# Services – Xử lý tiến trình chạy nền

### Service là gì?

* Là thành phần chạy ngầm, không hiển thị giao diện.
* Có thể chạy vô thời hạn, khi thành phần bắt đầu bị hủy (Activity)
* Không chạy Thread, phải tự tạo

### Khai báo trong Mainifresh:

<manifest ... >

...

<application ... >

<service android:name=".ExampleService" />

...

</application>

</manifest>

### Vòng đời:

|  |  |  |
| --- | --- | --- |
|  |  |  |

1. Component (Activity) bắt đầu Service: [startService()](http://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent))
   1. Service sẽ truy cập: [onStartCommand()](http://developer.android.com/reference/android/app/Service.html#onStartCommand(android.content.Intent,%20int,%20int))
   2. Dừng Service trong chính nó: [stopSelf()](http://developer.android.com/reference/android/app/Service.html#stopSelf()), [stopService()](http://developer.android.com/reference/android/content/Context.html#stopService(android.content.Intent))
2. Component (Activity) bắt đầu Service, Component muốn Bind (gắn kết) với Service: [bindService()](http://developer.android.com/reference/android/content/Context.html#bindService%28android.content.Intent,%20android.content.ServiceConnection,%20int%29)
   1. Service sẽ truy cập: [onBind()](http://developer.android.com/reference/android/app/Service.html#onBind%28android.content.Intent%29)
   2. Component giao tiếp Service = interface trả về [IBinder](http://developer.android.com/reference/android/os/IBinder.html)
3. Service template:
4. public class MusicService extends Service {
5. public static final String EXTRA\_RAW\_MUSIC\_ID = "com.imark.nghia.nghiaservice\_1\_start.extra.RAW\_MUSIC\_ID";
6. MediaPlayer player;
7. /\*\*
8. \* Khởi đông Service mặc định trong hàm "onStartCommand"
9. \*/
10. public static void startPlayDefaultMusic(Context context) {
11. Intent iMusicService = new Intent(context, MusicService.class);
12. context.startService(iMusicService);
13. }
14. /\*\*
15. \* khởi đông Service put bài hát vào
16. \*/
17. public static void startPlayMusic(Context context, int musicRawId) {
18. Intent iMusicService = new Intent(context, MusicService.class);
19. iMusicService.putExtra(MusicService.EXTRA\_RAW\_MUSIC\_ID, musicRawId);
20. context.startService(iMusicService);
21. }
22. /\*\*
23. \* Dừng Service -> hàm "onDestroy" trong Service
24. \*/
25. public static void stopPlayMusic(Context context) {
26. Intent iMusicService = new Intent(context, MusicService.class);
27. context.stopService(iMusicService);
28. }
29. /\*\*
30. \* Khởi tạo Service
31. \*/
32. @Override
33. public void onCreate() {
34. super.onCreate();
35. // khởi tạo Player
36. player = MediaPlayer.create(this, R.raw.defaut\_music);
37. player.setLooping(true);
38. }
39. /\*\*
40. \* The service is starting, hành động sau khi Component gọi "startService()"
41. \*/
42. @Override
43. public int onStartCommand(Intent intent, int flags, int startId) {
44. // Lấy Intent trong Service
45. int musicId = intent.getIntExtra(EXTRA\_RAW\_MUSIC\_ID, 0);
46. if (musicId != 0) {
47. // có nhạc
48. player = MediaPlayer.create(this, musicId);
49. }
50. // run player
51. player.start();
52. return START\_STICKY;
53. }
54. /\*\*
55. \* A client is binding to the service with
56. \* bindService()
57. \* Sử dụng để ứng dụng khác gọi Service
58. \*/
59. @Override
60. public IBinder onBind(Intent intent) {
61. return null;
62. }
63. @Override
64. public void onDestroy() {
65. // stop music
66. player.stop();
67. }

}

**Ghi chú code:**  
onStartCommand Return:

1. [START\_NOT\_STICKY](http://developer.android.com/reference/android/app/Service.html#START_NOT_STICKY):
   1. Tắt Service – Hệ thống (Android OS) không cần tạo lại. Mất kết quả.
   2. Sử dụng: Khởi động lại Unfinish Jobs cùng Aplication.
2. [START\_STICKY](http://developer.android.com/reference/android/app/Service.html#START_STICKY):
   1. Tắt Service – đủ tài nguyên – Hệ thống tự tạo lại = null Intent.
   2. Sử dụng khi: Phát Media, Không cần lệnh chạy, chạy vô thời hạn, chạy để sử lý.
3. [START\_REDELIVER\_INTENT](http://developer.android.com/reference/android/app/Service.html#START_REDELIVER_INTENT)
   1. Tắt Service – hệ thống tự tạo lại = Intent cuối cùng.
   2. Sử dụng khi: Service cần khởi động lặp tức, như tải 1 tập tin (tải lại sau khi chết =D).

