



# **Lập trình Android**

**Bài 32: *Tích hợp Mạng xã hội***

**Phòng LT & Mạng**

*<http://csc.edu.vn/lap-trinh-va-cSDL>*

**2019**





# Nội dung

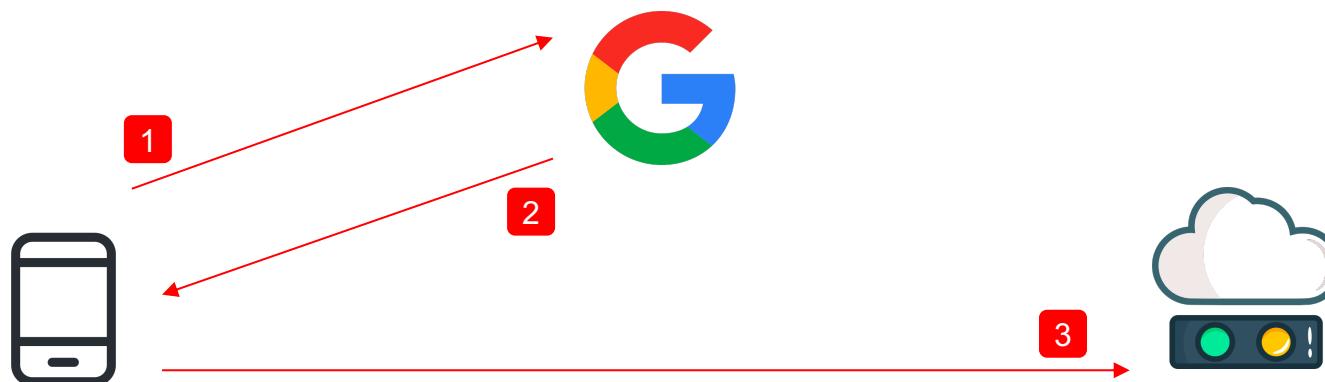
---

1. Tích hợp với Google
2. Tích hợp với Facebook
3. Tích hợp với YouTube



# Tích hợp với Google

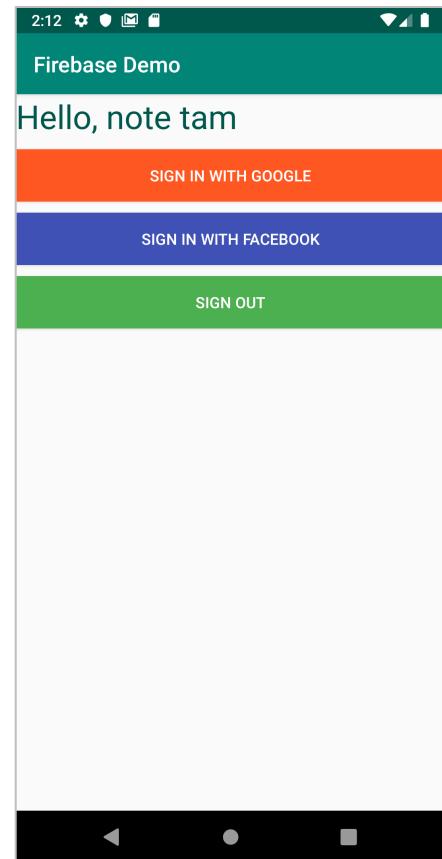
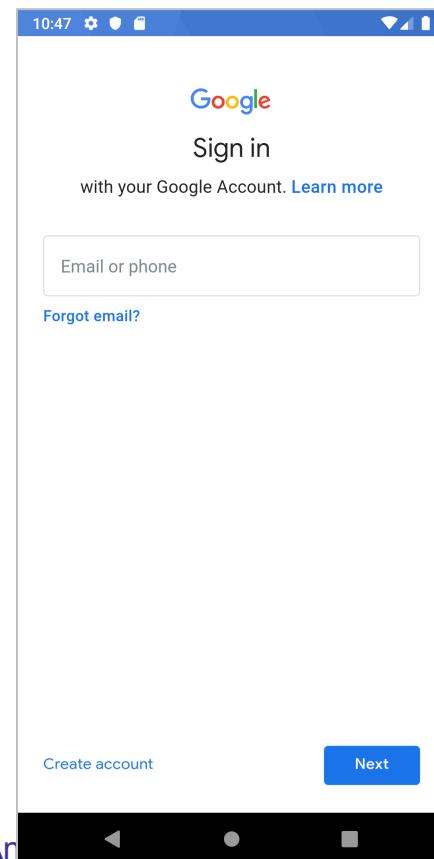
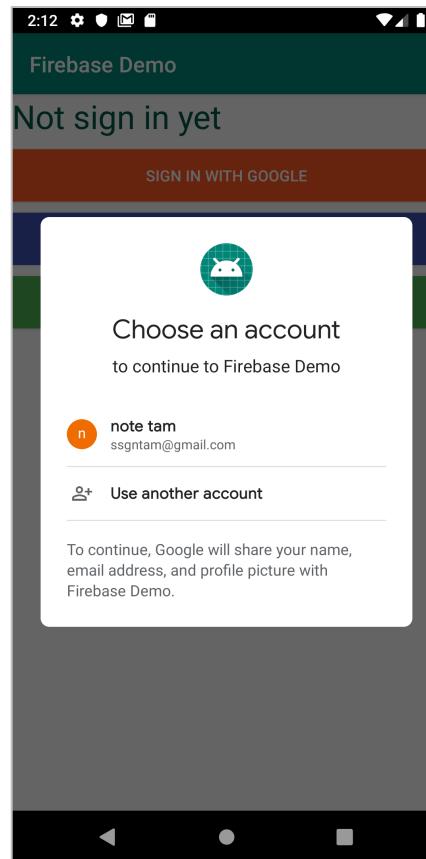
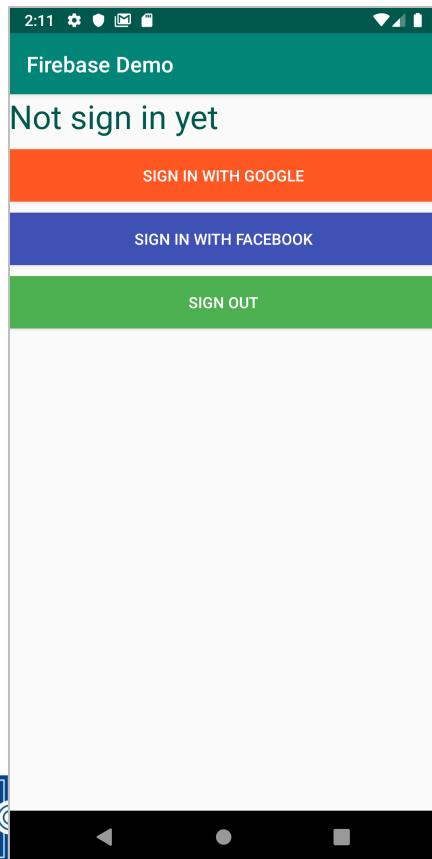
- Ứng dụng Android client có thể tích hợp tính năng **Google Sign In** để có thể đăng nhập vào một hệ thống thông qua tài khoản **Google**. Kịch bản như sau:
  - (1) Tại ứng dụng, người dùng chọn nút “Đăng nhập thông qua Google”
  - (2) Google xác thực đăng nhập và trả về thông tin người dùng cho ứng dụng
  - (3) Ứng dụng có thể gửi thông tin người dùng lên backend server để lưu trữ nếu có tích hợp với back-end server
  - → Như vậy, phía back-end không cần xác thực người dùng do Google đã thực hiện thay việc đó





