

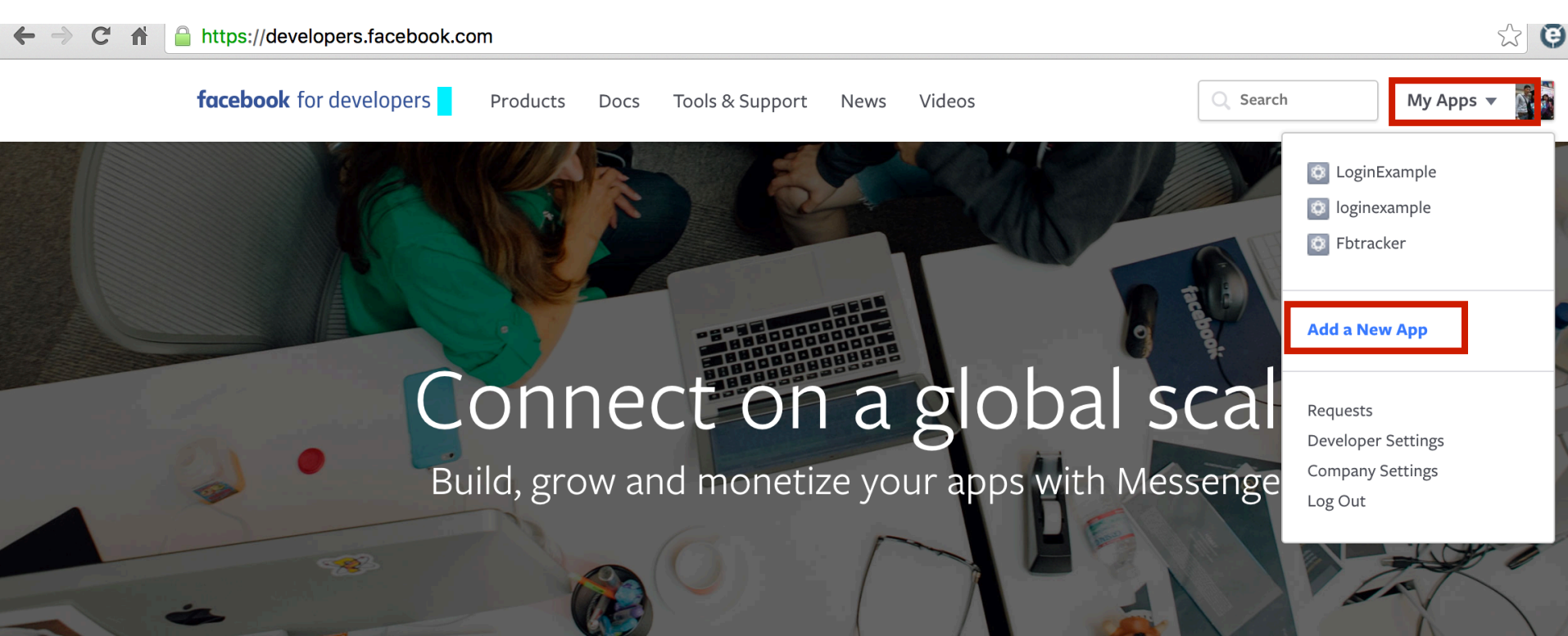
Android-Facebook API

Facebook SDK

- The Facebook SDK for Android is the easiest way to integrate your Android app with Facebook
- It enables:
 - Facebook Login - Authenticate people with their Facebook credentials
 - Share and Send dialogs - Enable sharing content from your app to Facebook
 - App Events - Log events in your application.
 - Graph API - Read and write to Graph API
- Two ways to set up your app to use the Facebook SDK:
 - Using **Quick Start**:
<https://developers.facebook.com/quickstarts/?platform=android>
 - Setting up your project with the Facebook SDK

Add Facebook SDK to Your Project

- Registered as facebook developer:
<https://developers.facebook.com/>
- Create a new App and get the App ID:
<https://developers.facebook.com/apps/>



