



**FPT POLYTECHNIC**

LẬP TRÌNH ANDROID NÂNG CAO

**Bài 4: Camera, Media,  
đồng bộ dữ liệu lên GCM**

---

[www.poly.edu.vn](http://www.poly.edu.vn)

---

## Nội dung bài học

- Camera
- Media
- Google cloud messaging



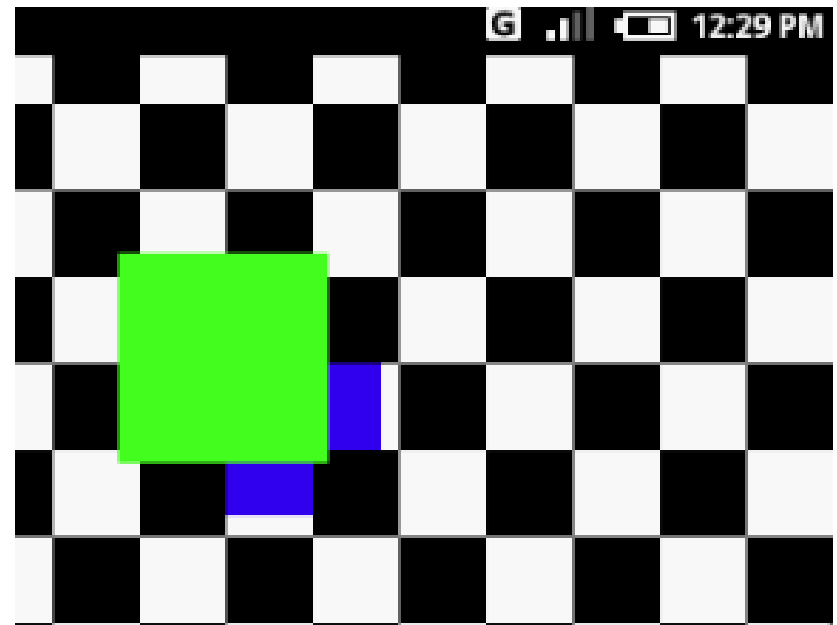
# Camera

- Lớp **Camera** được sử dụng để thiết lập camera, bắt đầu hoặc dừng sử dụng camera và preview ảnh
- Để truy cập camera, bạn phải khai báo quyền CAMERA trong Android Manifest
- Thêm phần tử <uses-feature> để khai báo các tính năng của camera mà ứng dụng hỗ trợ

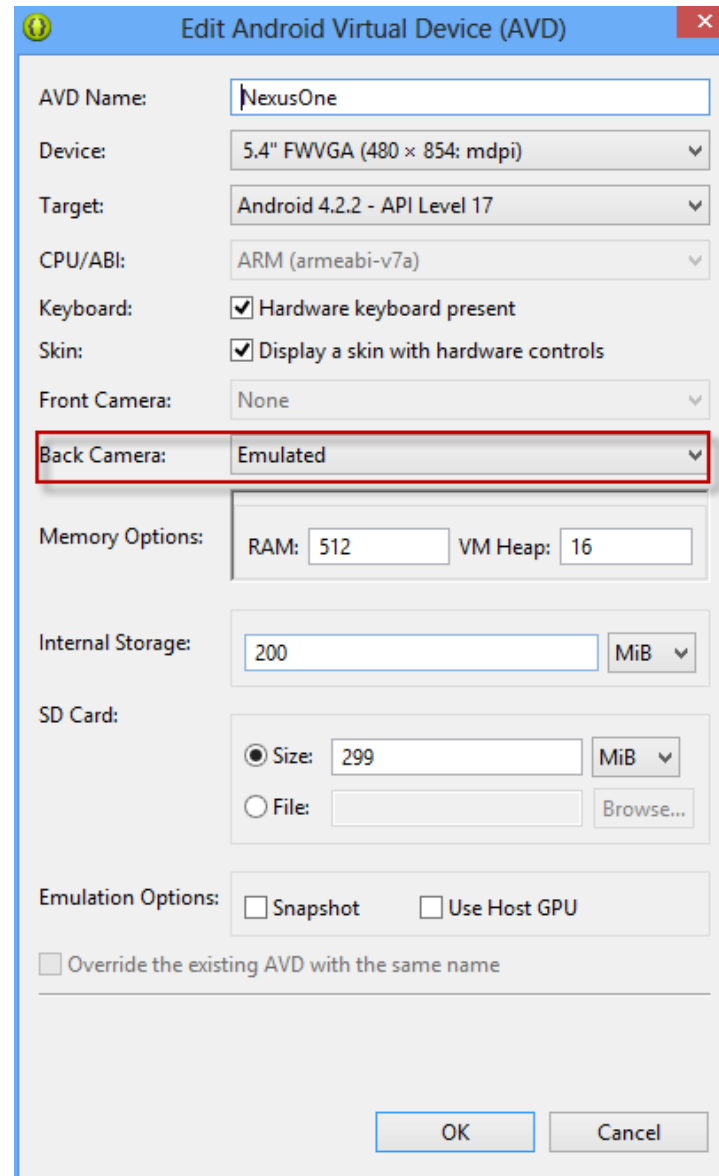
```
<uses-permission android:name="android.permission.CAMERA"/>  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
<uses-feature android:name="android.hardware.camera" android:required="false"/>
```