### Khởi động Service:

// khởi động Service -> đến hàm "onStartCommand" trong Service

Intent iMusicService = new Intent(this, MusicService.class);

startService(iMusicService);

// Dừng Service -> đến hàm "onDestroy" trong Service

Intent iMusicService = new Intent(this, MusicService.class);

stopService(iMusicService);

### Dừng Service:

**Tự dừng:**

* [stopSelf()](http://developer.android.com/reference/android/app/Service.html#stopSelf%28%29),
* [stopSelf(int)](http://developer.android.com/reference/android/app/Service.html#stopSelf%28int%29) : Dừng với Id của Component yêu cầu

**Dừng trong Component khác:**

* [stopService()](http://developer.android.com/reference/android/content/Context.html#stopService%28android.content.Intent%29)

### Nhận kết quả Service trả về = Receiver:

#### Trực tiếp trong Service:

**Toast, Notification**.

#### Gián tiếp qua Receiver:

Service gởi Broadcast Receiver:

/\*\*

\* Send Broadcast Receiver trong Service

\*/

private void publishResults(int result) {

Intent intent = new Intent(ACTION\_DOWNLOAD);

intent.putExtra(RESULT, result);

sendBroadcast(intent);

}

Activity khai báo Broadcoast nhận kết quả từ Service:

/\*\*

\* Tạo BR nhận kết quả từ DownloadService

\*/

private BroadcastReceiver receiver = new BroadcastReceiver() {

@Override

public void onReceive(Context context, Intent intent) {

// nhận Result từ DownloadService

Bundle bundle = intent.getExtras();

if (bundle != null) {

int result = bundle.getInt(DownloadService.RESULT); // kết quả

}

}

};

@Override

protected void onResume() {

super.onResume();

// đăng kí receiver

registerReceiver(receiver, new IntentFilter(DownloadService.ACTION\_DOWNLOAD));

}

@Override

protected void onPause() {

super.onPause();

// hủy đăng kí receiver

unregisterReceiver(receiver);

}

**Tham khảo:**

* <http://developer.android.com/guide/components/aidl.html>
* <http://o7planning.org/web/fe/default/vi/document/1162405/huong-dan-lap-trinh-android-service>

# Intent Service – Service trả về 1 lần

### Intent Service:

**Tự dừng lại** sau khi trả về kết quả, hay hàm **“onHandleIntent”**xử lý xong

### Template:

/\*\*

\* Created by Nghia-PC on 6/24/2015.

\* Service tự kết thúc khi xử lý xong

\*/

public class DownloadService extends IntentService {

private int result = Activity.RESULT\_CANCELED;

public static final String ACTION\_DOWNLOAD = "com.vogella.android.service.receiver";

public static final String URL = "urlpath";

public static final String FILENAME = "filename";

public static final String FILEPATH = "filepath";

public static final String RESULT = "result";

public DownloadService(){

super("DownloadServices");

}

@Override

public IBinder onBind(Intent intent) {

return null;

}

@Override

protected void onHandleIntent(Intent intent) {

// lấy intent gởi đến

String urlPath = intent.getStringExtra(URL);

String fileName = intent.getStringExtra(FILENAME);

// xóa file nếu tồn tại

File file = new File(Environment.getExternalStorageDirectory(), fileName);

if (file.exists()){

file.delete();

}

// ghi từng dòng đến khi kết thúc

InputStream inputStream = null;

FileOutputStream outputStream = null;

try {

// open url connection

java.net.URL url = new URL(urlPath);

inputStream = url.openConnection().getInputStream();

// read request

InputStreamReader reader = new InputStreamReader(inputStream);

outputStream = new FileOutputStream(file.getPath());

// input read => output write

int next = -1;

while ((next = reader.read()) != -1){

outputStream.write(next);