The screenshot shows the Facebook Developers website. The browser address bar displays <https://developers.facebook.com>. The navigation bar includes the 'facebook for developers' logo, links for 'Products', 'Docs', 'Tools & Support', 'News', and 'Videos', a search bar, and a 'My Apps' dropdown menu. The 'My Apps' menu is open, showing a list of existing apps: 'LoginExample', 'loginexample', and 'Fbtracker'. A red box highlights the 'Add a New App' button. Below the menu, the text 'Connect on a global scale' and 'Build, grow and monetize your apps with Messenger' is overlaid on a background image of people working at a desk.

facebook for developers | Products | Docs | Tools & Support | News | Videos | Search | My Apps

- LoginExample
- loginexample
- Fbtracker
- Add a New App**
- Requests
- Developer Settings
- Company Settings
- Log Out

Connect on a global scale
Build, grow and monetize your apps with Messenger

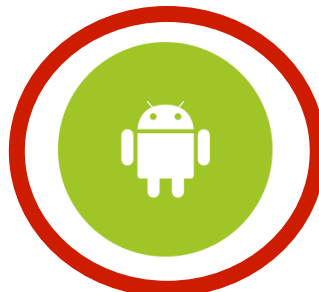
Add Facebook SDK to Your Project

Add a New App

Select a platform to get started



iOS



Android



Facebook Canvas



Website



Start Over

Skip and Create App ID

3



Quick Start for Android

LoginExamp|

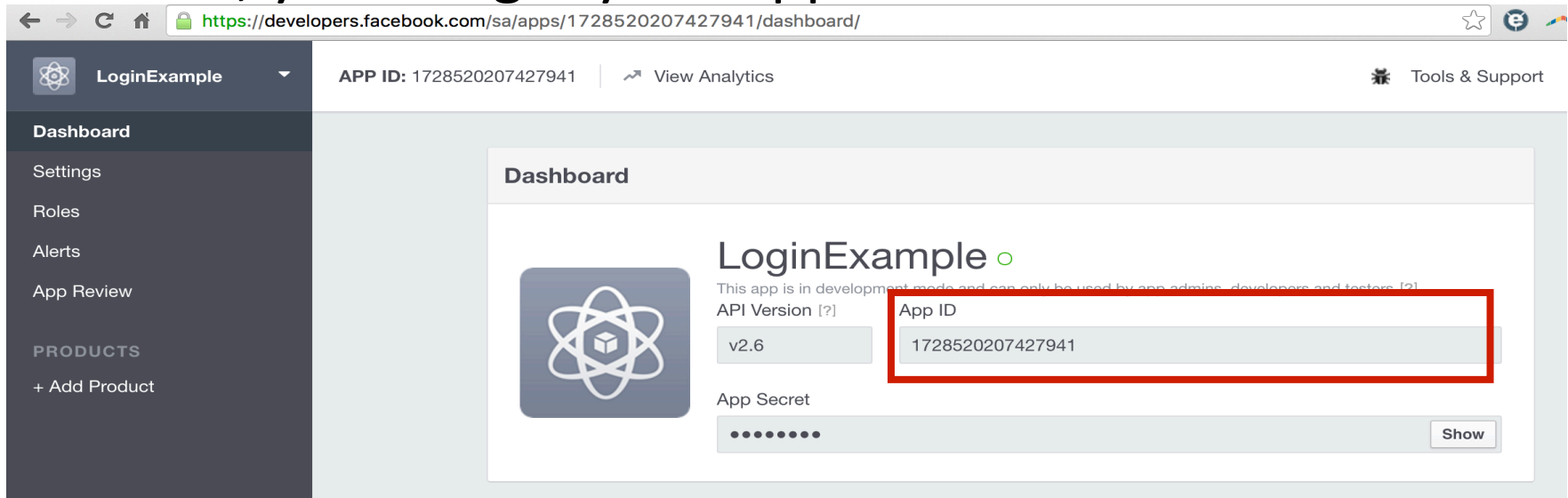
1

Create New Facebook App ID

2

Add Facebook SDK to Your Project

- Then, you will get your App ID



- To use Facebook SDK in a project, add it as a build dependency and import it
 - If you are starting a new project, follow all the steps in the next slide
 - To add Facebook SDK to an existing project, start with step 3