# Các bước tích hợp Google Sign In

- o (1) Cấu hình Google Developer Console
- o (2) Cấu hình ứng dụng Android
- o (3) Xử lý yêu cầu đăng nhập với Google
- o (4) Xử lý kết quả đăng nhập với Google





# Cấu hình Google Developer Console

- Đăng nhập **Google Developer Console** <https://console.developers.google.com> với tài khoản Google
- Tạo mới một project hoặc sử dụng project đã có sẵn (nếu đã tạo trước project Firebase, project đó sẽ được liệt kê ở Google Developer Console)
- Mở trang **Credentials** → **Create credentials** → **OAuth Client ID**
  - Chọn **Android**
  - Nhập **tên** cho credential
  - Nhập **SHA-1 key** của ứng dụng theo hướng dẫn
  - Nhập **package name** của ứng dụng
  - Copy **android client id** cho bước tiếp theo
- **Lấy SHA-1 debug key** của ứng dụng:
  - Windows: keytool -list -v -alias androiddebugkey  
-keystore %USERPROFILE%\.android\debug.keystore
  - Linux/Mac: keytool -list -v -alias androiddebugkey  
-keystore ~/.android/debug.keystore

← Create OAuth client ID

For applications that use the OAuth 2.0 protocol to call Google APIs, you can use an OAuth 2.0 client ID to generate an access token. The token contains a unique identifier. See [Setting up OAuth 2.0](#) for more information.

Application type

Web application

Android [Learn more](#)

Chrome App [Learn more](#)

iOS [Learn more](#)

Other

Name [?](#)

Android client 3

Signing-certificate fingerprint

Add your package name and SHA-1 signing-certificate fingerprint to restrict usage to your Android apps [Learn more](#)

Get the package name from your AndroidManifest.xml file. Then use the following command to get the fingerprint:

```
$ keytool -exportcert -keystore path-to-debug-or-production-keystore -list -v
```

12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:AA:BB:CC:DD

Package name

From your AndroidManifest.xml file

com.example

Create Cancel



# Cấu hình ứng dụng Android

- Cấu hình **build.gradle** của **project** sử dụng Google:

```
buildscript {  
    repositories {  
        google()  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:3.5.0'  
        classpath 'com.google.gms:google-services:4.3.1'  
    }  
}  
  
allprojects {  
    repositories {  
        google()  
        jcenter()  
    }  
}
```

- Cấu hình **build.gradle** của **module** sử dụng **Google Play Services**:

```
apply plugin: 'com.android.application'  
//...  
  
dependencies {  
    implementation 'com.google.android.gms:play-services-auth:17.0.0'  
}  
apply plugin: 'com.google.gms.google-services'
```



# Xử lý yêu cầu đăng nhập với Google

- (1) Trong phương thức **onCreate()** (vd của LogInActivity), khởi tạo đối tượng **GoogleSignInOptions**, gọi phương thức **requestIdToken()** với **android\_client\_id** lấy từ **credential** của Google Developer Console:

```
GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken(getString(R.string.android_client_id))
    .requestEmail()
    .build();
```

