Services

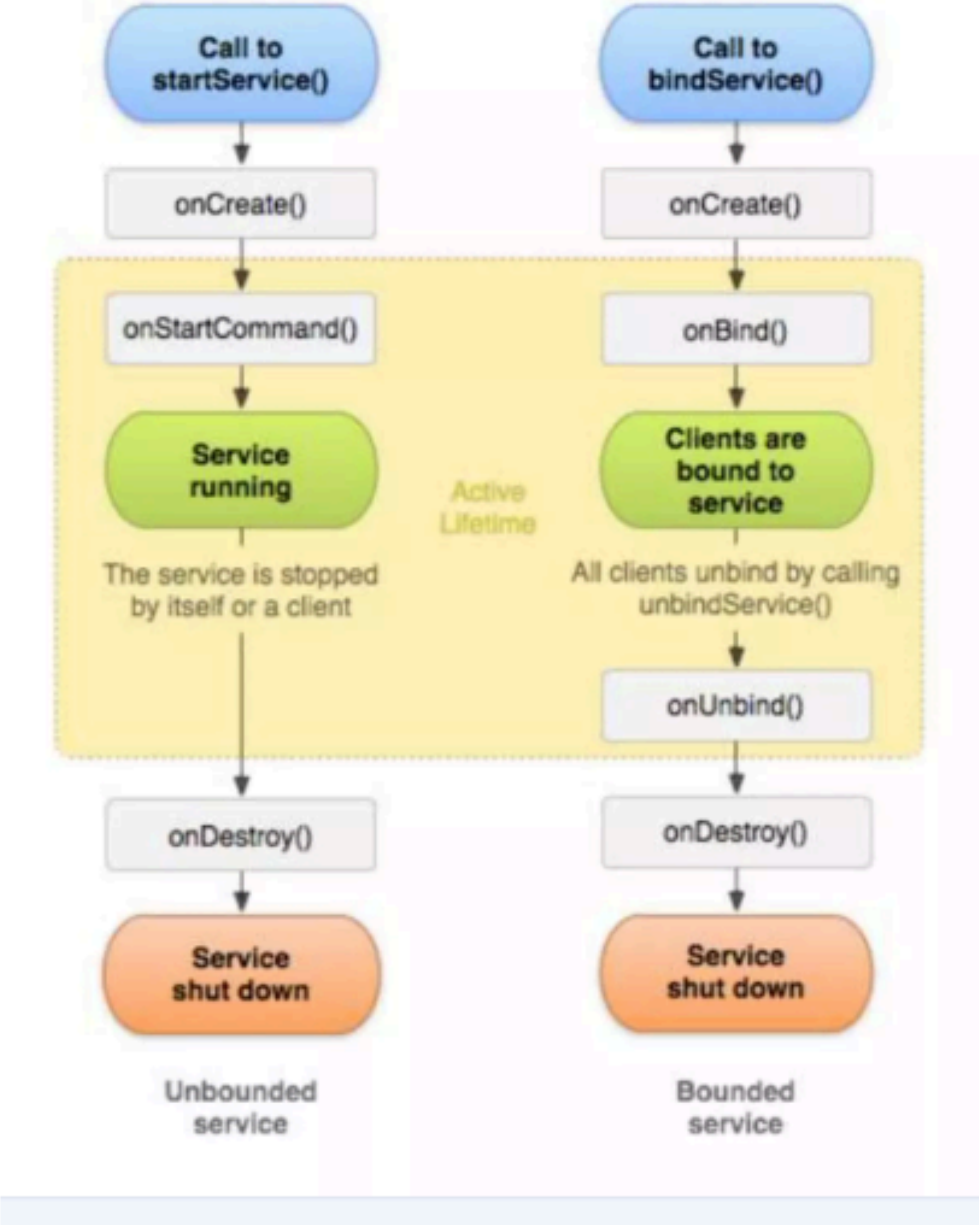
Nguyen Thuy An

- Android Services are components used to perform long-running operations in the background without a user interface, not depend on the app's UI

- Ensure tasks continue even if the user switches activities or apps.

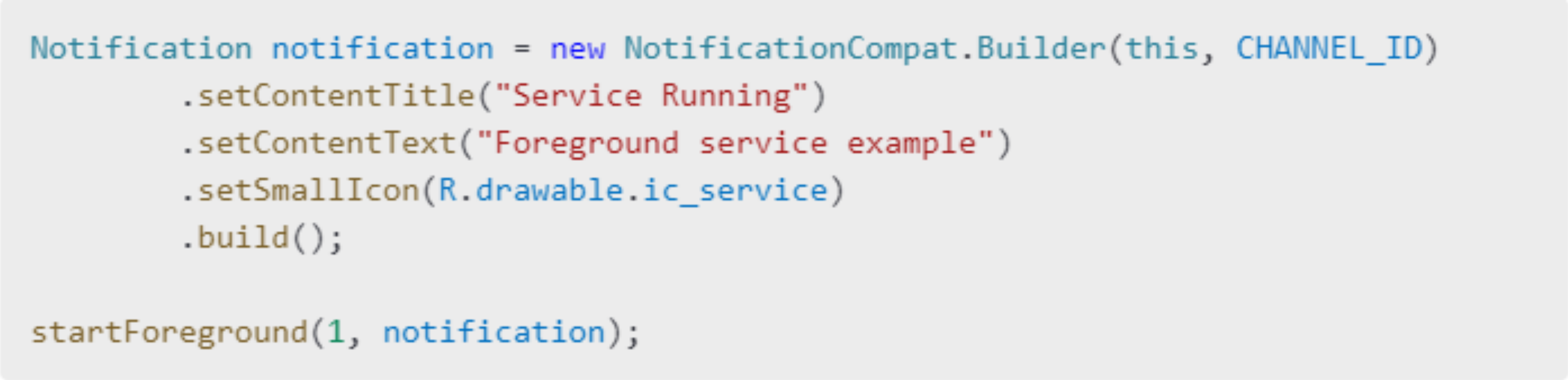
- Runs on the main thread by default. Can work in the background for processes like music playback, network operations, or data syncing.

