

# DỊCH VỤ CHẠY NỀN (SERVICE)

## 1. Khái niệm Service

Dịch vụ (Service) trong Android là thành phần giúp thực hiện các tác vụ dài hạn hoặc chạy ngầm mà không yêu cầu giao diện người dùng.

Dịch vụ tiếp tục chạy ngay cả khi người dùng thoát khỏi ứng dụng hoặc màn hình đã tắt, giúp đảm bảo trải nghiệm mượt mà.

Service không cung cấp giao diện cho người dùng.

Ví dụ:

Phát nhạc: Ứng dụng nghe nhạc sử dụng Service để phát nhạc ngay cả khi người dùng đã thoát khỏi giao diện.

Đồng bộ dữ liệu: Ứng dụng có thể đồng bộ hóa dữ liệu với máy chủ hoặc tải dữ liệu từ mạng mà không cần người dùng phải luôn mở ứng dụng.

Cập nhật vị trí: Các ứng dụng định vị có thể dùng Service để theo dõi vị trí của người dùng ngay cả khi không hiển thị giao diện ứng dụng.

## 2. Phân loại Service

Trong Android, có hai loại chính của Service, mỗi loại phục vụ các mục đích khác nhau:

- Started Service:
  - Bắt đầu bằng cách gọi startService().
  - Chạy trong nền và tự động dừng lại khi hoàn tất nhiệm vụ.
  - Ví dụ: Tải ảnh từ internet, lưu trữ dữ liệu vào cơ sở dữ liệu.
- Bound Service:
  - Kết nối với các thành phần khác bằng cách gọi bindService().
  - Cho phép các ứng dụng khác kết nối và tương tác, cung cấp dịch vụ lâu dài.
  - Ví dụ: Cung cấp dữ liệu định vị liên tục cho nhiều thành phần trong một ứng dụng GPS.

## 3. Các phương thức chính trong Service

Service có một số phương thức quan trọng mà lập trình viên cần hiểu để kiểm soát hoạt động của nó:

- onStartCommand(Intent intent, int flags, int startId):
  - Phương thức này được gọi khi dịch vụ bắt đầu bởi startService().
  - Là nơi xử lý tác vụ cần chạy nền.
  - Trả về giá trị để cho hệ thống biết cách xử lý khi dịch vụ bị hủy giữa chừng.
- onBind(Intent intent):
  - Được gọi khi có thành phần muốn kết nối (bind) với Bound Service.
  - Trả về một đối tượng IBinder, cho phép ứng dụng tương tác với dịch vụ.
  - Trả về null với Started Service.

- `onDestroy()`:
  - Được gọi khi dịch vụ kết thúc.
  - Lý tưởng để dọn dẹp các tài nguyên hoặc giải phóng bộ nhớ.

#### 4. Cách sử dụng Service

- Bắt đầu một Started Service

- Khai báo Service trong AndroidManifest.xml:

```
<service android:name=".MyService"/>
```

- Khởi chạy Service bằng cách gọi `startService()`:

```
Intent intent = new Intent(this, MyService.class);
startService(intent);
```

- Dừng Service khi tác vụ đã hoàn thành:

```
stopService(intent);
```

- Sử dụng Bound Service

- Kết nối đến Bound Service bằng cách gọi `bindService()`.
- Truyền dữ liệu và nhận phản hồi từ Service qua IBinder được trả về từ `onBind()`.

# Thực hành

## Yêu cầu:

Xây dựng ứng dụng nghe nhạc (chạy nền kể cả khi tắt ứng dụng).

## Hướng dẫn:

### **1. Tạo dự án.**

### **2. Tạo thư mục chứa nhạc trong dự án.**

Tạo thư mục res/raw chứa nhạc.

Kéo thả file nhạc vào thư mục vừa tạo.