- (2) Cũng trong **onCreate()**, tạo đối tượng **GoogleSignInClient**

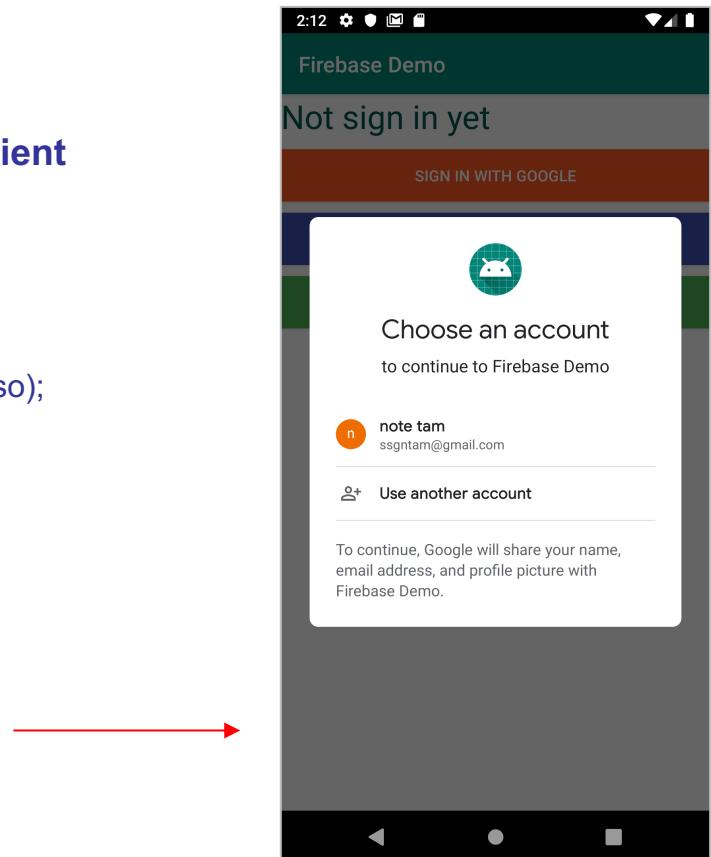
với các options trên:

```
private GoogleSignInClient mGoogleSignInClient;
// ...
mGoogleSignInClient = GoogleSignIn.getClient(this, gso);
```

- (3) Xử lý khi nhấn nút “Đăng nhập với Google”

mở **GoogleSignInIntent**:

```
private void signIn() {
    Intent signInIntent = mGoogleSignInClient.getSignInIntent();
    startActivityForResult(signInIntent, RC_SIGN_IN);
}
```





# Xử lý kết quả đăng nhập với Google

- Override phương thức **onActivityResult()** để xử lý kết quả trả về từ **GoogleSignInIntent**:

```
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    // Kết quả trả về khi thực hiện Intent từ GoogleSignInApi.getSignInIntent(...);
    if (requestCode == RC_SIGN_IN) {
        Task<GoogleSignInAccount> task = GoogleSignIn.getSignedInAccountFromIntent(data);
        try {
            // Google Sign In thành công → lấy thông tin tài khoản Google
            GoogleSignInAccount account = task.getResult(ApiException.class);
            // thực hiện đăng nhập với backend
            handleNextAuthWithGoogleAccount(account);
        } catch (ApiException e) {
            // Google Sign In thất bại → cập nhật UI
            Log.w(TAG, "Google sign in failed", e);
            // ...
        }
    }
}
```





# Đăng nhập với Firebase

- Giả sử backend là **Google Firebase**, có thể xử lý bước đăng nhập tiếp theo với **Firebase Authentication** như sau:

```
private void handleNextAuthWithGoogleAccount(GoogleSignInAccount acct) {  
    AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(), null);  
    mAuth.signInWithCredential(credential)  
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
            @Override  
            public void onComplete(@NonNull Task<AuthResult> task) {  
                if (task.isSuccessful()) {  
                    // đăng nhập với Firebase thành công, cập nhật UI  
                    Log.d(TAG, "signInWithCredential:success");  
                    FirebaseUser user = mAuth.getCurrentUser();  
                    updateUI(user);  
                } else {  
                    // đăng nhập với Firebase thất bại, thông báo đến người dùng  
                    Log.w(TAG, "signInWithCredential:failure", task.getException());  
                    updateUI(null);  
                }  
                // ...  
            }  
        });  
}
```



# Lấy thông tin tài khoản, đăng xuất

- Google Sign In có tính năng ghi nhớ đăng nhập nên có thể kiểm tra user đã đăng nhập chưa ở phương thức `onStart()`:

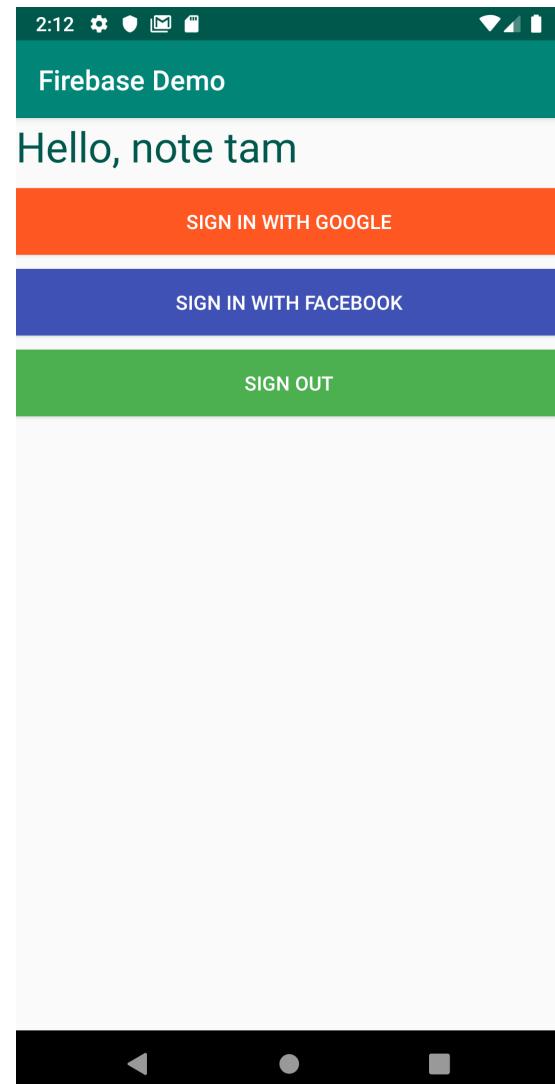
```
GoogleSignInAccount account = GoogleSignIn.getLastSignedInAccount(this);  
updateUI(account);
```

