

# Android TD 3

## Intents (suite)

---

Le but de ce TD est de programmer une seconde Activity dans l'application qui sera appelée par Intent. Pour vous aider, vous pouvez vous appuyer sur la page suivante : <http://developer.android.com/training/basics/firstapp/starting-activity.html>.

## 1 Démarrage de la 2nd Activity

1. Créer un nouveau projet.
2. Créer un 2nd fichier Java avec la définition d'une nouvelle Activity (copier-coller du fichier principal et changement du nom du fichier et de la classe).
3. Créer un nouveau fichier layout pour cette Activity (dans res/layout) avec un champs de texte "Hello 2nd Activity" et modifier la classe de l'Activity afin de charger ce layout.
4. Déclarer la 2nd Activity dans le fichier AndroidManifest.xml sur le modèle suivant :

```
<?xml version="1.0" encoding="utf-8"?>
...
<application android:label="@string/app_name" ...>
...
    <activity android:name="nomdelactivity"
              android:label="nomdelactivity"
              android:parentActivityName="nomdelactivityprincipale">
    </activity>
</application>
</manifest>
```

5. Ajouter un bouton dans le layout de l'Activity principale avec un évènement provoquant la création de l'Intent vers la 2nd Activity sur le modèle :

```
public void startAnotherActivity(View view)
{
    Log.i("UserActivity", "button_pressed");
    startActivity(new Intent(this, AnotherActivity.class));
}
```

Noter qu'il suffit ici de spécifier le fichier .class à charger pour la 2nd Activity.

6. Tester cette application et ajouter des logs pour voir ce qui se passe!

## 2 Passage de données vers la 2nd Activity

L'objectif ici est de passer un texte de l'Activity principale vers la seconde Activity.

1. Remplacer le TextView du layout principal par un EditText
2. Avant le lancement de l'Intent, récupérer le texte de l'EditText et le passer dans l'Intent sur le modèle suivant :

```
intent.putExtra("text", message);
startActivity(intent);
```

3. Dans la seconde Activity, à la fin de la méthode onCreate, récupérer le message passé dans l'Intent :

```
Intent intent = getIntent();
String message = intent.getStringExtra("text");
```

et modifier le texte du TextView avec ce message.

4. Tester cette application et ajouter des logs pour voir ce qui se passe!

### 3 Récupération de données de la 2nd Activity

On souhaite également récupérer un message de la 2nd Activity. Dans cet exercice, on passe un message à la 2nd Activity puis on l'édite dans cette Activity et on le retourne une fois cette Activity terminée afin de l'afficher.

1. Les 2 Activities doivent avoir un champs de saisie EditText pour afficher et modifier le message à communiquer entre les Activities.
2. Modifier la manière dont est lancée la 2nd Activity :

```
startActivityForResult(intent, 1);
```

3. Le message est retourné par la 2nd Activity sous forme d'Intent. Cette récupération est géré par la méthode onActivityResult à surcharger de la manière suivante :

```
@Override
public void onActivityResult(int requestCode,
                             int resultCode,
                             Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);
    String message = intent.getStringExtra("text");
    ((EditText)findViewById(R.id.et0)).setText(message);
}
```

4. Dans la 2nd Activity, il faut créer l'Intent à retourner lors de la fermeture de l'Activity, donc en surchargeant sa méthode finish :

```
@Override
public void finish()
{
    Log.i("UserActivity3", "finish");
    Intent intent = new Intent();
    EditText et0 = ((EditText)findViewById(R.id.et0));
    String message = et0.getText().toString();
    intent.putExtra("text", message);
    setResult(Activity.RESULT_OK, intent);
    super.finish();
}
```