Steps

1. Go to **Android Studio -> New Project -> Minimum SDK**
2. Select "**API 15: Android 4.0.3**" or higher and create your new project
3. In your project, open **your_app -> Gradle Scripts -> build.gradle (module level)**
4. Add the Maven Central Repository to **build.gradle** before **dependencies**:
5. Add **compile 'com.facebook.android:facebook-android-sdk:[4,5)'** to your **build.gradle** dependencies
6. Build your project
7. Import Facebook SDK into your app:

```
repositories {  
    mavenCentral()  
}
```

```
import com.facebook.FacebookSdk;
```

Initialize Facebook SDK

- You need to initialize Facebook SDK before you can use it
- Add a call to **FacebookSdk.sdkInitializefrom** from **onCreate** in your Application class:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    FacebookSdk.sdkInitialize(getApplicationContext());
    AppEventsLogger.activateApp(this.getApplication());
}
```

Add Facebook App ID

1. Open your **strings.xml** file, for example: **/app/src/main/res/values/strings.xml**
2. Add a new string with the name **facebook_app_id** containing the value of your Facebook App ID:

```
<string name="facebook_app_id">xxxxxxxxxxxxxxxxxx</string>
```

Get from slides #5

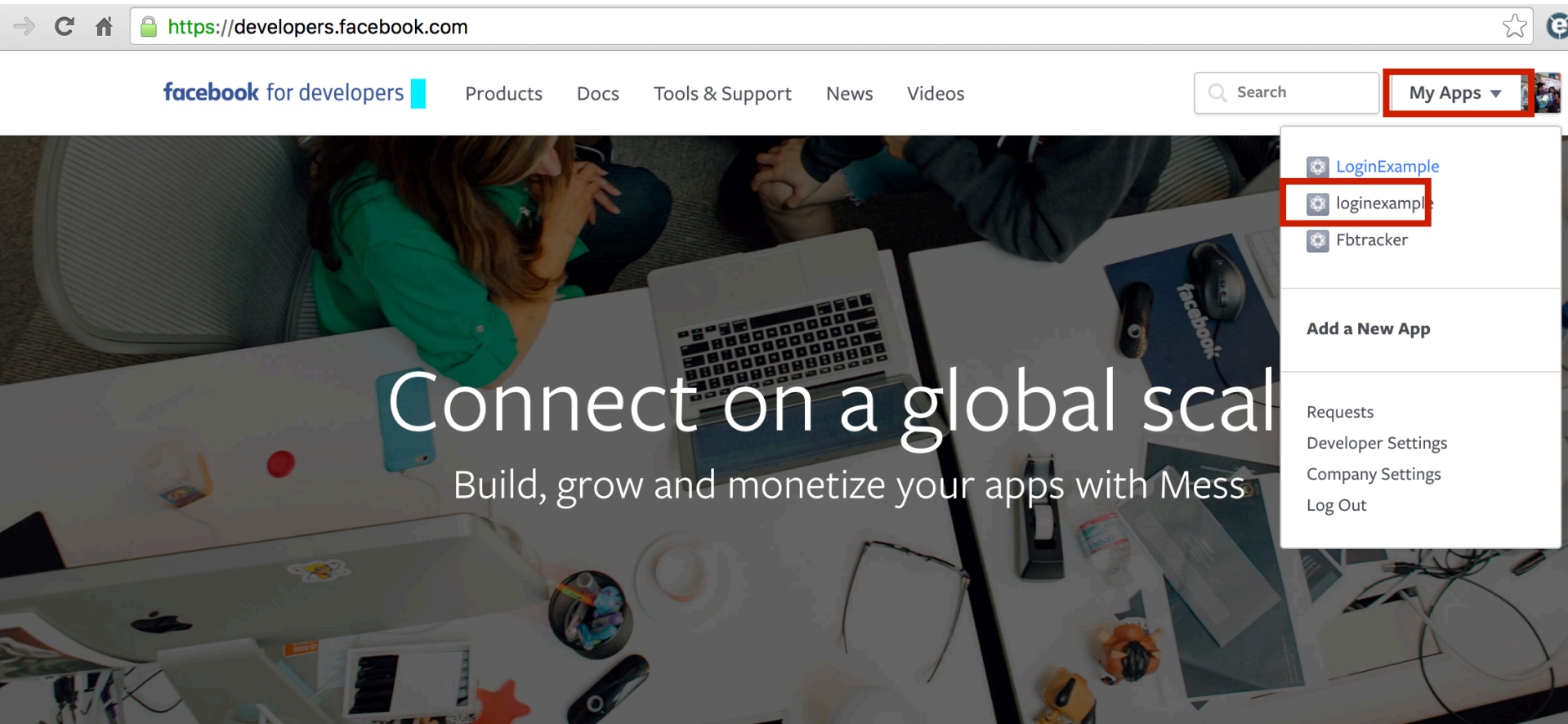
3. Add a **uses-permission** element to the manifest

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

4. Add the **meta-data** element to the **application** element:

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


App Setting on Developer Website



The image shows a browser window with the URL <https://developers.facebook.com>. The navigation bar includes the Facebook logo, the text "facebook for developers", and links for "Products", "Docs", "Tools & Support", "News", and "Videos". A search bar is present, and the "My Apps" dropdown menu is open, showing a list of applications: "LoginExample", "loginexample", and "Fbtracker". The "loginexample" item is highlighted with a red box. Below the list is a section for "Add a New App" and a list of links: "Requests", "Developer Settings", "Company Settings", and "Log Out".

facebook for developers | Products | Docs | Tools & Support | News | Videos

Search | My Apps

- LoginExample
- loginexample
- Fbtracker

Add a New App

Requests
Developer Settings
Company Settings
Log Out

Connect on a global scale
Build, grow and monetize your apps with Mess

Tell Facebook About Your Project

The screenshot shows the Facebook Developer console interface. On the left is a dark sidebar with navigation options: Dashboard, Settings (highlighted with a red box), Basic (selected), Advanced, Roles, Alerts, App Review, and PRODUCTS (+ Add Product). The top header displays 'LoginExample' with a dropdown arrow, 'APP ID: 1728520207427941', and a 'View Analytics' link. On the right, there is a 'Tools & Support' link.

The main content area is titled 'Basic' and contains the following settings:

- App ID:** [Empty text input field]
- App Secret:** [Masked text input field with 7 dots] and a 'Show' button.
- Display Name:** LoginExample
- Namespace:** [Empty text input field]
- App Domains:** [Empty text input field]
- Contact Email:** yibin1220@gmail.com