- Có thể lấy được các thông tin tài khoản Google của người dùng:

```
GoogleSignInAccount acct = GoogleSignIn.getLastSignedInAccount(getActivity());  
if (acct != null) {  
    String personName = acct.getDisplayName();  
    String personGivenName = acct.getGivenName();  
    String personFamilyName = acct.getFamilyName();  
    String personEmail = acct.getEmail();  
    String personId = acct.getId();  
    Uri personPhoto = acct.getPhotoUrl();  
}
```

- Đăng xuất tài khoản Google khỏi ứng dụng:

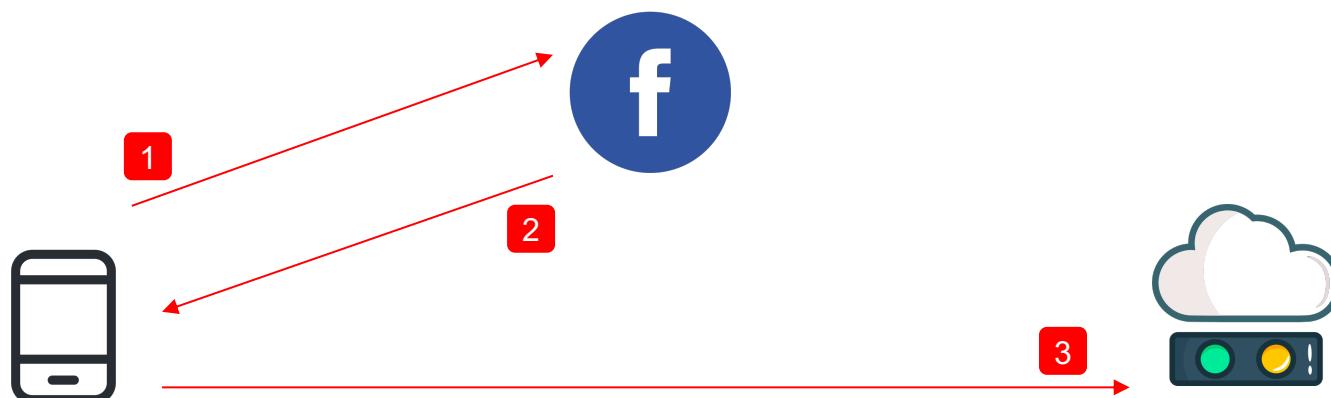
```
private void signOut() {  
    mGoogleSignInClient.signOut()  
        .addOnCompleteListener(this, new OnCompleteListener<Void>() {  
            @Override  
            public void onComplete(@NonNull Task<Void> task) {  
                // ...  
            }  
        });  
}
```