}

inputStream.close();

outputStream.close();

// successfully finished

result = Activity.RESULT\_OK;

} catch (Exception ex){

ex.printStackTrace();

}

// gởi kết quả về, receiver nhận

publishResults(file.getAbsolutePath(), result);

}

/\*\*

\* Send Broadcast Receiver trong Service

\*/

private void publishResults(String absolutePath, int result) {

Intent intent = new Intent(ACTION\_DOWNLOAD);

intent.putExtra(FILEPATH, absolutePath);

intent.putExtra(RESULT, result);

sendBroadcast(intent);

}

}

### Activity gọi:

public class MainActivity extends Activity {

private BroadcastReceiver receiver = new BroadcastReceiver() {

@Override

public void onReceive(Context context, Intent intent) {

// nhận Result từ DownloadService

Bundle bundle = intent.getExtras();

if (bundle != null) {

String downloadPath = bundle.getString(DownloadService.FILEPATH);

int resultCode = bundle.getInt(DownloadService.RESULT);

if (resultCode == RESULT\_OK){

// download success

String message = "Download done: " + downloadPath;

Toast.makeText(getBaseContext(), message, Toast.LENGTH\_SHORT).show();

txtStatus.setText(message);

} else {

// download failed

Toast.makeText(getBaseContext(), "Download failed", Toast.LENGTH\_SHORT).show();

txtStatus.setText("Download failed");

}

}

}

};

@Override

protected void onCreate(Bundle savedInstanceState) {

...

}

@Override

protected void onResume() {

super.onResume();

// đăng kí receiver

registerReceiver(receiver, new IntentFilter(DownloadService.ACTION\_DOWNLOAD));

}

@Override

protected void onPause() {

super.onPause();

// hủy đăng kí receiver

unregisterReceiver(receiver);

}

/\*\*

\* Hàm chính: Nhấp Download

\*\*/

private void btnDownload\_onClick(View v) {

String url = txtURL.getText().toString();

// Thêm thông tin, đường dẫn cho DownloadService

Intent intent = new Intent(this, DownloadService.class);

intent.putExtra(DownloadService.FILENAME, "index.html");

intent.putExtra(DownloadService.URL, url);

startService(intent); // Chạy DownloadService, chạy 1 lần, chạy ngầm

txtStatus.setText("Service Download start");

}

}

**Tham khảo:**

* <http://developer.android.com/guide/components/aidl.html>
* <http://o7planning.org/web/fe/default/vi/document/1162405/huong-dan-lap-trinh-android-service>

# Bound Service – Dịch vụ gắn kết

## Bound Service

* Component ràng buộc Service để lấy kết quả liên tục
* Component chạy Service: [bindService()](http://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int))
* Service truy cập hàm: [onBind()](http://developer.android.com/reference/android/app/Service.html#onBind(android.content.Intent))

### Template:

/\*\*

\* Bind service dùng Ibinder đồng bộ

\* Client & service chạy cùng ứng dụng, cùng tiến trình

\* Example: run music on background

\*/

public class LocalService extends Service {

Random mRandom = new Random();

private IBinder mBinder = new LocalBinder();

public LocalService() {

}

@Override

public IBinder onBind(Intent intent) {

return mBinder;

}

/\*\*

\* Tính tổng 3 số, 1 số ngẫu nhiên (trong Service)

\* @param number1

\* @param number2

\* @return

\*/

public int getTotal(int number1, int number2) {

return number1 + number2 + mRandom.nextInt();

}

/\*\*

\* Lớp dùng cho client Binder.

\* Service chạy cùng tiến trình với clients => không cần dùng IPC

\* perform interprocess communication (IPC): Thực hiện giao tiếp trong tiến trình

\*/

public class LocalBinder extends Binder {

public LocalService getService() {

// trả về khởi tạo của LocalService, để Content nhận lưu lại

return LocalService.this;

}

}

}

### Gọi trong Component khác (ví dụ: Activity):

public class MainActivity extends Activity {

EditText mEditValue1, mEditValue2;

Button mButtonAdd;

boolean mIsServiceBound;

LocalService mLocalService;

@Override

protected void onCreate(Bundle savedInstanceState) {

...

