

## Android

### Services - Broadcast Receiver

C. Raïevsky



## Service - Définition

Un service est :

- ▶ Un élément autonome d'une application Android ;
- ▶ Au même titre qu'une activité ;
- ▶ Sans interface utilisateur ;
- ▶ Qui s'exécute dans le thread principal de l'application.

Un service survit à l'activité qui l'a créé

## Rappels

### Thread UI

- ▶ Ne JAMAIS effectuer d'opération longue dans le thread UI.

### Solutions

- ▶ Threads + `runOnUiThread`
- ▶ Handlers
- ▶ Obsolete : `Async Task`

### Services

- ▶ Si l'opération doit continuer même lorsque l'utilisateur change d'activité

## Deux principaux modes d'interaction entre activités et service

### Started

- ▶ Exécution autonome après démarrage explicite
- ▶ Service responsable de s'arrêter proprement
- ▶ Possibilité de démarrer un service après un délai (Scheduled)

### Bound

- ▶ Établissement d'un lien entre l'activité et le service
- ▶ Service pouvant être déjà existant, créé sinon
- ▶ Le système va détruire un service qui n'est plus lié
- ▶ Interaction via une sorte d'API

## IntentService

### Sous classe de Service

- ▶ Gère la création d'un thread à part
- ▶ Similaire à une AsyncTask

## Cas d'étude - Télécharger et visionner une vidéo

### On souhaite permettre à l'utilisateur :

- ▶ de télécharger une vidéo qu'il a choisi,
- ▶ de la visionner une fois qu'elle est téléchargée.

### Pour ne pas bloquer l'interface durant le téléchargement

La récupération de la vidéo se fera en utilisant un `IntentService`.

## IntentService - Démarrage du service

### IntentService

- ▶ Pour simplifier la réalisation d'un travail en arrière plan.
- ▶ Reçoit les informations nécessaires à la tâche dans l'Intent.
- ▶ Exécute la tâche.
- ▶ Se termine tout seul.

### Lancement du service

```
mServiceIntent = new Intent(this, VideoDownloadService.class);
mServiceIntent.setData(Uri.parse(videoURL));
```

```
this.startService(mServiceIntent);
```

## IntentService - Implémentation minimale

```
public class VideoDownloadService extends IntentService {
    ...
    @Override
    protected void onHandleIntent(Intent currentIntent) {

        Uri data = currentIntent.getData();
        ...
        // Download video
        ...
        onDownloadFinished();
    }
    ...
}
```

## IntentService - Déclaration dans le manifeste

```

...
<application
  android:icon="@drawable/icon"
  android:label="@string/app_name">
  ...
  <service
    android:name=".VideoDownloadService"
    android:exported="false"/>
  ...
</application>
...

```

9 / 14

## IntentService - Retour du résultat à l'activité

Envoi d'un Broadcast à partir du Service

```

public class VideoDownloadService extends IntentService {
  ...
  public static final String BROADCAST_ACTION =
    "fr.iut.valence.exemple.BROADCAST";

  public static final String EXTENDED_DATA_STATUS =
    "fr.iut.valence.exemple.STATUS";

  public void onDownloadFinished(){

    Intent localIntent = new Intent(BROADCAST_ACTION);
    localIntent.putExtra(EXTENDED_DATA_STATUS, status);
    // Broadcasts the Intent to receivers in this app.
    LocalBroadcastManager.getInstance(this).sendBroadcast(localIntent);
  }
  ...
}

```

10 / 14

## IntentService - Retour du résultat à l'activité

Réception d'un Broadcast dans l'activité

```

public class VideoActivity extends Activity{
  ...
  private class ResponseReceiver extends BroadcastReceiver
  {
    // Called when the BroadcastReceiver gets an Intent
    // it has registered to receive.
    @Override
    public void onReceive(Context context, Intent intent) {
      ...
      // Handle Intents here.
      // i.e. Display a Notification
      ...
    }
  }
}

```

11 / 14

## IntentService - Retour du résultat à l'activité

Enregistrement du BroadcastReceiver

### En Java

```

public class VideoActivity extends Activity{
  ...
  public void onCreate(Bundle bundle){
    ...
    IntentFilter mStatusIntentFilter = new IntentFilter(
      VideoDownloadService.BROADCAST_ACTION);

    ResponseReceiver rr = new ResponseReceiver();
    LocalBroadcastManager.getInstance(this).registerReceiver(
      rr, mStatusIntentFilter);
  }
  ...
}

```

12 / 14

## À retenir

### Lancement d'un Service

- ▶ Toujours avec un Intent explicite.
- ▶ Sinon pas de garantie que ce sera le votre qui sera appelé.
- Faille de sécurité potentielle.

### Résultat du service

- ▶ Pas passé à l'activité dans un Intent explicite
- cause la mise au premier plan de l'activité
- casse le flot d'utilisation

### Dans un BroadcastReceiver

- ▶ Ne pas réaliser d'action en arrière plan → L'objet BroadcastReceiver a une durée de vie limitée.
- ▶ Ne pas lancer d'activité → casse le flot d'utilisation

C'est la fin.

:wq

Questions ?