# Tích hợp với Facebook

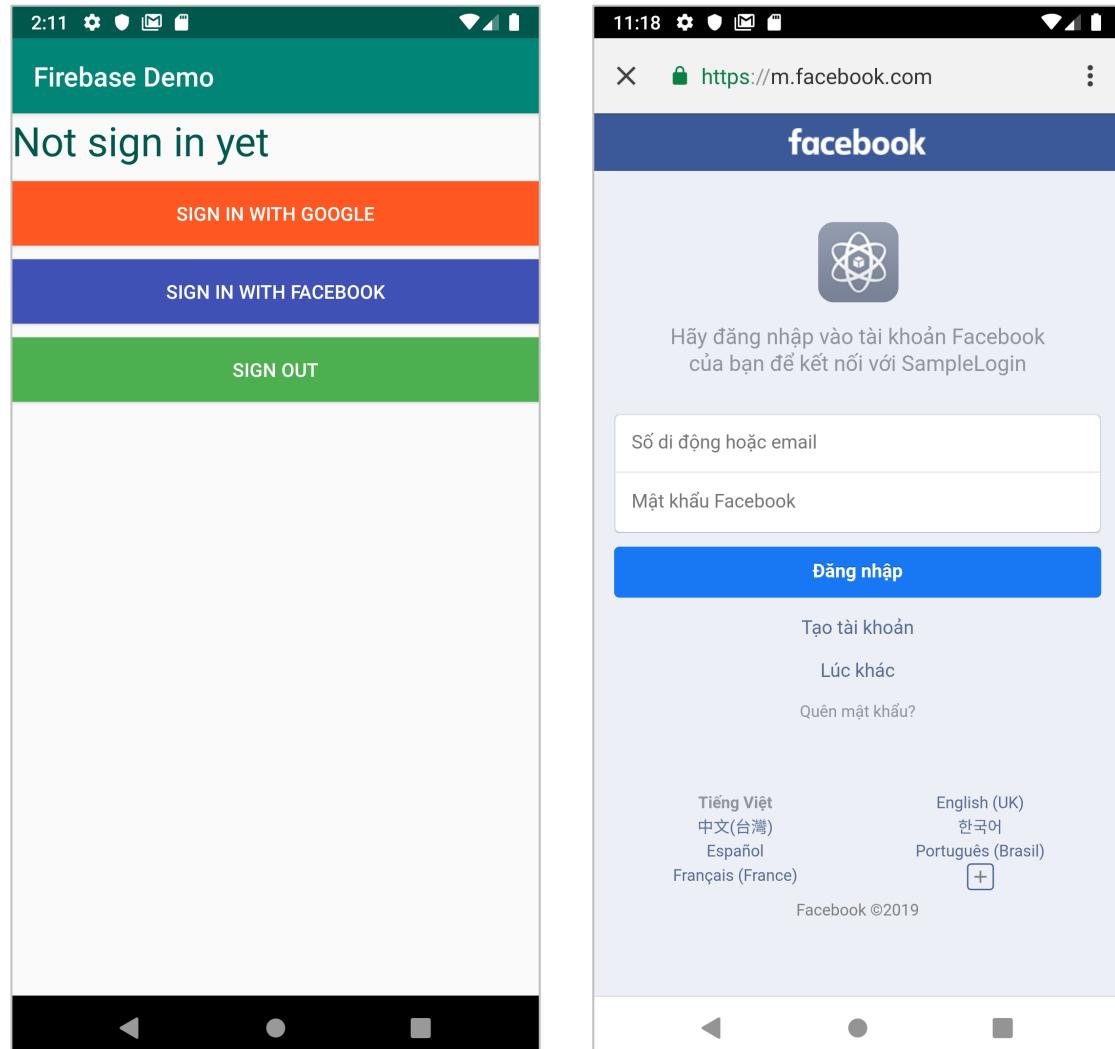
- Tương tự **Google Sign In**, có thể tích hợp **Facebook SDK** vào ứng dụng để có thể đăng nhập vào một hệ thống thông qua tài khoản **Facebook**. Kịch bản như sau:
  - (1) Tại ứng dụng, người dùng chọn nút Đăng nhập thông qua Facebook
  - (2) Facebook xác thực đăng nhập và trả về thông tin người dùng cho ứng dụng
  - (3) Ứng dụng có thể gửi thông tin người dùng lên backend server để lưu trữ nếu có tích hợp với back-end server
  - → Như vậy, phía back-end không cần xác thực người dùng do Facebook đã thực hiện việc đó





# Các bước tích hợp với Facebook SDK

- o (1) Cấu hình Facebook Developer Console
- o (2) Cấu hình ứng dụng Android
- o (3) Xử lý yêu cầu đăng nhập
- o (4) Xử lý kết quả đăng nhập





# Cấu hình Facebook Developer Console

- Đăng nhập <https://developers.facebook.com/> với tài khoản Facebook
- Chọn menu “My Apps” → Add New → Nhập thông tin của ứng dụng

The screenshot shows two overlapping browser windows. The top window is the Facebook Developer Console homepage, featuring a large '5' logo and the text 'Join us at this year's Oculus Developer Conference'. The 'My Apps' menu is open, showing two existing apps: 'MyLogin' and 'SampleLogin'. A red arrow points to the 'Add New App' button. The bottom window is a modal dialog titled 'Create a New App ID', which is a step in the 'Add New App' process. It contains fields for 'Display Name' (set to 'SampleLogin') and 'Contact Email'. At the bottom of the modal, a note says 'By proceeding, you agree to the Facebook Platform Policies' and contains 'Cancel' and 'Create App ID' buttons.





# Cấu hình Facebook Developer Console (2)

- Chọn platform là Android và nhập các thông tin của ứng dụng Android gồm: Google Play Package Name, Class Name, Key Hashes. Copy thông tin App ID cho bước sau.

The screenshot shows the Facebook Developer Console interface. In the center, a modal dialog titled 'Select Platform' is open, listing various platforms: Facebook Web Games, Website, iOS, Android, Windows App, Page Tab, Xbox, and PlayStation. The 'Android' option is highlighted with a red arrow pointing to it. To the right of the dialog, the main settings page for an app with ID '2228061257413571' is visible. The status is set to 'In Development'. The 'Package Name' field contains 'edu.csc.firebaseio'. The 'Default Activity Class Name' field contains 'edu.csc.firebaseio.MainActivity'. The 'Key Hashes' field contains 'gJU...'. The top navigation bar includes links for Docs, Tools, Support, and My Apps, along with a search bar and a 'View Analytics' button.

