



Android: Activity

Assane SECK
Ingénieur-Informaticien



Sommaire

Les Activity

Cycle de Vie d'une Activity

onCreate() et onDestroy()

onPause() et onResume()

onSaveInstanceState()

Plusieurs Activity

Démarrer d'autres Activity de
manière explicite

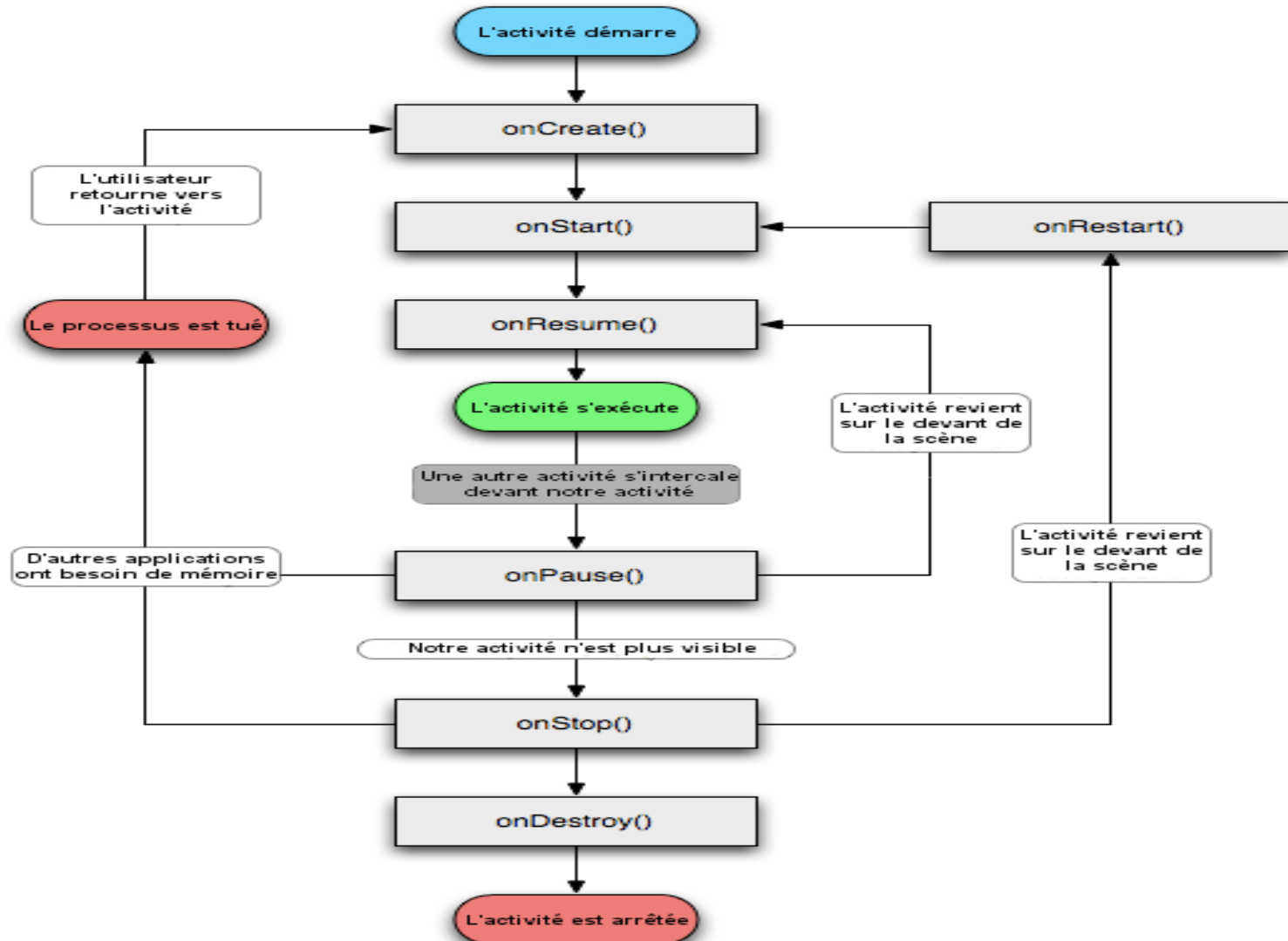
Obtenir le résultat d'une Activity



ANDROID



Cycle de Vie d'une Activity



DIFFERENTS ETATS D'ACTIVITY

- ▶ Etats les plus communs:
 - ▶ **Actif** : l'activity a été démarrée par l'utilisateur et apparaît en avant-plan
 - ▶ **Pause** : l'activity a été démarrée, tourne, est toujours visible, mais quelque chose est venu entraver son fonctionnement: un dialogue, un appel entrant,... L'utilisateur ne peut interagir avec l'activity.
 - ▶ **Arrêté** : l'activity a été lancée par l'utilisateur, tourne, mais n'est pas visible car cachée par une autre Activity.
 - ▶ **Morte** : l'activity n'a jamais été démarrée ou a été terminée par le système pour une raison (manque de mémoire,...)
-



ONCREATE ET ONDESTROY

onCreate()

- ▶ Lorsqu'une Activity est lancée pour la première fois, cette fonction est appelée mais avec un paramètre qui équivaut à null.
- ▶ Si l'Activity était en cours d'exécution mais a été terminée pour l'une ou l'autre raison, un Bundle est donné en paramètre. Le Bundle peut avoir été rempli avant avec la fonction **onSaveInstanceState** lorsque l'Activity fut détruite.
- ▶ Si l'Activity a différentes ressources basées sur différents états ou configurations (exemple: layout portrait ou landscape)

onDestroy()