# Camera

- Camera trên Emulator không hoạt động
  - Preview: di chuyển khối hình
  - Chụp ảnh luôn luôn hiển thị cùng một bức ảnh trắng
- Để test ứng dụng Camera trên Emulator phải kích hoạt camera trên Emulator
- Sau đó vào DDMS của Eclipse để test camera



# Camera



The screenshot shows the 'Edit Android Virtual Device (AVD)' window. The 'Back Camera' dropdown is highlighted with a red rectangle and is set to 'Emulated'. Other settings include AVD Name: NexusOne, Device: 5.4" FWVGA (480 x 854: mdpi), Target: Android 4.2.2 - API Level 17, CPU/ABI: ARM (armeabi-v7a), Keyboard: Hardware keyboard present, Skin: Display a skin with hardware controls, Front Camera: None, Memory Options: RAM: 512, VM Heap: 16, Internal Storage: 200 MiB, SD Card: Size: 299 MiB, Emulation Options: Snapshot and Use Host GPU are unchecked, and the 'Override the existing AVD with the same name' checkbox is also unchecked.

AVD Name:	NexusOne
Device:	5.4" FWVGA (480 x 854: mdpi)
Target:	Android 4.2.2 - API Level 17
CPU/ABI:	ARM (armeabi-v7a)
Keyboard:	<input checked="" type="checkbox"/> Hardware keyboard present
Skin:	<input checked="" type="checkbox"/> Display a skin with hardware controls
Front Camera:	None
Back Camera:	Emulated
Memory Options:	RAM: 512 VM Heap: 16
Internal Storage:	200 MiB
SD Card:	<input checked="" type="radio"/> Size: 299 MiB <input type="radio"/> File: Browse...
Emulation Options:	<input type="checkbox"/> Snapshot <input type="checkbox"/> Use Host GPU
<input type="checkbox"/> Override the existing AVD with the same name	
OK Cancel	

# Camera

DDMS - FpolyCameraApi/src/com/fpolycameraapi/MakePhotoActivity.java - ADT

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Java DDMS

Devices

Name	Online	NexusOn...
emulator-5554	302	8600
com.andr...	381	8616
com.andr...	409	8618
com.andr...	430	8620
com.andr...	443	8622
android.pr...	480	8626
com.andr...	489	8629
com.andr...	532	8635
android.pr...	547	8641
com.andr...	586	8648
com.andr...	620	8657
com.andr...	661	8661
com.andr...	681	8665
com.andr...	709	8668
com.andr...	729	8671
com.andr...	757	8673
com.andr...	780	8675
com.andr...	828	8677
com.svox...	843	8679
com.andr...	865	8681
com.fpoly...	896	8683

