

# Android - Les bonnes intentions

Jérémy S. Cochoy

INRIA Paris-Saclay | [jeremy.cochoy@gmail.com](mailto:jeremy.cochoy@gmail.com)

Mars 2017

1 GIT et GitHub

2 Intention

3 Activités

Votre nouveau livre de chevet.

*<https://developer.android.com/guide/index.html>*

## GitHub



# Qu'est-ce qu'une intention ?

## Une Intent

Une Intent est un objet qui représente un message, et permet de demander une action à un Composant.

## Les usages principaux :

- Lancer une activité.
- Lancer un service.
- Délivrer un message.

# Démarrer une activité

## Activity

Une Activity représente un écran dans une app. On peut lancer une activité en donnant un Intent à `startActivity()`. L'intention décrit l'activité et lui fournit les données.

## Avec résultat

Si l'activité doit fournir un résultat, on dispose de `startActivityForResult()` et de `onActivityResult()`.

# Démarrer un service

Un service effectue des actions en arrière plan, sans interface.

- One time operation : Donner un Intent à `startService()`.
- Si le service a une interface client-serveur, on dispose de `bindService()`.

# Délivrer un message

## Broadcast

Un broadcast est un message que n'importe quelle application peut recevoir.

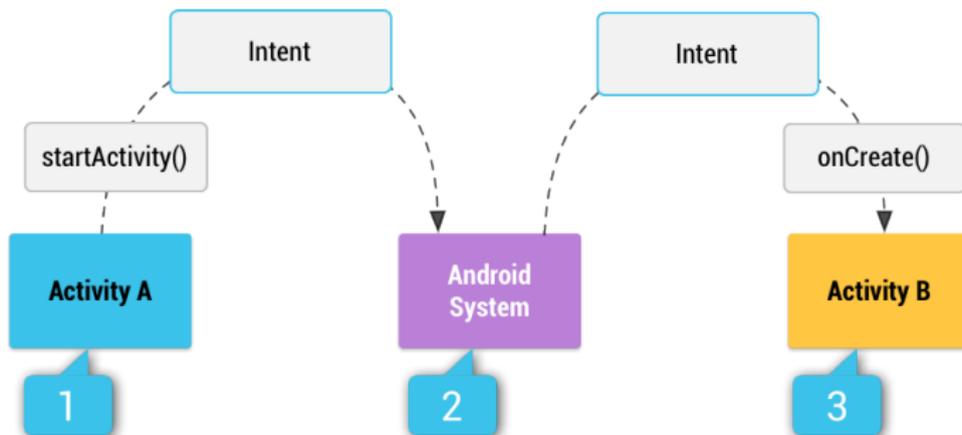
## Délivrer un message

Donner un Intent à `sendBroadcast()` ou `sendOrderedBroadcast()`.

## Deux types d'Intent :

- Explicite : Contient le nom du composant explicite.
- Implicite : Décrit une action à effectuer (url, localisation, ...).

## Filtres



# Exemple de filtre

## Filtrer le partage d'images

```
<activity android:name="ShareActivity">
  <intent-filter>
    <action android:name="android.intent.action.SEND"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="image/*"/>
  </intent-filter>
</activity>
```

# Construire une intention

Une intention contient les informations suivantes :

- Nom du composant
- Action
- Data
- Category
- Extras

# Nom du composant

Le nom du composant à lancer

Nécessaire pour une intention explicite.

Un Service doit toujours être lancer explicitement.

Le champ de l'`Intent` est un objet `ComponentName` qui contient un chemin complet vers le nom de la classe. Par exemple : `com.example.ExampleActivity`. La valeur peut être fixée avec `setComponent()`, `setClass()`, `setClassName()`, ou le constructeur de l'`Intent`.

# Action

Une chaîne qui spécifie le type d'action à effectuer.

## Quelques actions classiques

- `ACTION_VIEW` Avec `startActivity()` pour afficher une information (Photo, localisation...)
- `ACTION_SEEND` Avec `startActivity()` , pour "partager" (share).
- `ACTION_EDIT` Fournit un document à éditer.
- `ACTION_PICK` Sélectionne un fichier / image / etc...

# Data

L'objet `Uri` qui contient une référence / lien vers les données à transmettre.

Il est bon de préciser le type.

Les méthodes `setData()`, `setType()` sont exclusives, mais `setDataAndType()` permet de spécifier les deux.

# Extra

Permet de passer des données sous la forme clef-valeur.

Via la méthode `putExtra()`, ou un `Bundle` avec `putExtras()`.

## Exemple :

Dans un `ACTION_SEND`, on peut vouloir ajouter les valeurs `EXTRA_EMAIL` et `EXTRA_SUBJECT` pour envoyer un e-mail.

# Catégories

- Permet d'être plus précis sur quelles activités sont concernées.
- La `CATEGORY_DEFAULT` convient à la plupart des usages.
- Par exemple, `CATEGORY_BROWSABLE` spécifie que l'élément transmit peut être ouvert par un navigateur web.

