

Service

Himadri Parikh

Contents

- Introduction
- Requirements
- Design the application
- MainActivity.java
- Java Class
 - Register service in Manifest

Introduction

- A Service is an application component that can perform long-running operations in the background, and it doesn't provide a user interface.
- Another application component can start a service, and it continues to run in the background even if the user switches to another application.
- Additionally, a component can bind to a service to interact with it and even perform interprocess communication (IPC).
- For example, a service can handle network transactions, play music, perform file I/O, or interact with a content provider, all from the background.

Requirements

- We need to demonstrate playing music in background therefore we require:
 - Two buttons for start and stop
 - Java class which would work as service
 - Registering service in Manifest
 - Coding in MainActivity.java

Designing the app

- Take two buttons that would work as start and stop buttons.
- Create variables in MainActivity.java: //in onCreate()
 - Button btnStart, btnStop;
- Bind views with the variables:
 - btnStart = findViewById(R.id.btnStart);btnStop = findViewById(R.id.btnStop);

Coding in MainActivity.java

Use interface 'View.OnClickListener' and implement its method:

```
• @Override
public void onClick(View v) {
   if(v == btnStart){
      startService(new Intent(this, MyService.class));
   }
   else if(v == btnStop){
      stopService(new Intent(this, MyService.class));
   }
}
```

Service class

- Create a simple java class and extend it to service class as shown below:
 - public class MyService extends Service { }
- Implement its method:

```
@Override public IBinder onBind(Intent intent) { return null; }
```

- Make a local variable:
 - MediaPlayer mediaPlayer;

Override and implement onStartCommand() method:

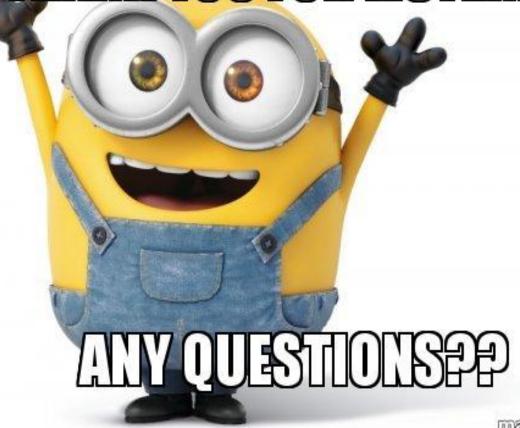
```
    @Override
        public int onStartCommand(Intent intent, int flags, int startId) {
            mediaPlayer = MediaPlayer.create(this, Settings.System.DEFAULT_RINGTONE_URI);
            mediaPlayer.setLooping(true);
            mediaPlayer.start();
            return START_STICKY;
        }
```

• Stop the music with onDestroy() method:

```
    @Override
public void onDestroy() {
    super.onDestroy();
    mediaPlayer.stop();
}
```

- Register service in Manifest file inside <application> tag:
 - <service android:name=".MyService"/>
- Here we are done with service demo.





Himadri Vrajesh Parikh

himadrichaudhary1314@gmail.com

https://www.linkedin.com/in/himadri-parikh-5062

+91-7990403184

makeameme.org