- Key Hash:

- Windows:

```
keytool -exportcert -alias androiddebugkey -keystore %USERPROFILE%\.android\debug.keystore | openssl sha1 -binary | openssl base64
```



- Linux/Mac:

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



# Cấu hình ứng dụng Android

---

- Cấu hình **build.gradle** của **project**:

```
buildscript {  
    repositories {  
        // ...  
        mavenCentral()  
    }  
}
```

- Cấu hình dependencies cho **build.gradle** của **module**:

```
implementation 'com.facebook.android:facebook-android-sdk:[5,6)'
```

- Khai báo Facebook App ID tại **res/values/strings.xml**:

```
<string name="facebook_app_id">1294553234040192</string>  
<string name="fb_login_protocol_scheme">fb1294553234040192</string>
```

- Khai báo sử dụng permission Internet ở **AndroidManifest.xml**

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



# Cấu hình ứng dụng Android (2)

- Cấu hình **AndroidManifest.xml**, thêm thẻ **meta-data** và **activity** sau vào thẻ **application**:

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

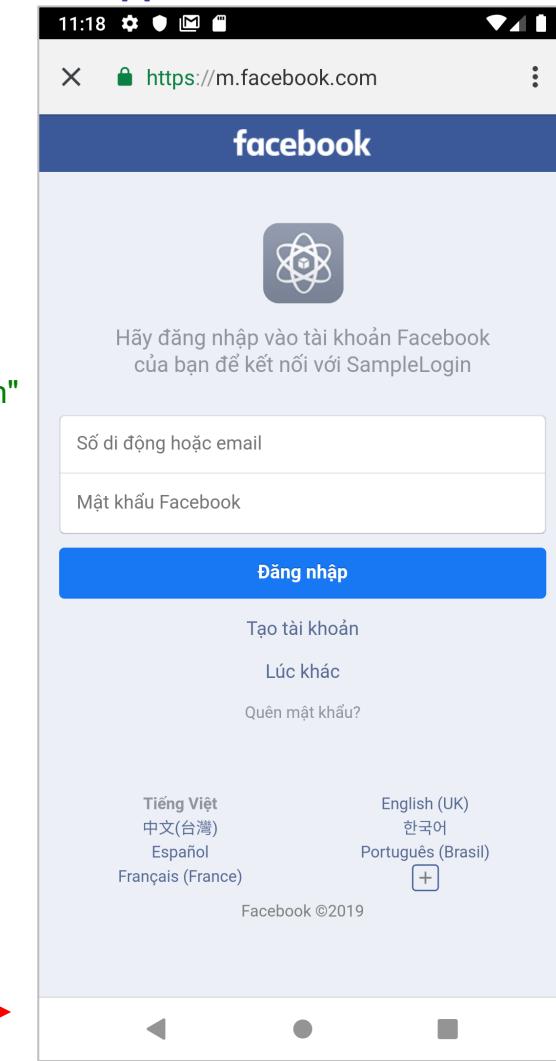
<activity
    android:name="com.facebook.FacebookActivity"
    android:configChanges="keyboard|keyboardHidden|screenLayout|screenSize|orientation"
    android:label="@string/app_name" />

<activity
    android:name="com.facebook.CustomTabActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />

        <data android:scheme="@string/fb_login_protocol_scheme" />
    </intent-filter>
</activity>
</application>
```

**CustomTabActivity** cho phép người dùng  
đăng nhập **Facebook** nếu device chưa cài  
đặt **Facebook app**





# Khởi tạo Facebook SDK

---

- Có thể khởi tạo **Facebook SDK** một lần cho toàn bộ ứng dụng tại class Application:

```
public class App extends Application {  
    @Override  
    public void onCreate() {  
        super.onCreate();  
  
        FacebookSdk.sdkInitialize(this);  
        AppEventsLogger.activateApp(this);  
    }  
}
```



# Xử lý yêu cầu đăng nhập Facebook

- Khai báo một **CallbackManager** để xử lý yêu cầu đăng nhập và kết quả đăng nhập, khởi tạo ở phương thức **onCreate()**:

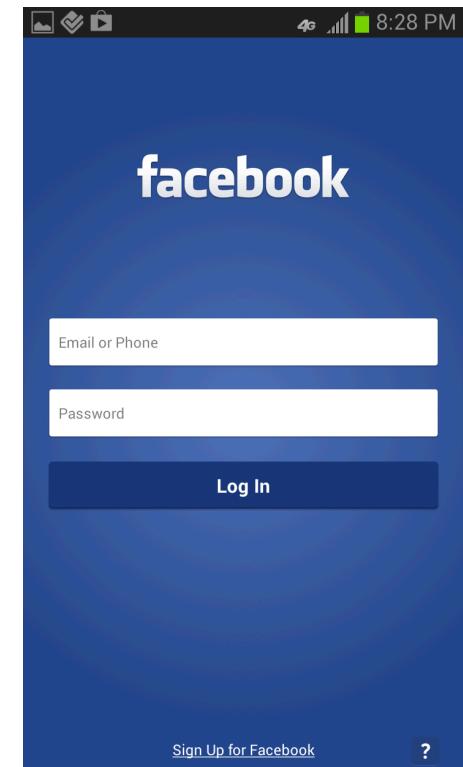
```
private CallbackManager callbackManager;
```

```
// ...
```

```
callbackManager = CallbackManager.Factory.create();
LoginManager.getInstance().registerCallback(callbackManager,
    new FacebookCallback<LoginResult>() {
        @Override
        public void onSuccess(LoginResult loginResult) {
            // getFacebookAccountInfo(loginResult.getAccessToken());
        }

        @Override
        public void onCancel() {
        }

        @Override
        public void onError(FacebookException exception) {
        }
    });
});
```



- Yêu cầu đăng nhập có thể gọi phương thức sau (vd: click lên button “Đăng nhập với Facebook”). Nếu device cài đặt sẵn **Facebook app**, thì **FacebookActivity** sẽ được mở, ngược lại **CustomTabActivity** sẽ được mở (xem khai báo 2 activity này tại **AndroidManifest.xml**):