}

/\*\*

\* Định nghĩa phản hồi cho Service binding,

\* chuyển đến hàm bindService() của Service

\*/

private ServiceConnection mLocalService\_Connection = new ServiceConnection() {

@Override

public void onServiceConnected(ComponentName name, IBinder service) {

// bound(ràng buộc) to LocalService

// chuyển IBinder => lấy khởi tạo LocalService

LocalService.LocalBinder binder = (LocalService.LocalBinder) service;

mLocalService = binder.getService();

mIsServiceBound = true;

Toast.makeText(getBaseContext(), "Local Service connected", Toast.LENGTH\_SHORT).show();

}

@Override

public void onServiceDisconnected(ComponentName name) {

mIsServiceBound = false;

Toast.makeText(getBaseContext(), "Local Service disconnected", Toast.LENGTH\_SHORT).show();

}

};

@Override

protected void onStart() {

super.onStart();

// bound LocalService

Intent intent = new Intent(this, LocalService.class);

bindService(intent, mLocalService\_Connection, Activity.BIND\_AUTO\_CREATE);

mIsServiceBound = true;

}

@Override

protected void onStop() {

super.onStop();

// unBound LocalService

unbindService(mLocalService\_Connection);

}

/\*\*

\* Hàm chính: Tính total dự trên service

\*/

private void showTotalNumber() {

if (mIsServiceBound) {

// gọi phương thức trong localService

// có thể bị treo máy. đề nghị chạy trong thread để không giảm hiệu xuất

int number1 = Integer.parseInt(mEditValue1.getText().toString());

int number2 = Integer.parseInt(mEditValue2.getText().toString());

int total = mLocalService.getTotal(number1, number2);

Toast.makeText(this, "Tổng là: " + total, Toast.LENGTH\_LONG).show();

}

}

}

## Sử dụng Messager

* Giao tiếp tiến trình điều kiển (Tương tác ngược lại Activity)
* Không cần sử dụng AIDL (Android Interface Definition Language)

### Cách sử dụng:

* Dịch vụ triển khai [Handler](http://developer.android.com/reference/android/os/Handler.html) để nhận lệnh gọi lại cho mỗi lệnh gọi từ một máy khách.
* [Handler](http://developer.android.com/reference/android/os/Handler.html) được sử dụng để tạo một đối tượng [Messenger](http://developer.android.com/reference/android/os/Messenger.html) (là một tham chiếu tới [Handler](http://developer.android.com/reference/android/os/Handler.html)).
* [Messenger](http://developer.android.com/reference/android/os/Messenger.html) tạo một [IBinder](http://developer.android.com/reference/android/os/IBinder.html) mà dịch vụ trả về máy khách từ [onBind()](http://developer.android.com/reference/android/app/Service.html#onBind%28android.content.Intent%29).
* Máy khách sử dụng [IBinder](http://developer.android.com/reference/android/os/IBinder.html) để khởi tạo [Messenger](http://developer.android.com/reference/android/os/Messenger.html) (tham chiếu tới [Handler](http://developer.android.com/reference/android/os/Handler.html) của dịch vụ), mà máy khách sử dụng để gửi các đối tượng [Message](http://developer.android.com/reference/android/os/Message.html) tới dịch vụ.
* Dịch vụ nhận được từng [Message](http://developer.android.com/reference/android/os/Message.html) trong [Handler](http://developer.android.com/reference/android/os/Handler.html) của mình—cụ thể là theo phương pháp [handleMessage()](http://developer.android.com/reference/android/os/Handler.html#handleMessage%28android.os.Message%29).

### Template:

/\*\*

\* Created by Nghia-PC on 6/26/2015.

\* Service chuyển chữ thành Hoa, trả về cho Activity

\*/

public class MessageService extends Service {

// lệnh đến service để hiển thị tin nhắn

public static final int MSG\_SAY\_HELLO = 1;

public static final int MSG\_TO\_UPPER\_CASE = 2;

public static final int MSG\_TO\_UPPER\_CASE\_RESPONSE = 3;

public static final String KEY\_UPPER\_DATA = "UPPER DATA";

// client gởi messages đến IncomingHandler

final Messenger mMessenger = new Messenger(new IncomingHandler());

/\*\*

\* Khi ràng buộc service,

\* trả về Messager để gởi Message đến Service

\* @param intent

\* @return

\*/

@Override

public IBinder onBind(Intent intent) {

Toast.makeText(getApplicationContext(), "binding", Toast.LENGTH\_SHORT).show();

return mMessenger.getBinder();