- ▶ Quand une Activity se ferme (via un appel à la fonction **finish()**)
 - ▶ Quand une Activity est terminée (manque de mémoire,...)
-



ONPAUSE, ONRESUME.....

onResume()

- ▶ appelé peu avant qu'une Activity réapparait en avant-plan
- ▶ après le lancement de l'Activity
- ▶ après être relancé depuis un état arrêté
- ▶ après qu'un popup disparaisse
- ▶ c'est le moment parfait pour faire des mises à jour sur le UI thread

onPause()

- ▶ appelé lorsque quelque chose vole l'attention d'une Activity
- ▶ une autre Activity

onStop

- ▶ Appelée quand votre activité n'est plus du tout visible quelque soit la raison.
- ▶ Dans cette méthode vous devez arrêter tous les traitements et services exécutés par votre application.

onDestroy()

- ▶ arrêter des threads
 - ▶ libérer des ressources (camera,...)
-



PLUSIEURS ACTIVITES

- ▶ Une application contient souvent plusieurs Activity
- ▶ Ces Activity sont accessibles via des «Intents»
- ▶ L'Activity principale est lancée via le launcher Android
- ▶ Une Activity peut être lancée de plusieurs façons:
 - Explicite: Vous savez exactement quelle Activity lancer
 - Implicite: Vous avez une action ou un Uri vers quelque chose, mais vous ne savez pas quelle Activity peut traiter ceci.



LANCER UNE ACTIVITY DEPUIS NOTRE PROJET

Il existe deux types d'Activity:

- ▶ Une Activity pour laquelle nous aimerions avoir un résultat
- ▶ Exemple: authentication sur un site web
- ▶ Une Activity pour laquelle nous n'attendons pas de résultat
- ▶ Exemple: écriture d'un email



DEMARRER UNE ACTIVITY DE MANIERE EXPLICITE

Démarrer une Activity implique deux étapes:

- ▶ Créer un Intent
 - ▶ Utiliser une de ces fonctions sur le contexte avec cet Intent comme paramètre:
 - ▶ `startActivity()` : démarre une activity avec l'Intent. L'Activity ne sera pas informée lorsque l'Activity qui se lance sera complétée.
 - ▶ `startActivityForResult()` : démarre une activity avec un Intent et un numéro au choix, qui représente la request. Lorsque l'Activity lancée termine, elle notifiera l'Activity parent via `onActivityResult()`
 - ▶ **!!! Important: toujours déclarer une Activity dans le manifest au cas ou elle est copiée ou créer manuellement !!!**
 - ▶ `Intent i = new Intent(this, DisplayEvent.class);`
 - ▶ `i.putExtra(DisplayEvent.ID, 10);`
 - ▶ `startActivity(i);`
-



OBTENIR LE RESULTAT DES ACTIVITY

- ▶ Lorsqu'un appel est fait à `startActivityForResult()`, `onActivityResult()` recevra plusieurs paramètres lorsque l'Activity enfant terminera:
- ▶ Request code, afin d'identifier l'Activity pour laquelle nous obtenons un résultat
- ▶ Result code
`RESULT_OK`, `RESULT_CANCELED`,...
- ▶ Un Intent
- ▶ Cet intent contient plus d'informations sur le résultat
- ▶ Par exemple: token d'authentification



METTRE LE RESULTAT DEPUIS UNE ACTIVITY ENFANT

- ▶ L'Activity enfant met le résultat en faisant appel à setResult
- ▶ Paramètres
- ▶ Resultcode : RESULT_OK, RESULT_CANCELED,...
- ▶ Un Intent optionnel



Fonctionnalités Essentielles

- ▶ **Intent(Context, ClasseRedirect);**
- ▶ **startActivityForResult(intent, requestCode);**
- ▶ **protected void onActivityResult(int requestCode, int resultCode, Intent data) {**
 - data.getStringExtra("NOM");**
- ▶ **}**
- ▶ **intent.putExtra("CLE " , "VALEUR")**
- ▶ **setResult(RESULT_OK, i);**



Multiple Activity Exemple: parent.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center_vertical"
    android:orientation="vertical">
    <TextView
        android:id="@+id/labelParent"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
    <Button
        android:id="@+id/btparent"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Entrez votre nom" />
</LinearLayout>
```

Multiple Activity Exemple: child.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <EditText
        android:id="@+id/textChild"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="Votre Nom" />
    <Button
        android:id="@+id/btChild"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Valider" />
</LinearLayout>
```



Multiple Activity Exemple: AndroidManifest.xml

```
<activity  
    android:name="com.example.univ.ChildActivity"  
    android:label="Child Activity" >  
</activity>
```

A ajouter dans la balise application du fichier AndroidManifest.xml au cas ou elle est absente



Multiple Activity: Code Java Child Activity

```
public class ChildActivity extends Activity {
    EditText txtnom;
    Button btOk;
    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.child);
        btOk = (Button) findViewById(R.id.btChild);
        txtnom = (EditText) findViewById(R.id.textChild);
        btOk.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                Intent i = new Intent();
                i.putExtra("NOM", txtnom.getText().toString());
                setResult(RESULT_OK, i);
                finish();
            }
        });
    }
}
```


Multiple Activity: Code Java Activity Parent

```
public class MainActivity extends Activity {
    Button btParent;
    TextView labelParent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.parent);
        labelParent = (TextView) findViewById(R.id.LabelParent);
        btParent = (Button) findViewById(R.id.btparent);
        btParent.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                Intent intent = new Intent(MainActivity.this,
                ChildActivity.class);
                startActivityForResult(intent, 123456);
            }
        });
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        if (requestCode == 123456) {
            if (resultCode == RESULT_OK) {
                labelParent.setText(data.getStringExtra("NOM"));
            }
        }
    }
}
```

Exercice

- ▶ Créer une page qui permet la navigation entre les pages.
- ▶ Créer un formulaire qui permettra la saisie des champs (Nom, Prénom, Adresse, Genre, Filière, Classe)
- ▶ Créer une page résumé qui présentera les informations saisies sur le formulaire précédent.



FIN