Threads Heap Allocation Tracker Network Statistics File Explorer Emulator Control System Information

Name	Size	Date	Time	Permissions	Info
acct		2014-03-19	10:47	drwxr-xr-x	
cache		2014-03-19	10:31	drwxrwx---	
config		2014-03-19	10:47	dr-x-----	
d		2014-03-19	10:47	lrwxrwxrwx	-> /sys/ker...
data		2014-03-19	08:20	drwxrwx--x	
default.prop	116	1970-01-01	00:00	-rw-r--r--	
dev		2014-03-19	10:47	drwxr-xr-x	
etc		2014-03-19	10:47	lrwxrwxrwx	-> /system...
init	109412	1970-01-01	00:00	-rwxr-x---	
init.goldfish.rc	2487	1970-01-01	00:00	-rwxr-x---	
init.rc	18414	1970-01-01	00:00	-rwxr-x---	
init.trace.rc	1795	1970-01-01	00:00	-rwxr-x---	
init.usb.rc	3947	1970-01-01	00:00	-rwxr-x---	
mnt		2014-03-19	10:47	drwxrwxr-x	
asec		2014-03-19	10:47	drwxr-xr-x	
obb		2014-03-19	10:47	drwxr-xr-x	
sdcard		1970-01-01	00:00	d---rwxr-x	
Alarms		2014-03-19	08:20	d---rwxr-x	
Android		2014-03-19	09:27	d---rwxr-x	
DCIM		2014-03-19	08:28	d---rwxr-x	
Download		2014-03-19	08:20	d---rwxr-x	
LOST.DIR		2014-03-19	08:19	d---rwxr-x	
Movies		2014-03-19	08:20	d---rwxr-x	
Music		2014-03-19	08:20	d---rwxr-x	
Notifications		2014-03-19	08:20	d---rwxr-x	
Pictures		2014-03-19	09:05	d---rwxr-x	
CameraAPIDemo		2014-03-19	10:49	d---rwxr-x	
Picture_20140519090516.jpg	16451	2014-03-19	09:05	----rwxr-x	
Picture_20140819090841.jpg	24708	2014-03-19	09:08	----rwxr-x	
Picture_20144919104922.jpg	14512	2014-03-19	10:49	----rwxr-x	

LogCat

Saved Filters + -

All messages (no filters) (1005)

com.fpolycameraapi (Session Filter)

Search for messages. Accepts Java regexes. Prefix with pid; app; tag; or text: to limit scope. verbose

L...	Time	PID	TID	Application	Tag	Text
D	03-19 10:49:2...	896	896	com.fpolycameraapi	MakePhotoActivitv	onPictureTaken

126M of 419M

## Các bước sử dụng Camera

- Sử dụng **open(int)** để nhận instance của Camera
- Lấy thông tin các thiết lập mặc định sử dụng **getParameters()**
- Nếu cần thiết, thay đổi đối tượng trả về Camera.Parameters và gọi **setParameters(Camera.Parameters)**
- Nếu cần thiết, gọi **setDisplayOrientation(int)**
- Quan trọng: truyền SurfaceHolder (đã được khởi tạo đầy đủ) tới **setPreviewDisplay(SurfaceHolder)**. Không có Surface, camera không thể bắt đầu preview

## Các bước sử dụng Camera

- Gọi **startPreview()** để bắt đầu cập nhật preview surface. Preview phải bắt đầu trước khi bạn chụp ảnh
- Khi bạn muốn, gọi **takePicture(Camera.ShutterCallback, Camera.PictureCallback, Camera.PictureCallback)** để capture ảnh. Đợi callback cung cấp ảnh chụp thật
- Sau khi chụp ảnh, preview display sẽ dừng. Để chụp thêm ảnh gọi **startPreview()** lần nữa
- Gọi **stopPreview()** để dừng cập nhật preview surface
- Quan trọng: gọi **release()** để giải phóng camera được sử dụng bởi ứng dụng khác. Ứng dụng nên giải phóng Camera ngay lập tức trong **onPause()** (và mở lại trong **onResume()**)