```
LoginManager.getInstance().logInWithReadPermissions(this, Arrays.asList("email", "public_profile"));
```



# Xử lý kết quả đăng nhập Facebook

- Override phương thức **onActivityResult()** để xử lý kết quả trả về từ **Facebook Intent**:

```
@Override
```

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    callbackManager.onActivityResult(requestCode, resultCode, data);  
    super.onActivityResult(requestCode, resultCode, data);  
}
```

- Facebook Sign In** giữ trạng thái đăng nhập của người dùng, do vậy có thể kiểm tra user đã đăng nhập trên ứng dụng hay chưa ở phương thức **onStart()** bằng cách lấy và kiểm tra thông tin **access token**:

```
AccessToken accessToken = AccessToken.getCurrentAccessToken();  
boolean isLoggedIn = accessToken != null && !accessToken.isExpired();
```



# Lấy thông tin tài khoản Facebook

- Dựa trên **access token** có được, có thể lấy thông tin tài khoản **Facebook** như sau:

```
private void getFacebookAccountInfo(AccessToken accessToken) {  
    // tạo đối tượng GraphRequest  
    GraphRequest request = GraphRequest.newMeRequest(accessToken, new GraphRequest.GraphJSONObjectCallback() {  
        @Override  
        public void onCompleted(JSONObject object, GraphResponse response) {  
            try { // xử lý kết quả trả về sau khi request thực hiện xong  
                String name = object.getString("name");  
                String email = object.getString("email");  
                String image = object.getJSONObject("picture").getJSONObject("data").getString("url");  
                displayName.setText(name);  
                emailID.setText(email);  
            } catch (JSONException e) {  
                e.printStackTrace();  
            }  
        }  
    });  
}
```

```
Bundle parameters = new Bundle(); // tạo bundle để thiết lập các tham số yêu cầu tới Facebook server  
parameters.putString("fields", "id,name,email,picture.width(200)");  
request.setParameters(parameters);  
request.executeAsync(); // thực thi yêu cầu lấy thông tin tài khoản
```





# Đăng xuất tài khoản Facebook khỏi ứng dụng

---

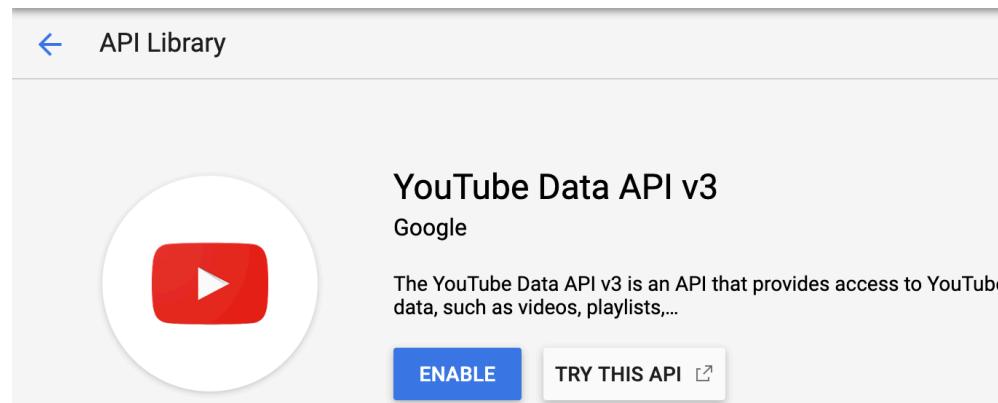
- Để đăng xuất tài khoản **Facebook** khỏi ứng dụng, gọi phương thức sau:

```
LoginManager.getInstance().logOut();
```

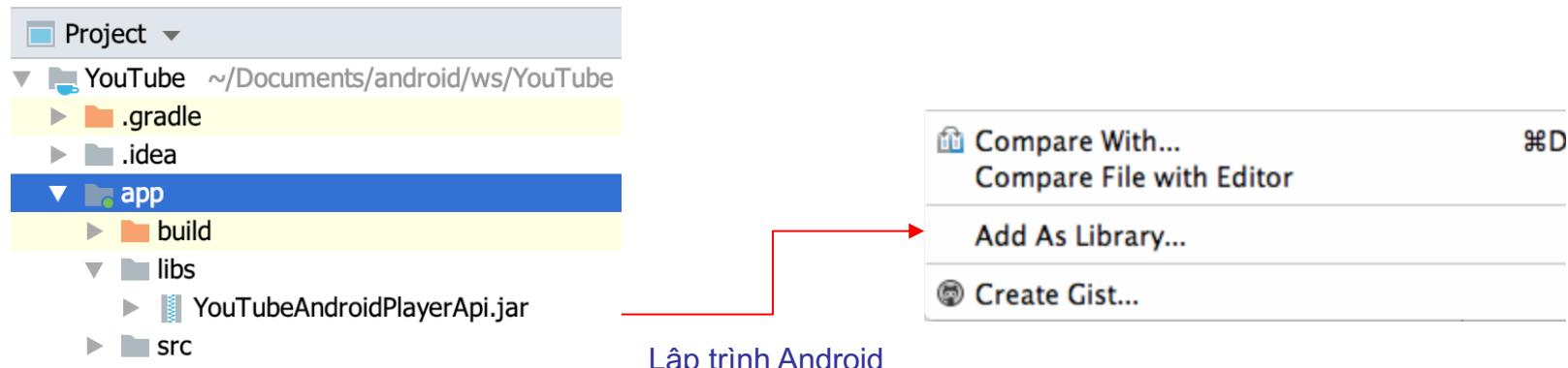


# Tích hợp với YouTube

- Google cung cấp API để **streaming video** từ **YouTube**. Để sử dụng API này, cần enable **“YouTube Data API v3”** tại Google Developer Console (<https://console.developers.google.com>) → Credentials → Enable API