- Privacy Policy URL:** Privacy policy for Login dialog and App Details
- Terms of Service URL:** Terms of Service for Login dialog and App Details
- App Icon:** A placeholder image showing a blue square with a white plus sign and the text '1024 x 1024' below it.
- Category:** Apps for Pages (dropdown menu)

At the bottom of the page, there is a 'Quick Start' button and a partially visible 'Android' tab.

https://developers.facebook.com/apps/你的app_id/settings/

Tell Facebook About Your Project

Facebook Developer Console interface for app configuration.

APP ID: 1728520207427941 | [View Analytics](#) | [Tools & Support](#)

App Name: LoginExample

Category: Apps for Pages

Image: 1024 x 1024

Android Configuration:

- Google Play Package Name:** com.example.nmsl.loginexample
- Class Name:** com.example.nmsl.loginexample.MainActivity
- Key Hashes:** Put your package path Put your activity name
- Amazon Appstore URL (Optional):** Ex. http://www.amazon.com/dp/B004GJDQT8
- Need key hashes:** (Text annotation)
- Single Sign On:** Yes (Will launch from Android Notifications)
- Deep Linking:** Yes (News Feed links launch this app)

Add Your Development Key Hashes

- Generate the key hashes

- Mac:

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

Password: android

- Windows:

navigate to the jre-xx/bin folder

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

May don't have openssl.exe -> donwload from

<http://downloads.sourceforge.net/gnuwin32/openssl-0.9.8h-1-bin.zip>

-> put the bin/openssl.exe to jre-xxx/bin/

Add Your Development Key Hashes

- Put the generated key onto the web and save changes

The screenshot shows the Firebase console interface for an app named "LoginExample". The top navigation bar includes the app name, APP ID (1728520207427941), and a "View Analytics" link. A sidebar on the left contains navigation options: Dashboard, Settings (with "Basic" selected), Roles, Alerts, App Review, and PRODUCTS (+ Add Product). The main content area is titled "Android" and contains configuration fields for Google Play Package Name (com.example.nmsl.loginexample), Class Name (com.example.nmsl.loginexample.MainActivity), and Key Hashes. A red box highlights the first key hash: 2Qmdr/LxJDPJ3YqeBcPxPFleOKY=. Below this, there is a field for Amazon Appstore URL (Optional) with the example http://www.amazon.com/dp/B004GJDQT8. At the bottom, there are two toggle switches: "Single Sign On" (Yes) and "Deep Linking" (Yes).

Dashboard

Settings

Basic

Advanced

Roles

Alerts

App Review

PRODUCTS

+ Add Product

APP ID: 1728520207427941 | View Analytics

Tools & Support

Apps for Pages

1024 x 1024

Android Quick Start

Google Play Package Name

Class Name

com.example.nmsl.loginexample

com.example.nmsl.loginexample.MainActivity

Key Hashes

2Qmdr/LxJDPJ3YqeBcPxPFleOKY=

Put your key hashes

Amazon Appstore URL (Optional)

Ex. http://www.amazon.com/dp/B004GJDQT8

Yes Single Sign On Will launch from Android Notifications

Yes Deep Linking News Feed links launch this app

Implement Facebook Login in Your App

- Add FacebookActivity to the manifest file (for login or share)

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

- Create a button for login in layout

