http://google.com/

App Icon (1024 x 1024)

1024 x 1024

Category
Education

Find out more information about app categories here

Business Use
This app uses Facebook tools or data to
- Support my own business
- Provide services to other businesses

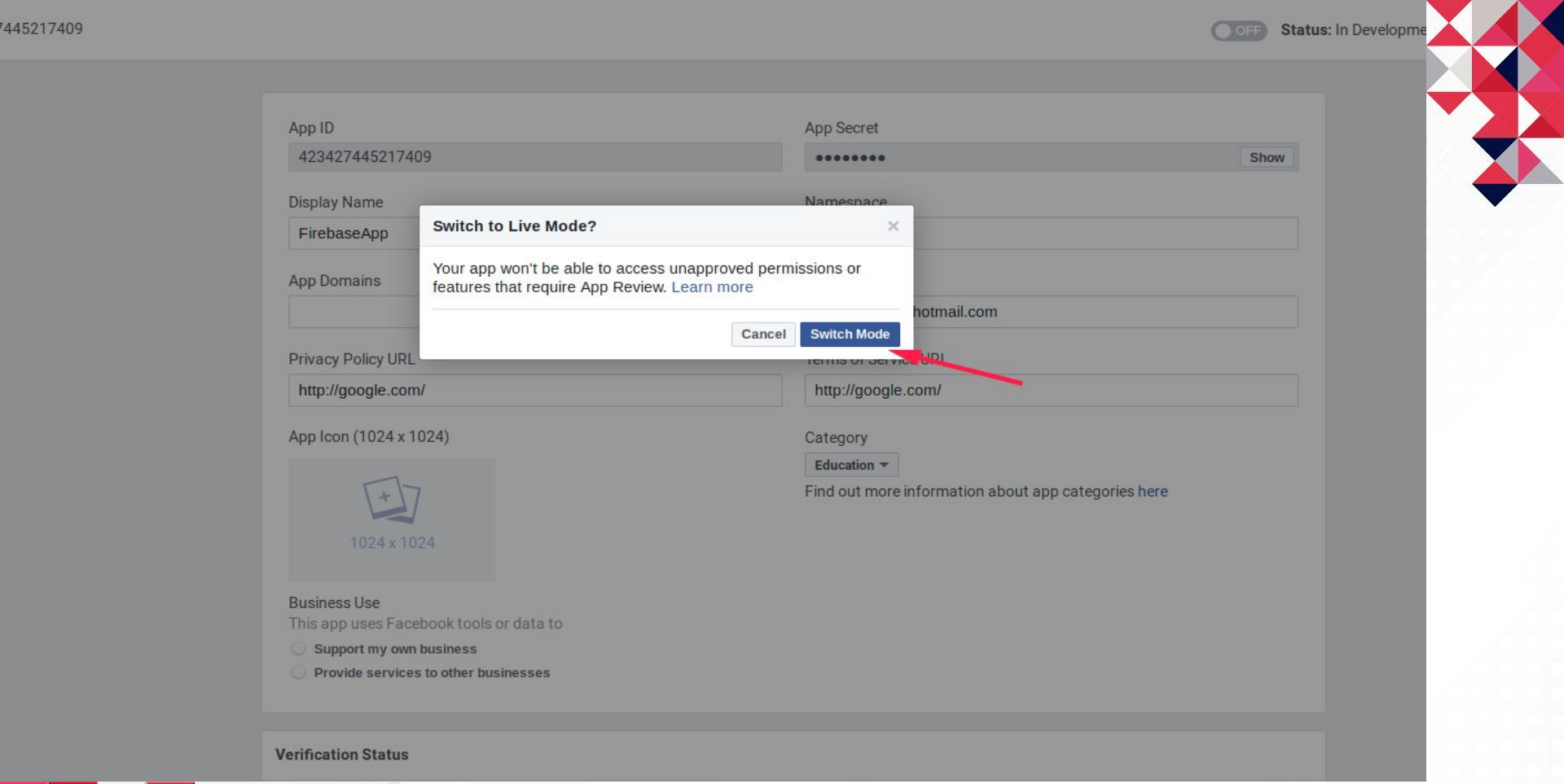
Verification Status

Some permissions and features may require business or individual verification to access live data. You can start this process at any time.