- Service Life Cycles
    - Started:
        - `startService`() or `startForegroundService()`
        - `onCreate()->onStartCommand()->(Service chạy)->onDestroy()`
    - Bound:
        - `bindService()`
        - `onCreate()->onBind()->(Service chạy)->onUnbind()-> onDestroy()`



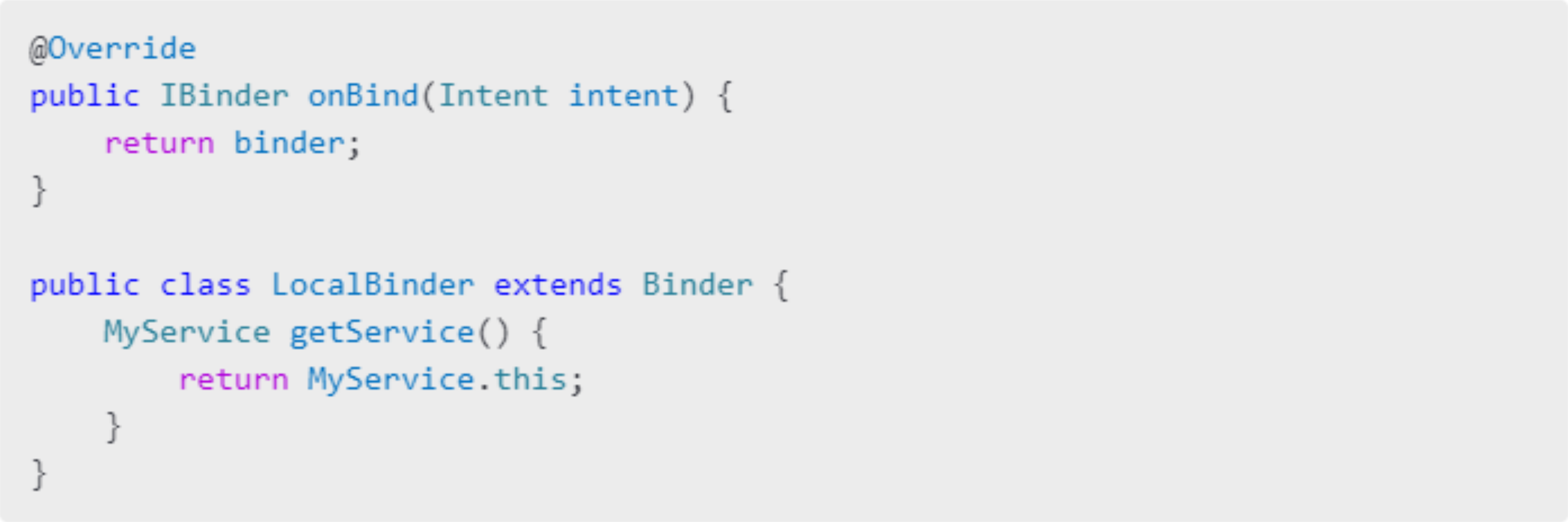
Call to startService()
Call to bindService()
onCreate()
onCreate()
onStartCommand()
onBind()
Service running
Active Lifetime
Clients are bound to service
The service is stopped by itself or a client
All clients unbind by calling unbindService()
onUnbind()
onDestroy()
onDestroy()
Service shut down
Service shut down
Unbounded service
Bounded service

- Foreground Service: runs visibly with a notification
    - music playback, Gps tracking, file download
    - stays alive when app is terminated
- Background Service: runs without user awareness, improve battery life
    - WorkManager, JobScheduler
    - run only app is running
- Bound Service: allows components (e.g., activities) to bind and interact with it
    - message apps, media players



```java
Notification notification = new NotificationCompat.Builder(this, CHANNEL_ID)
        .setContentTitle("Service Running")
        .setContentText("Foreground service example")
        .setSmallIcon(R.drawable.ic_service)
        .build();

startForeground(1, notification);
```



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

public class LocalBinder extends Binder {
    MyService getService() {
        return MyService.this;
    }
}
```

- Declare in AndroidManefest.xml
- Used to do long-term background work
- It is implemented to do something in non-UI thread
- Can interactive with other component