

TP Android

Les Intents avec passage de données

Romain Raveaux
Polytech'Tours

Dans le TP précédent, nous avons vu comment démarrer une nouvelle fenêtre par le biais d'une action sur un bouton posé sur une vue active. A présent, nous allons voir comment une activité 1 peut transmettre des données à une autre activité.

Sommaire

1°) Passer des données entre Activity sous android.....	2
Fenêtre 1 vers Fenêtre 2.....	2
Retour vers la Fenêtre 1.....	5
2°) Passage d'objets en entre fenêtres.....	6
3°) à Rendre :.....	6

1°) Passer des données entre Activity sous android

Dans ce TP, nous allons développer une application permettant de naviguer entre deux fenêtres à l'aide d'Activity, tout en passant un objet entre ces deux fenêtres.

Dans chaque fenêtre, nous disposerons un bouton et un champ de texte (EditText).

Dans le sujet du TP, vous trouverez du code. Ce dernier n'est pas forcément complet et c'est à vous de le compléter. N'oubliez pas de créer le code JAVA et les fichiers XML correspondant.

Fenêtre 1 vers Fenêtre 2

1°) A faire : Créez 2 fenêtres fenetre1 et fenetre2. Dans chaque fenêtre, nous disposerons un bouton et un champ de texte (EditText).

2°) A faire :Récupération des objets de la vue graphique

```
Button btaction = (Button) findViewById(R.id. Btecrire);
```

3°) A faire :Associer un événement à un objet

```
//Création du listener
```

```
OnClickListener monlisteneraction = new Button.OnClickListener() {  
    public void onClick(View v) {  
        //Todo  
    }  
};
```

```
//on ajoute le listener au bouton
```

```
btaction.setOnClickListener( monlisteneraction );
```

4°) Envoyer une donnée à une autre activity.

```
//création de notre intent
```

```
Intent defineIntent = new Intent(lecontext, FenetreClass2.class);
```

```
// objet qui vas nous permettre de passe des variables ici la variable passInfo
```

```
Bundle objetbunble = new Bundle();
```

```
objetbunble .putString("passInfo",textchampsaisie.getText().toString());
```

```
// on passe notre objet a notre activities
```

```
defineIntent.putExtras(objetbunble );
```

```
// on appelle notre activité
```

```
lecontext.startActivity(defineIntent);
```

Un exemple de code vous est proposé ci-dessous.

```
public class FenetreClass1 extends Activity{
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.fenetre1);
        lecontext = this;
        //récupération du text dans le champ de saisie
        final EditText textchampsaisie = (EditText) findViewById(R.id.text);
        Button btaction = (Button) findViewById(R.id. btecrire);
        //action sur le bouton click appelle de la nouvelle activité
        btaction.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View v) {
                //création de notre item
                Intent defineIntent = new Intent(lecontext, FenetreClass2.class);
                // objet qui vas nous permettre de passe des variables ici la variable passInfo
                Bundle objetbunble = new Bundle();
                objetbunble .putString("passInfo",textchampsaisie.getText().toString());
                // on passe notre objet a notre activities
                defineIntent.putExtras(objetbunble );
                // on appelle notre activité
                lecontext.startActivity(defineIntent);
            }
        });
    }
}
```

Récupération des informations passées par la classe **FenetreClass1.class** à la classe **FenetreClass2.class**

```
public class FenetreClass2 extends Activity{
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.fenetre2);
        lecontext = this;
        lecontext.setTitle("fenetre 2");
        //récupération du text dans le champ de saisie
        final EditText textchampsaisie = (EditText) findViewById(R.id.text);
        Bundle objetbunble = this.getIntent().getExtras();
        // récupération de la valeur
        String InfoPasse= objetbunble .getString("passInfo");
        // on affiche l'information dans l'edittext
        textchampsaisie .setText(InfoPasse);
    }
}
```

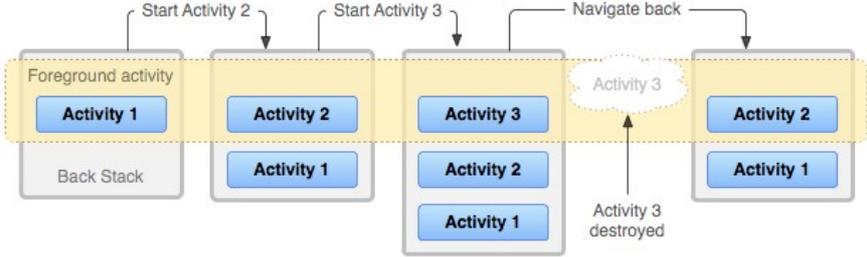
Ensuite le plus important, il faut déclarer ces différentes activity dans le fichier AndroidManifest.xml. Un exemple vous est proposé ci-dessous :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.com.test"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name=".FenetreClass1" android:label="@string/app_name">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

    <activity android:name=".FenetreClass2"
    android:screenOrientation="portrait"></activity>

</application>
    <uses-sdk android:minSdkVersion="3" />
</manifest>
```

1°) Construire le diagramme des états des activités durant ces actions : (voir exemple ci-dessous)



Retour vers la Fenêtre 1

Nous avons réussi à passer une information, de la fenêtre 1 vers la fenêtre 2. Maintenant, nous souhaitons que la fenêtre 2 retourne une information à la fenêtre 1 par l'appuie sur un bouton.

1°) Dans la fenêtre 1, il faut démarrer l'activity avec l'intention d'attendre un résultat.

```
lecontext.startActivityForResult(defineIntent,Fenetre1ID);
```

Fenetre1ID : est un int, un identifiant choisi arbitrairement représentant l'ID de l'activity Fenetre1.

2°) Dans la fenêtre 2, placer les éléments nécessaires afin d'envoyer la chaîne de caractères HelloWorld à la fenêtre 1 lors de l'appuie sur un bouton.

- Le listener sur le bouton
- Le bundle à envoyer à l'autre fenêtre.
- L'intent de fenêtre 1 vers fenêtre 2
- Un code retour (0 par exemple)

Terminer votre fonction par la commande suivante :

```
//Envoie le résultat suivant l'intent défini.
```

```
setResult(coderetourok,intent);
```

```
//Demande la destruction de la fenêtre courante.
```

```
finish();
```

3°) La fenêtre 1 doit pouvoir capter les informations envoyées par la fenêtre 2. Pour ce faire, il faut implémenter dans la fenêtre 1 la fonction onActivityResult :

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    // Si le résultat provient d'une demande de la fenêtre1
    if (requestCode == this.Fenetre1ID) {

        // le code retour est bon
        if (resultCode == coderetourok) {

            //récupérer les informations
            //et les afficher dans TextView
        }
    }
}
```

Récupérer les informations transmises par la fenêtre 2 et afficher les dans le TextView de la fenêtre 1.

4°) Construire le diagramme des états des activités durant ces actions.

2°) Passage d'objets en entre fenêtres

- Créer une classe `Personne` qui contiendra le nom, le prénom et l'âge d'une personne.
- La fenêtre 1 permettra de remplir des champs de texte contenant le nom, le prénom et l'âge d'une personne.
- Un bouton « `Submit` » enverra les informations de la fenêtre 1 vers une fenêtre 2. La fenêtre 1 enverra un objet de type `Personne` à la fenêtre 2.
- La fenêtre 2 affichera le nom et le prénom de la personne
- Conclure sur la contrainte liée aux passages d'objets entre Activity