# Services Android

Ce document explique l'utilisation des services natifs Android à travers un exemple d'application. Les services Android permettent l'exécution de tâches en arrière-plan, indépendamment de l'interface utilisateur. Ils sont utiles pour des tâches prolongées, telles que la lecture de musique, le téléchargement de fichiers, ou la gestion d'opérations réseau.

## 1. Présentation de l'Application

L'application inclut un service nommé 'MyService', qui est lancé et arrêté par des boutons dans une activité principale. Elle illustre comment démarrer, arrêter, et gérer un service Android de base.

## 2. Composants de l'Application

### MainActivity

La classe MainActivity contient deux boutons permettant de démarrer et d'arrêter le service. Voici les principales fonctionnalités :

- Démarrage du service : Un Intent explicite est utilisé pour appeler la méthode startService().  
- Arrêt du service : Un Intent similaire appelle la méthode stopService().

Extrait de code pour démarrer le service :  
Intent intent = new Intent(getApplicationContext(), MyService.class);  
startService(intent);

### MyService

La classe MyService hérite de Service et implémente les méthodes clés pour la gestion du cycle de vie du service :

- onCreate() : Initialisé lorsque le service est créé.  
- onStartCommand() : Exécuté à chaque fois que le service est démarré via startService().  
- onDestroy() : Appelé lorsque le service est arrêté.

Exemple d'utilisation dans onStartCommand() :  
Log.i("myLog", "onStartCommand in service");

## 3. Configuration dans le fichier Manifest

Le fichier AndroidManifest.xml déclare le service et garantit qu'il peut être utilisé par l'application. Voici un extrait du fichier :

<service  
 android:name=".MyService"  
 android:enabled="true"  
 android:exported="true" />

## 4. Cycle de Vie du Service

Le cycle de vie d'un service inclut les étapes suivantes :

- \*\*Création\*\* : onCreate() est appelé une seule fois, lors de la création du service.  
- \*\*Exécution\*\* : onStartCommand() est appelé chaque fois que startService() est invoqué.  
- \*\*Destruction\*\* : onDestroy() est appelé lorsque le service est arrêté.

Main activity Code :

package com.example.appservices;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
  
 Button start , stop ;  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.*enable*(this);  
 setContentView(R.layout.*activity\_main*);  
 ViewCompat.*setOnApplyWindowInsetsListener*(findViewById(R.id.*layout*), (v, insets) -> {  
 Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.*systemBars*());  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
 return insets;  
 });  
  
 start = findViewById(R.id.*btnStartService*);  
 stop = findViewById(R.id.*btnStopService*);  
  
 Log.*i*("myLog","thread id in main acitvity is " + Thread.*currentThread*().getId());  
  
  
 *// handling the button* start.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 *//start service* Intent intent = new Intent(getApplicationContext(),MyService.class);  
 startService(intent);  
 }  
 });  
  
 stop.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent(getApplicationContext(), MyService.class);  
 stopService(intent);  
  
 }  
 });  
  
 }  
}

MyService code :

package com.example.appservices;  
  
import android.app.Service;  
import android.content.Intent;  
import android.os.IBinder;  
import android.util.Log;  
  
public class MyService extends Service {  
 public MyService() {  
 }  
  
 @Override  
 public IBinder onBind(Intent intent) {  
 *// TODO: Return the communication channel to the service.* throw new UnsupportedOperationException("Not yet implemented");  
 }  
  
 @Override  
 public void onCreate() {  
 Log.*i*("myLog", "onCreate in service");  
 super.onCreate();  
 }  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 Log.*i*("myLog", "onStartCommand in service");  
 Log.*i*("myLog","thread id in Service is " + Thread.*currentThread*().getId());  
 return super.onStartCommand(intent, flags, startId);  
 }  
  
 @Override  
 public void onDestroy() {  
 Log.*i*("myLog", "onDestroy in service");  
  
 super.onDestroy();  
 }  
}