# Video recording

- Lấy thông tin và khởi tạo Camera và bắt đầu preview như miêu tả ở trên
- Gọi **unlock()** để cho phép media process thao tác với camera
- Truyền camera tới **setCamera(camera)**. Sử dụng **MediaRecorder** để quay phim
- Khi kết thúc quay, gọi **reConnect()** để khóa camera
- Nếu muốn, khởi tạo lại preview và tiếp tục chụp ảnh hoặc quay video
- Gọi **stopPreview()** và **release()** giống như trên

# Playing Audio

- Android cung cấp 2 API để play audio
  - SoundPool
  - MediaPlayer
- SoundPool có thể được sử dụng để play các clip có kích thước nhỏ. Có thể lặp âm thanh và play một số file đồng thời
- SoundPool chỉ nên bật các file âm thanh có kích thước không quá 1MB
- SoundPool tải các file không đồng bộ
- Từ Android API8, có thể kiểm tra việc tải file đã hoàn thành chưa sử dụng OnLoadCompleteListener

# Playing Audio

- Phone volume có thể được cấu hình để điều khiển audio stream, ví dụ trong khi gọi volume button cho phép tăng/giảm dung lượng
- MediaPlayer phù hợp hơn cho các movie và music có dung lượng lớn



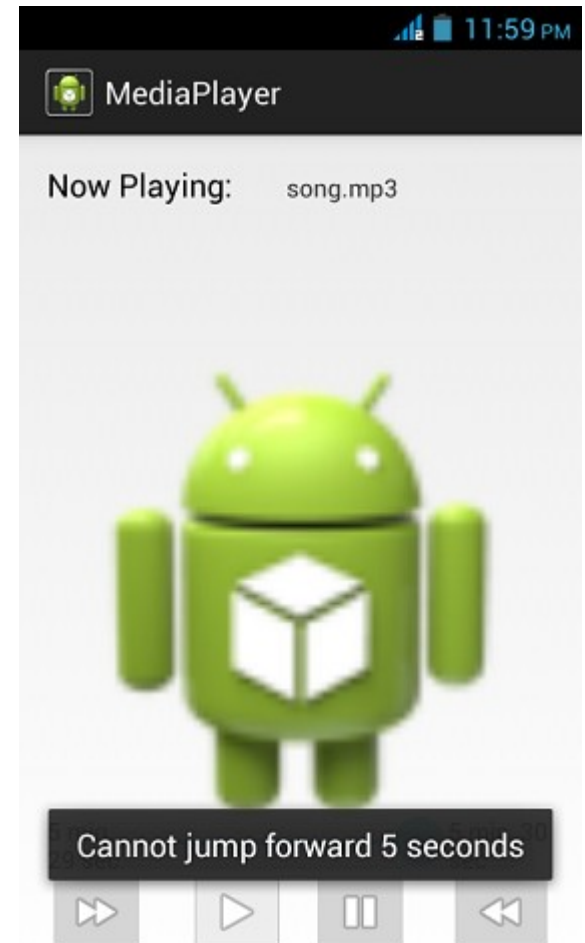
# Playing Audio

```
MediaPlayer mp = MediaPlayer.create(context, R.raw.sound);  
mp.start();  
... mp.stop();
```

- Nếu stop, phải gọi mp.reset() và mp\_prepare() trước khi gọi mp\_start() lần nữa
- Có thể tạm dừng mp.pause() rồi quay trở lại sử dụng mp.start()
- Đảm bảo gọi mp.release() khi kết thúc