```
<Button android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Login FB"  
    android:id="@+id/login_fb"  
    android:layout_below="@+id/textView"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true" />
```

Implement Facebook Login in Your App (cont.)

```
// callback manager for login process
CallbackManager callbackManager;
// access token may be used for other operations after login
private AccessToken accessToken;

@Override
protected void onCreate(Bundle savedInstanceState) {
    //initialize facebook sdk
    //should be initialized before setContentView
    FacebookSdk.sdkInitialize(getApplicationContext());
    super.onCreate(savedInstanceState); setContentView(R.layout.activity_main);
    //create callback manager
    callbackManager = CallbackManager.Factory.create();
}
```

Implement Facebook Login in Your App (cont.)

- Set onclick listener on the login button

```
Button loginButton = (Button) findViewById(R.id. Login_fb);  
loginButton.setOnClickListener(new Button.OnClickListener(){
```

```
@Override public void onClick(View v) {  
    //login with LoginManager  
    LoginManager.getInstance().loginWithReadPermissions(  
        MainActivity.this, Arrays.asList("public_profile", "user_friends")); } });
```


Implement Facebook Login in Your App (cont.)

- Set onclick listener on the login button

```
//add callback function to LoginManage in inner class
LoginManager.getInstance().registerCallback(callbackManager,
    new FacebookCallback<LoginResult>() {
```

```
//if login success then get the access token
@Override
public void onSuccess(LoginResult loginResult) {
    //store the access token
    accessToken = loginResult.getAccessToken();
    Log.d("FB", "Access token got.");
```

Implement Facebook Login in Your App (cont.)

```
//send request and call graph api
```

```
GraphRequest request = GraphRequest.newMeRequest(  
    accessToken, new GraphRequest.GraphJSONObjectCallback() {
```