Lưu ý: Đặt tên tệp không có ký tự đặc biệt, không có khoảng trắng và phải là chữ thường, ví dụ: breakfast.mp3

### **3. Khai báo MusicService trong AndroidManifest.xml**

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        <!-- Các khai báo đang có-->
        <!-- Khai báo dịch vụ-->
        <service android:name=".MusicService" />
    </application>
</manifest>
```

### **4. Tạo lớp Service: MusicService.**

```
public class MusicService extends Service {
    private MediaPlayer mediaPlayer;
    @Override
    public void onCreate(){
        super.onCreate();
        //Khởi tạo MediaPlayer với tệp nhạc trong res/raw
        mediaPlayer= MediaPlayer.create(this,R.raw.breakfast);
        mediaPlayer.setLooping(true); //Đặt phát nhạc lặp lại
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startID){
        mediaPlayer.start(); //Bắt đầu phát nhạc
        Toast.makeText(this, "Phát nhạc nền", Toast.LENGTH_SHORT).show();
        return START_STICKY;
    }
    @Override
    public void onDestroy() {
        super.onDestroy();
```

```

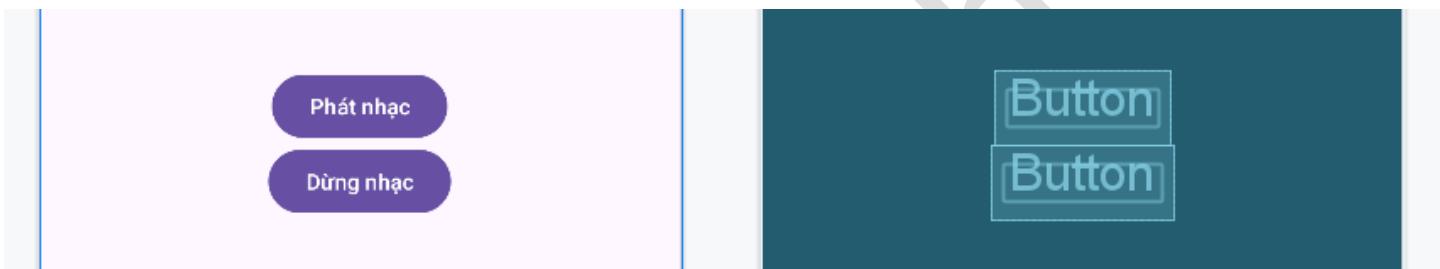
        if (mediaPlayer != null) {
            mediaPlayer.stop();
            mediaPlayer.release(); // Giải phóng tài nguyên khi dừng Service
        }
        Toast.makeText(this, "Dừng nhạc nền", Toast.LENGTH_SHORT).show();
    }

@Override
public IBinder onBind(Intent intent) {
    return null; // Không cần bind vì đây là Started Service
}
}

```

## 5. Xây dựng giao diện ứng dụng

Trong activity\_main.xml tạo giao diện đơn giản gồm 2 nút Phát nhạc và Dừng nhạc:



## 6. Khởi động và Dừng MusicService từ MainActivity

```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Tìm nút buttonStart trong giao diện và gán một sự kiện click cho nó
        findViewById(R.id.buttonStart).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Tạo Intent để khởi động MusicService, đây là Service phát
nhạc chạy nền
                Intent intent = new Intent(MainActivity.this,
MusicService.class);
                startService(intent); // Bắt đầu chạy Service bằng phương thức
startService()
            }
        });
    }
}

```

```
// Tìm nút buttonStop trong giao diện và gán một sự kiện click cho nó
findViewById(R.id.buttonStop).setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Tạo Intent để dừng MusicService
        Intent intent = new Intent(MainActivity.this,
MusicService.class);
        stopService(intent); // Dừng Service bằng phương thức
stopService()
    }
});
```

## 7. Chạy thử ứng dụng.

Ứng dụng khi thực thi sẽ lấy file `breakfast.mp3` trong thư mục `res/raw` để phát khi click nút Phát nhạc

Nhạc chỉ dừng lại khi lại khi click nút Dừng nhạc.

## 8. Phát triển ứng dụng.

Phát triển ứng dụng để ứng dụng có thể truy cập vào bộ nhớ của điện thoại để lấy nhạc và phát.

### Gợi ý:

- Xin quyền truy cập bộ nhớ.

- Trong `MainActivity`, thêm chức năng chọn tệp nhạc từ bộ nhớ thiết bị bằng cách sử dụng `Intent`. Khi người dùng chọn tệp, bạn sẽ nhận được đường dẫn tệp đó và truyền cho `MusicService`.

- Trong `MusicService`, lấy `URI` của tệp nhạc từ `Intent` và khởi tạo `MediaPlayer` để phát nhạc từ `URI` này.

Trường Mạnh Đạt