}

/\*\*

\* Handler tin nhắn tới từ Client

\*/

public class IncomingHandler extends Handler {

@Override

public void handleMessage(Message msg) {

try {

switch (msg.what) {

case MSG\_SAY\_HELLO:

// ---- Hiện hello ----

Toast.makeText(getApplicationContext(), "hello!", Toast.LENGTH\_SHORT).show();

break;

case MSG\_TO\_UPPER\_CASE:

// ---- Chuyển chữ hoa ----

// Lấy data gởi đến

String data = msg.getData().getString("data");

// init response message

Message message = Message.obtain(null, MSG\_TO\_UPPER\_CASE\_RESPONSE);

Bundle bundle = new Bundle();

bundle.putString(KEY\_UPPER\_DATA, data.toUpperCase());

message.setData(bundle);

// gởi phản hồi về activity

msg.replyTo.send(message);

default:

super.handleMessage(msg);

break;

}

} catch (RemoteException e) {

e.printStackTrace();

}

}

}

}

### Activity sử dụng:

public class MainActivity extends Activity {

/\*\* Messenger giao tiếp Service \*/

Messenger mMessenger = null;

/\*\* Flag đánh dấu service chạy \*/

boolean isServiceBound;

/\*\*

\* Lớp tương tác giao diện chính của Service

\*/

private ServiceConnection mConnection = new ServiceConnection() {

public void onServiceConnected(ComponentName className, IBinder service) {

// Hàm sẽ gọi khi kết nối Service khởi tạo, Trả về đối tượng tương tác với Service.

// Giao tiếp = Messenger => Chuyển IBinder thành Messenger

mMessenger = new Messenger(service);

isServiceBound = true;

}

public void onServiceDisconnected(ComponentName className) {

// gọi khi bị mất kết nối, đường truyền không mong đợi

// :)) đùa à

mMessenger = null;

isServiceBound = false;

}

};

@Override

protected void onCreate(Bundle savedInstanceState) {

...

}

@Override

protected void onStart() {

super.onStart();

// kết nối Service

bindService(new Intent(this, MessageService.class), mConnection, Context.BIND\_AUTO\_CREATE);

}

@Override

protected void onStop() {

super.onStop();

// Hủy kết nối Service

if (isServiceBound) {

unbindService(mConnection);

isServiceBound = false;

}

}

public void sayHello(View v) {

if (!isServiceBound) return;

try {

// Tạo và gởi Message to Service, sử dụng "what"

Message msg = Message.obtain(null, MessageService.MSG\_SAY\_HELLO, 0, 0);

mMessenger.send(msg);

} catch (RemoteException e) {

e.printStackTrace();

}

}

/\*\*

\* lấy chữ trong Editext, gởi service, nhận phản hồi

\*/

private void Button2\_onClick(View v) {

if (isServiceBound) {

try {

String text = mEditText.getText().toString();

// Khởi tạo tin nhắn: msg.what = ConvertService.TO\_UPPER\_CASE

Message message = Message.obtain(null, MessageService.MSG\_TO\_UPPER\_CASE);

// đưa dữ liệu vào messege

Bundle b = new Bundle();

b.putString("data", text);

message.setData(b);

// Nhận phản hồi từ Service bằng lớp IncomingHandler()

message.replyTo = new Messenger(new ResponseHandler());

// gởi message đến Service

mMessenger.send(message);

} catch (Exception ex){

ex.printStackTrace();

}

} else {

Toast.makeText(MainActivity.this, "Service is disconneted!", Toast.LENGTH\_LONG).show();

}

}

// Lớp nhận phải hồi từ Service

class ResponseHandler extends Handler {

@Override

public void handleMessage(Message msg) {

int respCode = msg.what;

switch (respCode) {

case MessageService.MSG\_TO\_UPPER\_CASE\_RESPONSE: {

// Nhận dữ liệu từ Service

String value = msg.getData().getString(MessageService.KEY\_UPPER\_DATA);

Toast.makeText(getBaseContext(), value, Toast.LENGTH\_LONG).show();

}

}

}

}

}

## Android Interface Definition Language (AIDL) – Chưa thực hiện

**Tham khảo:**

* <http://developer.android.com/intl/vi/guide/components/bound-services.html>