

SERVICE

Benoît Paques

Service

- Composant applicatif servant à effectuer des opérations potentiellement longues en background
- N'a pas d'interface graphique
- Continue de tourner même si l'utilisateur change d'application
- Peut interagir avec d'autres composants (binding)
- Permet une communication interprocess (IPC)
- Doit être ajouté dans le manifeste

```
<manifest ... >
  ...
  <application ... >
    <service android:name=".ExampleService" />
    ...
  </application>
</manifest>
```

États d'un Service

- « Started »
 - Un Service est « started » quand un composant application (Activity, ...) le démarre en appelant `startService()`
 - Une fois démarré il peut tourner indéfiniment
 - Habituellement un Service effectue une seule opération et ne retourne pas de résultat
 - Ex : Téléchargement d'un fichier, lecture de musique
 - Le Service doit s'arrêter lui même

États d'un Service

- « Bound »
 - Un Service est « bound » quand il est démarré par `bindService()`
 - Il offre une interface client-serveur qui permet une interaction avec le composant qui s'est lié à lui
 - Il tourne aussi longtemps qu'il y a des composants qui lui sont liés
 - Plusieurs composants peuvent se lier au Service

Création

- Il faut créer une classe fille de Service ou une de ses sous classes
- Il y a plusieurs callback à override lié au cycle de vie du Service
 - onStartCommand() : le système appelle cette méthode quand un autre composant démarre le Service avec startService(). Une fois exécutée le service est « started » et peut tourner indéfiniment, il faut penser à l'arrêter en appelant stopSelf() ou stopService(). Non nécessaire si le Service ne doit servir qu'en mode « bound ».
 - onBind() : le système appelle cette méthode quand un autre composant veut se lier au Service en appelant bindService(). Il faut fournir une interface de communication en retournant un objet IBinder. Il faut toujours implémenter cette méthode, si le Service ne doit servir qu'en mode « started » il faut retourner null.
 - onCreate() : le système appelle cette méthode à la création du Service avant onStartCommand() ou onBind(). Elle n'est appelée qu'une seule fois à la création.
 - onDestroy() : le système appelle cette méthode quand le Service va être détruit. Il faut nettoyer toutes les ressources, Thread, Listener, Receiver. C'est la dernière méthode appelée.

Démarrage d'un Service

- `startService(Intent intent)`
- `bindService(Intent intent)`
- Tourne dans le même processus que l'application
- Tourne dans le Thread UI
- En cas de calcul lourd il faut démarrer un nouveau Thread dans le Service

```
Intent intent = new Intent(this, HelloService.class);  
startService(intent);
```

2 classes de Service

- Service :
 - Classe de base. Il faut créer un nouveau Thread lors de calcul lourd.
- IntentService
 - Sous classe de Service qui utilise un worker thread pour effectuer les requêtes une par une. Il suffit d'implémenter la méthode `onHandleIntent()` qui est appelée à chaque nouvel Intent lancé.

Example IntentService

```
public class HelloIntentService extends IntentService {

    /**
     * A constructor is required, and must call the super IntentService(String)
     * constructor with a name for the worker thread.
     */
    public HelloIntentService() {
        super("HelloIntentService");
    }

    /**
     * The IntentService calls this method from the default worker thread with
     * the intent that started the service. When this method returns, IntentService
     * stops the service, as appropriate.
     */
    @Override
    protected void onHandleIntent(Intent intent) {
        // Normally we would do some work here, like download a file.
        // For our sample, we just sleep for 5 seconds.
        long endTime = System.currentTimeMillis() + 5*1000;
        while (System.currentTimeMillis() < endTime) {
            synchronized (this) {
                try {
                    wait(endTime - System.currentTimeMillis());
                } catch (Exception e) {
                }
            }
        }
    }
}
```


TP #11 Service

- Créer un Service en mode « bound » et un Service en mode « started »
 - Service « bound » :
 - Les Activity doivent afficher la valeur de l'incrément que le Service lui renvoie
 - Lancer un Service qui incrémente une variable à chaque fois qu'une Activity se bind à lui
 - Service « started » :
 - 1 Activity permettant de démarrer/stopper la lecture
 - Lance la lecture d'un flux audio

TP #11 Notation

- 2 point pour les services
- 1 point pour les Activity affichant la valeur retournée
- 1 point pour l'Activity controlant la lecture audio
- 1 point pour la lecture d'un fichier audio