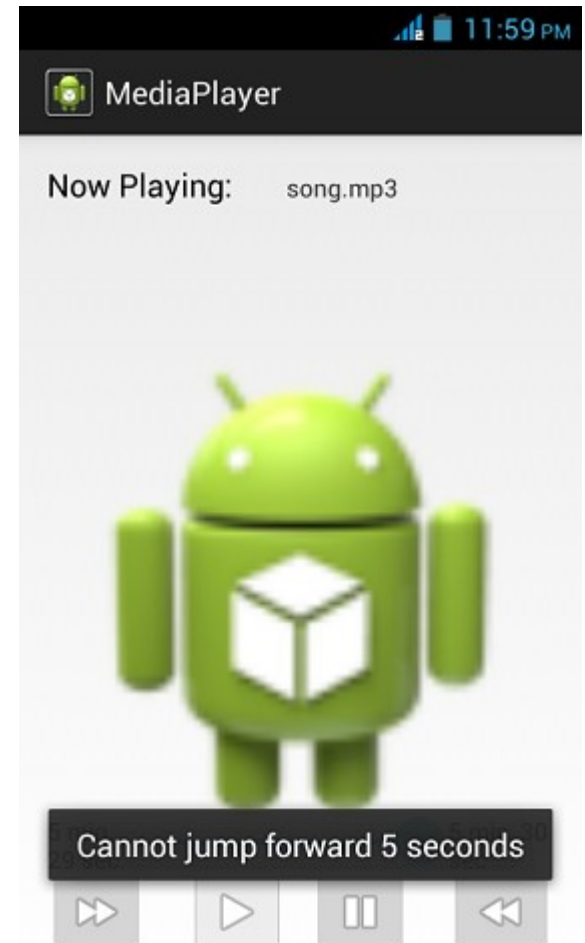
# Playing File hoặc Stream

- Có thể truyền đối tượng Uri trở tới local file hoặc internet stream
- Làm việc với HTTP/RTSP stream
- Làm việc với định dạng file nhất định



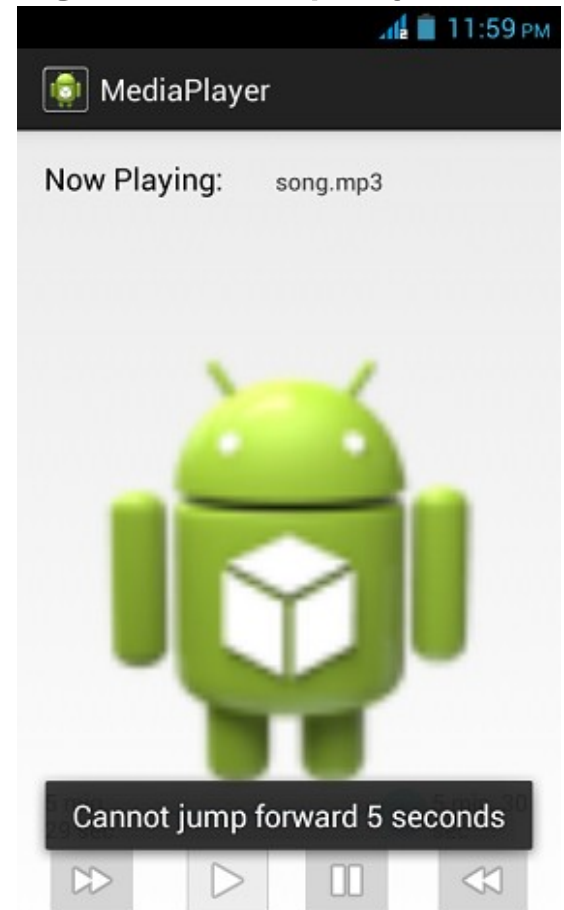
# Playing Video

- Tương tự như audio, ngoại trừ phải copy resource trước
- Bổ sung: truyền **SurfaceHolder** tới player có thể sinh video
- Giống như cửa sổ preview của Camera
- **getHolder()** của **SurfaceView**
- Copy file tới emulator card để test



# Media Event

- Media Player có thể ném ra một ngoại lệ: ví dụ format không được chấp nhận
- Media Player hỗ trợ một số listener để tương tác với player
  - OnErrorListener
  - OnBufferingUpdateListener