# Exemple explicite :

## Explicit intent

```
// Executed in an Activity , so 'this' is the Context
// The fileUrl is a string URL, such as "http://www.example.
    com/image.png"
Intent downloadIntent = new Intent(this , DownloadService.
    class);
// DownloadService is the name of the app.
downloadIntent.setData(Uri.parse(fileUrl));
startService(downloadIntent);
```

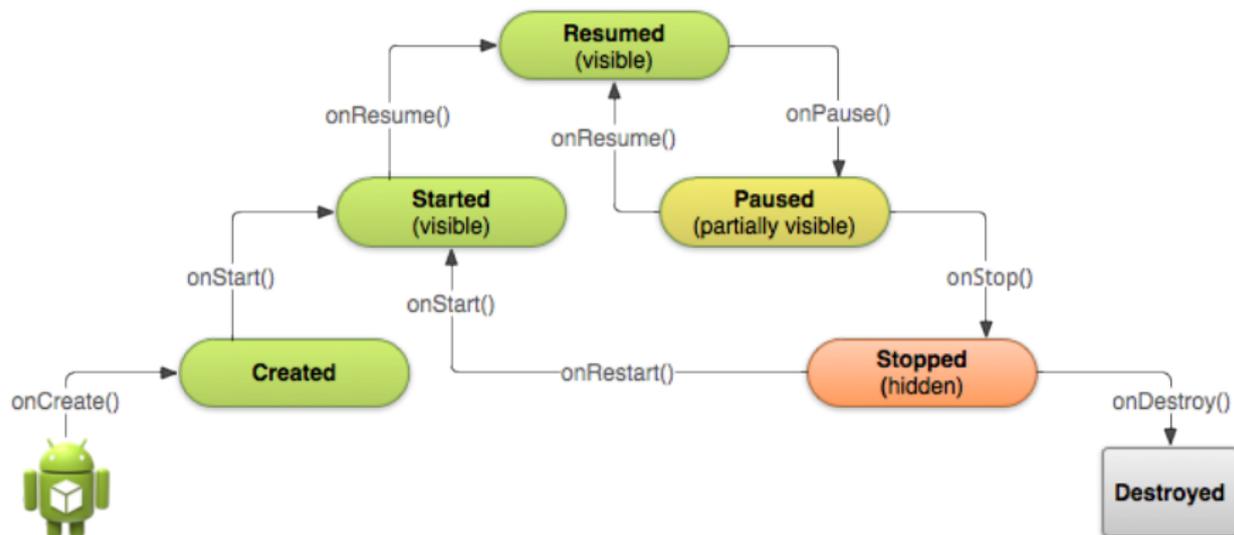
# Exemple implicite :

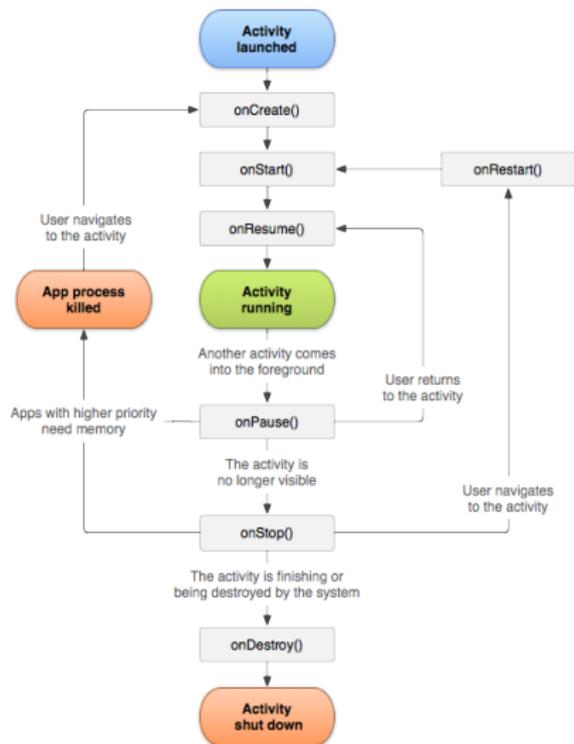
## Implicit intent

```
// Create the text message with a string
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, textMessage);
sendIntent.setType("text/plain");

// Verify that the intent will resolve to an activity
if (sendIntent.resolveActivity(getPackageManager()) != null)
{
    startActivity(sendIntent);
}
```

# Cycle de vie d'une activité





```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "
        resumed").
    }
}
```

```
@Override
protected void onPause() {
    super.onPause();
    // Another activity is taking focus (this activity is
    // about to be "paused").
}
@Override
protected void onStop() {
    super.onStop();
    // The activity is no longer visible (it is now "
    // stopped")
}
@Override
protected void onDestroy() {
    super.onDestroy();
    // The activity is about to be destroyed.
}
}
```

Merci et à bientôt.