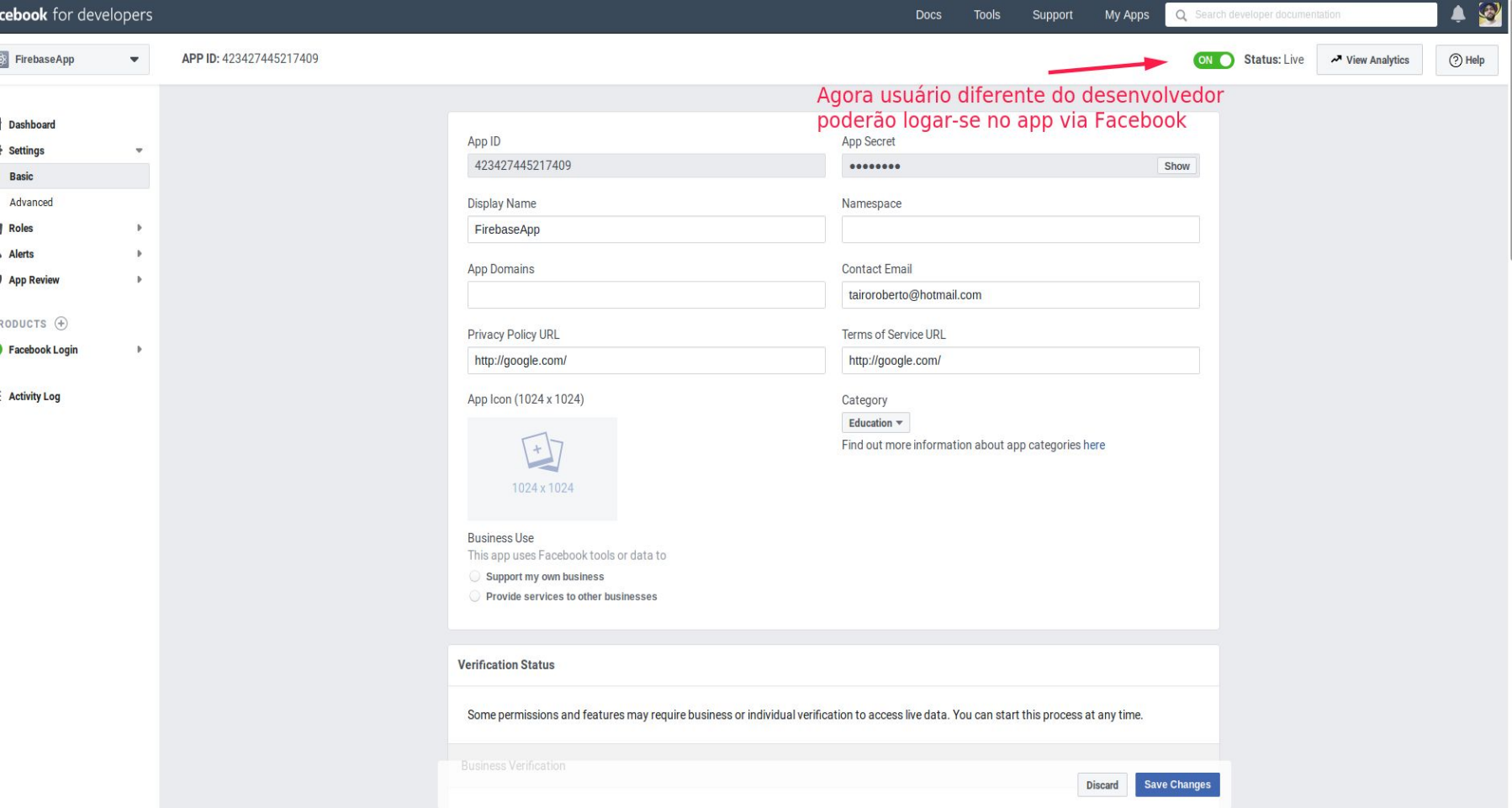
Business Verification

Discard Save Changes

DigitalHouse > Coding School



DigitalHouse >
Coding School



DigitalHouse >
Coding School

```
class MainActivity : AppCompatActivity() {

    private val button: Button by lazy { findViewById<Button>(R.id.login_button) }
    private lateinit var callbackManager: CallbackManager

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        callbackManager = CallbackManager.Factory.create()

        button.setOnClickListener { loginFacebook() }
    }

    private fun irParaHome(uiid: String) {
        AppUtil.salvarIdUsuario(application.applicationContext, uiid)
        startActivity(Intent(applicationContext, HomeActivity::class.java))
        finish()
    }

    private fun loginFacebook() {
        val instanceFirebase = LoginManager.getInstance()

        instanceFirebase.logInWithReadPermissions( activity: this, listOf("email", "public_profile"))
        instanceFirebase.registerCallback(callbackManager, object : FacebookCallback<LoginResult> {

                override fun onSuccess(loginResult: LoginResult) {
                    val credential: AuthCredential = FacebookAuthProvider.getCredential(loginResult.accessToken.token)
                    FirebaseAuth.getInstance().signInWithCredential(credential).addOnCompleteListener { irParaHome(loginResult.accessToken.userId) }
                }

                override fun onCancel() {
                    Toast.makeText( context: this@MainActivity, text: "Cancelado!", Toast.LENGTH_SHORT).show()
                }

                override fun onError(error: FacebookException) {
                    Toast.makeText( context: this@MainActivity, error.message, Toast.LENGTH_SHORT).show()
                }
            })
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        callbackManager.onActivityResult(requestCode, resultCode, data)
    }
```

Inicializamos o CallbackManager

Solicitamos permissão para o perfil público

Registramos o callbackManager com um listener para sucesso ou erro

Verificamos a resposta do facebook no onActivityResult

DigitalHouse > Coding School

```
object AppUtil {

    fun salvarIdUsuario(context: Context, uiid: String?) {
        val preferences: SharedPreferences =
            context.getSharedPreferences( name: "APP", Context.MODE_PRIVATE)
        preferences.edit().putString("UIID", uiid).apply()
    }
}
```

DigitalHouse >
Coding School