# Media Event

```
public class MyService extends Service implements MediaPlayer.OnErrorListener {  
    MediaPlayer mMediaPlayer;  
  
    public void initMediaPlayer() {  
        // ...initialize the MediaPlayer here...  
  
        mMediaPlayer.setOnErrorListener(this);  
    }  
  
    @Override  
    public boolean onError(MediaPlayer mp, int what, int extra) {  
        // ... react appropriately ...  
        // The MediaPlayer has moved to the Error state, must be reset!  
    }  
}
```

## Sử dụng wake lock

- Khi thiết kế ứng dụng dùng để chơi nhạc dưới dạng background, thiết bị có thể sleep khi service đang chạy bởi vì hệ thống Android cố gắng tiết kiệm pin khi thiết bị đang sleep
- Nếu service đang chơi một bản nhạc, bạn có thể muốn ngăn cản hệ thống can thiệp vào service nghe nhạc
- Để đảm bảo service vẫn tiếp tục chạy, bạn phải sử dụng wake lock. Wake lock là một cách để báo cho hệ thống biết là một số chức năng vẫn chạy kể cả khi phone ở trạng thái idle

## Sử dụng wake lock

- Để đảm bảo rằng CPU tiếp tục chạy khi MediaPlayer đang chơi nhạc, gọi phương thức **setWakeMode()** khi khởi tạo MediaPlayer
- Khi đó MediaPlayer sẽ bị khóa khi chơi nhạc và giải phóng khóa khi tạm dừng hoặc dừng chơi nhạc

```
mMediaPlayer = new MediaPlayer();  
// ... other initialization here ...  
mMediaPlayer.setWakeMode(getApplicationContext(), PowerManager.PARTIAL_WAKE_LOCK);
```

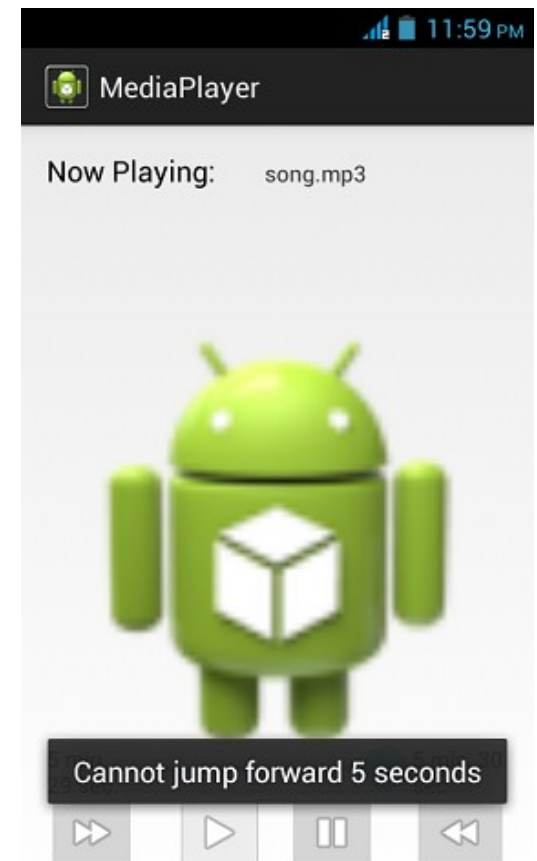
## Sử dụng wake lock

- Tuy nhiên, wake lock chỉ đảm bảo là CPU vẫn không sleep
- Nếu bạn muốn stream một bản nhạc qua Wifi, bạn có thể cần giữ cho Wifi không bị tắt. Do đó, bạn cần giữ WifiLock và bạn phải lấy thông tin WifiLock và giải phóng một cách thủ công
- Do đó, khi bạn bắt đầu chuẩn bị MediaPlayer với một URL, bạn nên tạo và lấy thông tin Wifi lock

```
WifiLock wifiLock = ((WifiManager) getSystemService(Context.WIFI_SERVICE))  
    .createWifiLock(WifiManager.WIFI_MODE_FULL, "mylock");  
  
wifiLock.acquire();
```

