

# Atelier Développement Mobile

Lab 06

Réaliser par : Dhahbi Mohamed Amine

Mechergui Med Aziz

Encadré par: Maddouri Faouzi

Classe: L3DSI3

## Problématique:

Comment créer un service sur un Android?

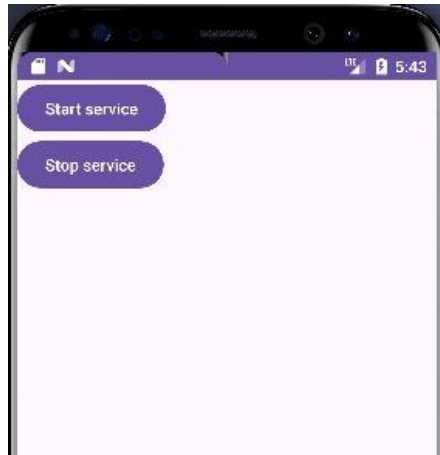
## Solution:

1- Concevoir une interface utilisateur intégrant deux boutons distincts pour le démarrage et l'arrêt d'un service.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <Button
        android:id="@+id/start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start service"
        tools:layout_editor_absoluteX="168dp"
        tools:layout_editor_absoluteY="251dp" />

    <Button
        android:id="@+id/stop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Stop service"
        tools:layout_editor_absoluteX="151dp"
        tools:layout_editor_absoluteY="539dp" />
</LinearLayout>
```

Figure 1: Code Xml de l'activitymain.xml



*Figure 2: Interface utilisateur*

2- Création du service :

```
public class MyService extends Service {
    public MyService() { no usages
    }

    @Override no usages
    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( tag: "my log", msg: "onCreate in service");
        super.onCreate();
    }

    @Override 5 usages
    public int onStartCommand(Intent intent, int flags, int startId) {
        Log.i( tag: "my log", msg: "onStartCommand in service: ");
        Log.i( tag: "mylog", msg: "thread id in main activity is "+Thread.currentThread().getId());
        return super.onStartCommand(intent, flags, startId);
    }

    @Override
    public void onDestroy() {
        Log.i( tag: "mylog", msg: "onDestroy in service: ");
        super.onDestroy();
    }
}
```

*Figure 3: La class Service*

**-onCreate()** : Appelée à la création du service, elle initialise le service et consigne un message dans le log.

**-onBind()** : Non utilisée ici, elle lève une exception indiquant que le service n'est pas lié.

**-onStartCommand()** : Exécutée chaque fois que le service est démarré avec startService().

**-onDestroy()** : Appelée à la destruction du service, elle nettoie les ressources et consigne un message.

3- Intégration dans le Main Activity : Configuration des deux boutons.

```
public class MainActivity extends AppCompatActivity {
    Button start, stop; 2 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        start=(Button)findViewById(R.id.start);
        stop=(Button) findViewById(R.id.stop);
        Log.i( tag: "mylog", msg: "thread id in main activity is "+Thread.currentThread().getId());
        start.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                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);
            }
        });
    }
}
```

Figure 4: La class Main\_Activity

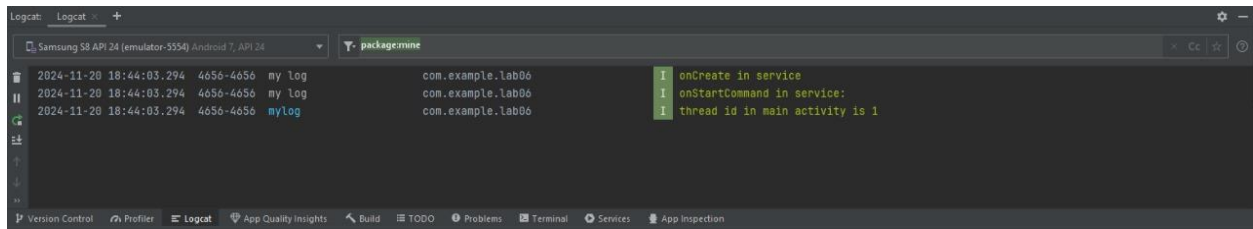


Figure 5: Lors de lancement de service

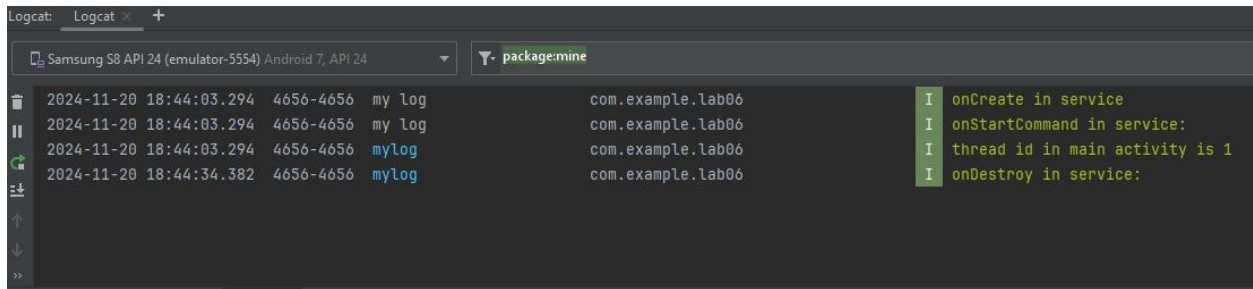


Figure 6: Lors de l'arrêt du service.