- Tiếp tục, tải về thư viện **YouTubeAndroidPlayerApi** tại <https://developers.google.com/youtube/android/player/downloads/> → Copy vào thư mục **libs** của project → Right click → Add as Library





# Các bước tích hợp YouTube Player API (1)

- o (1) Khai báo thẻ **YouTubePlayerView** trong XML layout, thay vì **VideoView**:

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity">  
  
    <com.google.android.youtube.player.YouTubePlayerView  
        android:id="@+id/viewPlayer"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content" />  
  
</LinearLayout>
```

- o (2) Activity sử dụng XML layout này cần kế thừa từ **YouTubeBaseActivity**:

```
public class VideoActivity extends YouTubeBaseActivity
```



# Các bước tích hợp YouTube Player API (2)

- (3) Khởi tạo **YouTubePlayerView** với API key lấy từ **Google Developer Console**

```
YouTubePlayerView youTubePlayerView = findViewById(R.id.viewPlayer);
```

```
youTubePlayerView.initialize("CHÈN API KEY VÀO ĐÂY",  
    new YouTubePlayer.OnInitializedListener() {  
        @Override  
        public void onInitializationSuccess(YouTubePlayer.Provider provider,  
            YouTubePlayer youTubePlayer, boolean b) {  
            Log.i("CSC", "success");  
            youTubePlayer.cueVideo("6L-pwD21yb4");  
        }  
    }
```

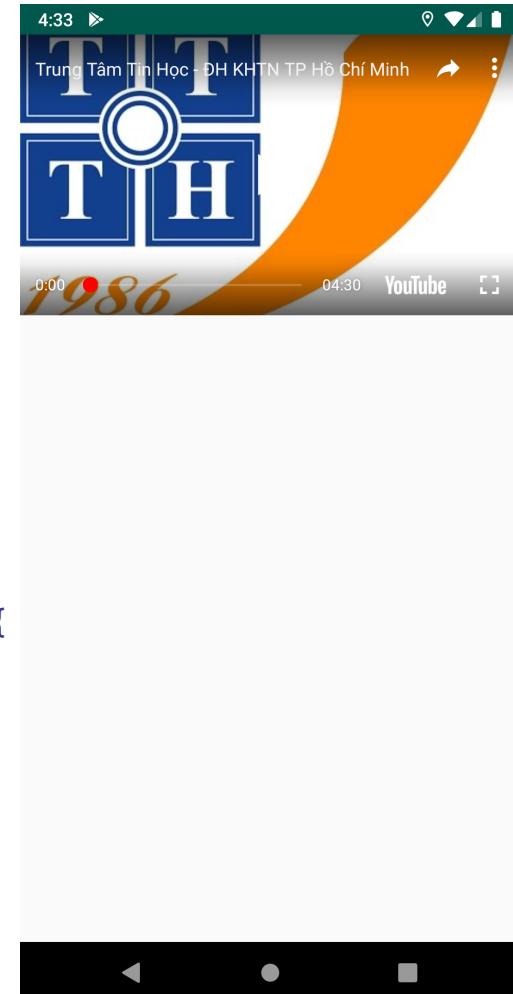
```
@Override  
public void onInitializationFailure(YouTubePlayer.Provider provider,  
    YouTubeInitializationResult youTubeInitializationResult) {  
    Log.i("CSC", "failure");  
}  
});
```

- Sau khi khởi tạo xong player view, có thể tải về video, hiển thị và play

tự động video đó bởi phương thức **loadVideo("video\_id")** hoặc chỉ tải về

và hiển thị bởi phương thức **cueVideo("video\_id")**. Lưu ý: video\_id không bao gồm toàn bộ URL

<https://www.youtube.com/watch?v=6L-pwD21yb4>





# Các bước tích hợp YouTube Player API (3)

- Để video luôn ở chế độ **landscape**, điều chỉnh thuộc tính **screenOrientation** của activity trong file **AndroidManifest.xml**

```
<activity android:name=".MainActivity"  
        android:screenOrientation="landscape">
```

```
</activity>
```



- Lưu ý, cần cấp **permission** cho phép ứng dụng sử dụng **Internet**:

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





# Sử dụng YouTubePlayerFragment

- Nếu không muốn kế thừa activity từ **YouTubeBaseActivity**, có thể sử dụng **YouTubePlayerFragment**:

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/youtubeFragment"
    android:name="com.google.android.youtube.player.YouTubePlayerFragment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

- Quá trình khởi tạo và streaming video **YouTubePlayerFragment** tương tự **YouTubePlayerView**:

```
YouTubePlayerFragment youtubeFragment = (YouTubePlayerFragment)
    getFragmentManager().findFragmentById(R.id.youtubeFragment);
youtubeFragment.initialize("CHÈN API KEY VÀO ĐÂY",
    new YouTubePlayer.OnInitializedListener() {
        @Override
        public void onInitializationSuccess(YouTubePlayer.Provider provider,
            YouTubePlayer youTubePlayer, boolean b) {
            youTubePlayer.loadVideo("6L-pwD21yb4");
        }
        @Override
        public void onInitializationFailure(YouTubePlayer.Provider provider,
            YouTubInitializationResult youTubInitializationResult) {
        }
    });
});
```





## Q&A

---