```
    //response
```

```
    @Override
```

```
    public void onCompleted(JSONObject object, GraphResponse response) {
```

```
        //pack the information and send request
```

```
        Bundle parameters = new Bundle();
```

```
        parameters.putString("fields", "id,name,link");
```

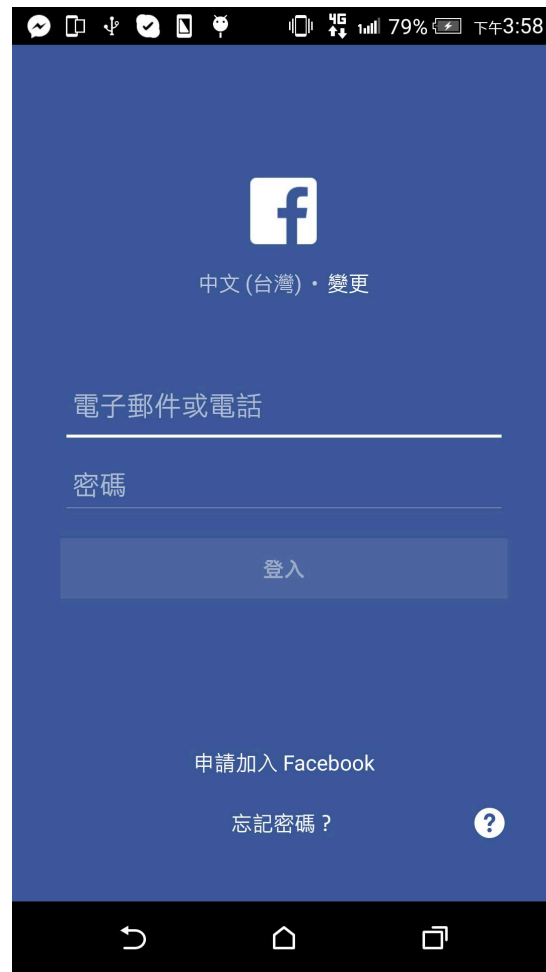
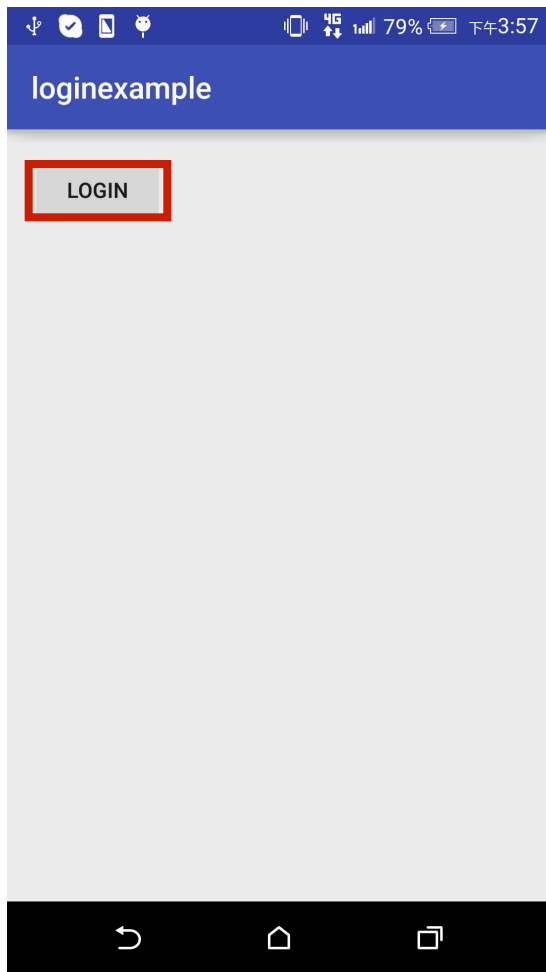
```
        request.setParameters(parameters);
```

```
        request.executeAsync(); }
```

Implement Facebook Login in Your App (cont.)

```
//if login is canceled
@Override
public void onCancel() {
    // App code
    Log.d("FB", "CANCEL"); }
```

```
//if login failed
@Override
public void onError(FacebookException exception) {
    // App code
    Log.d("FB", exception.toString()); } };
```



Facebook SDK

- Login - Sign in to your app with their Facebook login
- Share - [share](#), [send a message](#) and [like content](#)
- Graph API - Get data in and out of Facebook's social graph (Query data, post new stories, upload photos, etc)
- App Links - Link posts, stories and requests shared from your app back to your app
- App Invites - Send invites
- App Events - Understand people's actions in your app

Sample Codes

- <https://github.com/facebook/facebook-android-sdk>
- HellofacebookSample – Shows profile access, status updates and photo upload
- Iconicus – Little game with send/share button
- MessengerSendSample – Send a picture to friends
- RPSSample – Use Native Share Dialog, Open Graph publishing, pickers, invites, and deep linking
- Scrumptious - Login, requests, pickers, picture uploads, and Open Graph publishing
- ShareIt – Like/share
- SwitchUserSample – Switch user