# Định dạng được hỗ trợ

- Core Format hỗ trợ trên mọi thiết bị
  - Thiết bị có thể chọn hỗ trợ thêm các định dạng khác
  - Một số không được bổ sung vào SDK < 2.2, do đó cần test trên các thiết bị và emulator cũ
- Định dạng chung:
  - .3gp , .mp4.m4a
  - .mp3, .ogg, .wav audio
  - jpg, gif, png, bmp images





# DEMO

Thao tác với sound trong ứng dụng  
Android



# Google cloud messaging (GCM)

- Là dịch vụ miễn phí để gửi message tới các thiết bị Android
- GCM có thể tăng trải nghiệm người dùng
- GCM cho phép gửi cho 1000 người trên một message
- Giảm thiểu gánh nặng cho server



# Google cloud messaging (GCM)

- Bạn có thể nhúng dữ liệu trong message
- GCM message có thể chứa 4kb dữ liệu





## Backup API

- Khi người dùng mua một thiết bị mới hoặc reset dữ liệu, họ có thể muốn Google Play khôi phục ứng dụng trên thiết bị cũ trong khi dùng thiết bị mới, dữ liệu cũ trong ứng dụng cũng muốn được khôi phục
- Mặc định, Google Play không hỗ trợ và toàn bộ thiết lập cũ sẽ bị mất



## Backup API

- Khi dữ liệu khá nhỏ (nhỏ hơn 1MB) như user preference, note, game high score, Backup API là giải pháp tốt nhất



# Đăng ký Android Backup Service

- Vào đường link  
<https://developer.android.com/google/backup/signup.html?csw=1> để đăng ký
- Nhập thông tin Application Package Name
- Sau đó, Google Play sẽ trả lại key và hướng dẫn cách đưa thông tin key vào Android Manifest

# Đăng ký Android Backup Service



## Android Backup Service Key

## Thank you for registering for an Android Backup Service Key!

Your key is:

AEdPqrEAAAAIFe0tDW5puvdRV4Q\_2Yh1NHbDTT7HFTVoFEs1Ig

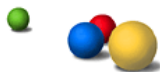
This key is good for the app with the package name:

fpoly.com

Provide this key in your AndroidManifest.xml file with the following<meta-data> element, placed inside the<application> element

```
<meta-data android:name="com.google.android.backup.api_key" android:value="AEdPqrEAAAAIFe0tDW5puvdRV4Q_2Yh1NHbDTT7HFTVoFEs1Ig" />
```

For more information, see the [Backup Dev Guide](#) or go back to the [Android Backup Service registration](#).



©2013 Google - [Privacy Policy](#)



DEMO

Đăng ký Android Backup Service



## **Viết code Backup agent**

- Cách lý tưởng nhất là kế thừa lớp BackupAgentHelper
- Trong phương thức onCreate(), tạo BackupHelper
- Các lớp này dùng để back up một số loại dữ liệu
- Android framework cung cấp hai helper:
  - FileBackupHelper
  - SharedPreferencesBackupHelper

# Viết code Backup agent

```
public class TheBackupAgent extends BackupAgentHelper {
    // The names of the SharedPreferences groups that the application maintains. These
    // are the same strings that are passed to getSharedPreferences(String, int).
    static final String PREFS_DISPLAY = "displayprefs";
    static final String PREFS_SCORES = "highscores";

    // An arbitrary string used within the BackupAgentHelper implementation to
    // identify the SharedPreferencesBackupHelper's data.
    static final String MY_PREFS_BACKUP_KEY = "myprefs";

    // Simply allocate a helper and install it
    public void onCreate() {
        SharedPreferencesBackupHelper helper =
            new SharedPreferencesBackupHelper(this, PREFS_DISPLAY, PREFS_SCORES);
        addHelper(MY_PREFS_BACKUP_KEY, helper);
    }
}
```

# Tổng kết nội dung bài học

- Camera
- Media
- Google cloud messaging

