

Compte rendu :

TP6 Développement Mobile : Développement d'un Service Android

Enseignant : Maddouri Faouzi

Réalisé par :

Guesmi Ikram

Farhaoui Eya

Année universitaire : 2024/2025

Etablissement: ISET Rades

Contents

1	Création de l'Interface Utilisateur	2
1.1	Code: Interface Utilisateur en XML	2
1.2	Interface utilisateur:	2
2	Création du Service :	2
2.1	Méthodes Implémentées	3
2.2	Code de la classe Myservice	3
3	Intégration de la Logique dans MainActivity	4
3.1	Gestion des Événements OnClickListener	4
3.2	Code de la class MainActivity	5

Objectif

Créer et intégrer un service simple dans une application Android pour démontrer son cycle de vie (création, démarrage, destruction) et interagir avec une interface utilisateur pour démarrer et arrêter le service.

1 Création de l'Interface Utilisateur

Dans cette interface, nous avons créé deux boutons :

- **Bouton "Start service"** : Permet de lancer le service à l'aide de la méthode `startService()`.
- **Bouton "Stop service"** : Permet d'arrêter le service à l'aide de la méthode `stopService()`.

1.1 Code: Interface Utilisateur en XML

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10    <Button
11        android:id="@+id/start"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:text="Start service"
15        tools:layout_editor_absoluteX="176dp"
16        tools:layout_editor_absoluteY="260dp" />
17
18    <Button
19        android:id="@+id/stop"
20        android:layout_width="wrap_content"
21        android:layout_height="wrap_content"
22        android:text="stop service"
23        tools:layout_editor_absoluteX="192dp"
24        tools:layout_editor_absoluteY="382dp" />
25 </androidx.constraintlayout.widget.ConstraintLayout>
```

1.2 Interface utilisateur:

2 Création du Service :

Nous avons créé un nouveau service nommé `MyService` dans le package `com.example.service`, qui hérite de la classe `Service`. Ce service inclut plusieurs méthodes importantes.

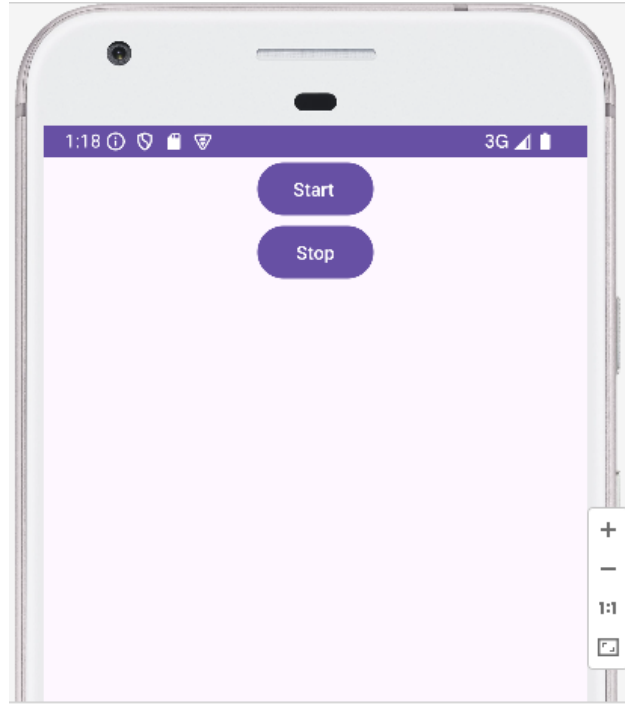


Figure 1: Interface obtenue

2.1 Méthodes Implémentées

- **onCreate()** : Nous avons ajouté une journalisation dans cette méthode afin qu'un message soit enregistré chaque fois que le service est créé.
- **onStartCommand()** :
 - Cette méthode gère les actions lorsque le service est démarré.
 - Nous avons configuré le code pour récupérer l'ID du thread courant, ce qui permet d'observer sur quel thread le service s'exécute.
 - Une fois le traitement terminé, nous avons implémenté un arrêt automatique du service en utilisant la méthode **stopSelf()**.
- **onDestroy()** : Un message est journalisé pour indiquer que le service a été détruit, ce qui nous permet de suivre son cycle de vie jusqu'à sa fin.

Des appels à `Log.i()` ont été ajoutés dans les méthodes principales du service pour suivre le comportement de l'application

2.2 Code de la classe MyService

```

1 package com.example.service;
2
3 import android.app.Service;
4 import android.content.Intent;
5 import android.os.IBinder;
6 import android.util.Log;
7
8 public class MyService extends Service {
9     public MyService() {
10    }
11
12    @Override
13    public IBinder onBind(Intent intent) {
14        // TODO: Return the communication channel to the service.
15        throw new UnsupportedOperationException("Not yet implemented");
16    }
17
18    @Override
19    public int onStartCommand(Intent intent, int flags, int startId) {
20        Log.i("mylog", "onStartCommand in service");
21        Log.i("mylog", "thread id in service is " + Thread.currentThread().getId());
22        stopSelf();
23        return super.onStartCommand(intent, flags, startId);
24    }
25
26    @Override
27    public void onCreate() {
28        Log.i("mylog", "onCreate in service");
29        super.onCreate();
30    }
31
32    @Override
33    public void onDestroy() {
34        Log.i("mylog", "onDestroy in service");
35        super.onDestroy();
36    }
37 }

```

3 Intégration de la Logique dans MainActivity

Dans notre activité principale (`MainActivity`), nous avons initialisé deux boutons pour permettre l'interaction avec le service `MyService`. Ces boutons ont été configurés pour gérer le démarrage et l'arrêt du service grâce à des gestionnaires d'événements (*OnClickListener*).

3.1 Gestion des Événements `OnClickListener`

- Pour le bouton "Start service" :
 - Nous avons créé un objet `Intent` pour démarrer le service `MyService`.

- La méthode `startService()` a été appelée avec l'intention correspondante pour activer le service.

- **Pour le bouton "Stop service" :**

- Un objet `Intent` a été créé pour identifier le service `MyService`.
- Nous avons utilisé la méthode `stopService()` pour arrêter le service.

- **Journalisation :**

- Nous avons journalisé l'ID du thread principal dans `MainActivity` afin de pouvoir le comparer avec celui utilisé par le service, ce qui permet une meilleure compréhension du fonctionnement multithreadé.

3.2 Code de la class MainActivity

```
1 package com.example.service;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.util.Log;
8 import android.view.View;
9 import android.widget.Button;
10
11 public class MainActivity extends AppCompatActivity {
12
13     Button start, stop;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19         start = (Button) findViewById(R.id.start);
20         stop = (Button) findViewById(R.id.stop);
21         Log.i("mylog", "thread_id_in_main_activity_is_" + Thread.currentThread().
            getId());
22
23         start.setOnClickListener(new View.OnClickListener() {
24             @Override
25             public void onClick(View view) {
26                 // Start service
27                 Intent intent = new Intent(getApplicationContext(), MyService.
                    class);
28                 startService(intent);
29             }
30         });
31
32         stop.setOnClickListener(new View.OnClickListener() {
33             @Override
34             public void onClick(View view) {
35                 // Stop service
```

```
36         Intent intent = new Intent(getApplicationContext(), MyService.  
37             class);  
38         stopService(intent);  
39     }  
40 }  
41 }
```