
Intents

Utilisation, Exemples...

Définition et Utilisations

Intents

- Une application Android peut contenir plusieurs activités :
 - Une activité utilise la méthode *setContentView* pour s'associer avec une interface graphique
 - Les activités sont indépendantes les unes des autres, cependant, elles peuvent collaborer pour échanger des données et des actions
 - Typiquement, l'une des activités est désignée comme étant la première à être présentée à l'utilisateur quand l'application est lancée : on l'appelle l'activité *de démarrage*
 - Les activités interagissent en mode *asynchrone*.
 - Le passage d'une activité à une autre est réalisé en demandant à l'activité en cours d'exécuter un *Intent*.

Définition et Utilisations

Intent

- Un intent est un message qui peut être utilisé pour demander une action à partir d'un autre composant de l'application
- Un Intent permet invoquer des Activités, des *Broadcast Receivers* ou des *Services*. Les différentes méthodes utilisées pour appeler ces composantes sont les suivantes :
 - `startActivity(intent)` : lance une activité
 - `sendBroadcast(intent)` : envoie un *intent* à tous les composants *Broadcast Receivers* intéressés
 - `startService(intent)` ou `bindService(intent, ...)` : communiquent avec un service en arrière plan.

Construction d'un Intent

Intent

- Un intent comporte des informations que le système Android utilise
- **Nom du composant à démarrer**
- **Action à réaliser**
 - ACTION-VIEW, ACTION_SEND...
- **Donnée**
 - URI référençant la donnée sur laquelle l'action va agir
- **Catégorie**
 - Information supplémentaire sur le type de composants qui va gérer l'intent
 - CATEGORY-BROWSABLE, CATEGORY-LAUNCHER...
- **Extras**
 - Paires clef-valeur qui comportent des informations additionnelles pour réaliser l'action demandée
- **Drapeaux (Flags)**
 - Définissent la classe qui fonctionne comme métadonnée pour cet intent
 - Peuvent indiquer comment lancer une activité, comment la traiter une fois lancée

Types d'Intent

Intent

- Il existe deux types d'Intents
 - Intents Explicites:
 - Spécifient le composant à démarrer par nom (nom complet de la classe)
 - Permettent de démarrer un composant de votre propre application, car le nom de la classe est connu
 - *Exemple*: démarrer une activité en réponse à l'action d'un utilisateur
 - Intents Implicites
 - Ne nomment pas un composant spécifique, mais déclarent une action à réaliser
 - Permet à un composant d'une application d'appeler un composant d'une autre application
 - *Exemple*: montrer à l'utilisateur un emplacement sur une Map

Intents

INTENTS IMPLICITES

Arguments et Utilisation

Intents Implicites

- Les principaux arguments d'un Intent implicite sont :
 - **Action** : l'action à réaliser, peut être prédéfinie (*ACTION_VIEW*, *ACTION_EDIT*, *ACTION_MAIN*...) ou créée par l'utilisateur.
 - **Donnée** : Les données principales sur lesquelles on va agir, tel que le numéro de téléphone à appeler.
- Il est typiquement appelé comme suit:

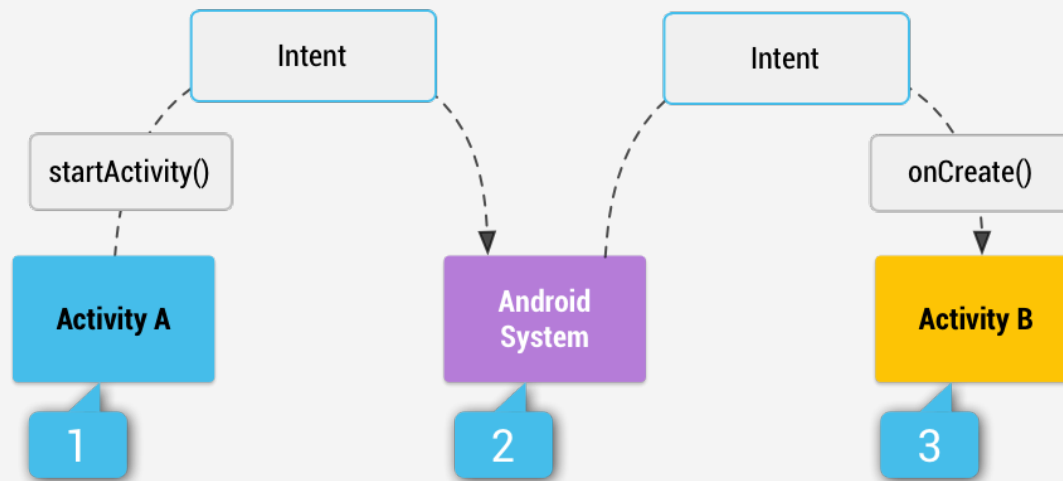
```
Intent myActivityResult = new Intent (<action>, <donnee>) ;  
startActivity (myActivityResult) ;
```



Comportement d'un Intent Implicite

Intents Implicites

- Un intent implicite se comporte comme suit:
 1. Activité A crée un Intent avec une action et le passe en paramètre à *startActivity*
 2. Le système Android cherchent toutes les applications pour trouver un *Intent Filter* qui correspond à cet Intent
 3. Quand une correspondance est trouvée, le système démarrent l'activité (Activity B) en invoquant sa méthode *onCreate* et en lui passant l'intent



IntentFilters

Intents Implicites

- Un Intent Filter est une expression dans le fichier Manifest d'une application qui spécifie le type d'intents que le composant veut recevoir
- Permet aux autres activités de lancer directement votre activité en utilisant un certain Intent
- Si vous ne déclarez pas d'Intent Filters à votre activité, elle ne pourra être déclenchée que par un Intent Explicite
- Il est recommandé de ne pas déclarer d'Intent Filters pour les services, car cela peut causer des problèmes de sécurité

```
<activity
    android:name=".PopupActivity"
    android:label="Popup" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Actions et Données Prédéfinies d'un Intent

Intents Implicites

Voici des exemples d'actions prédéfinies communément utilisées

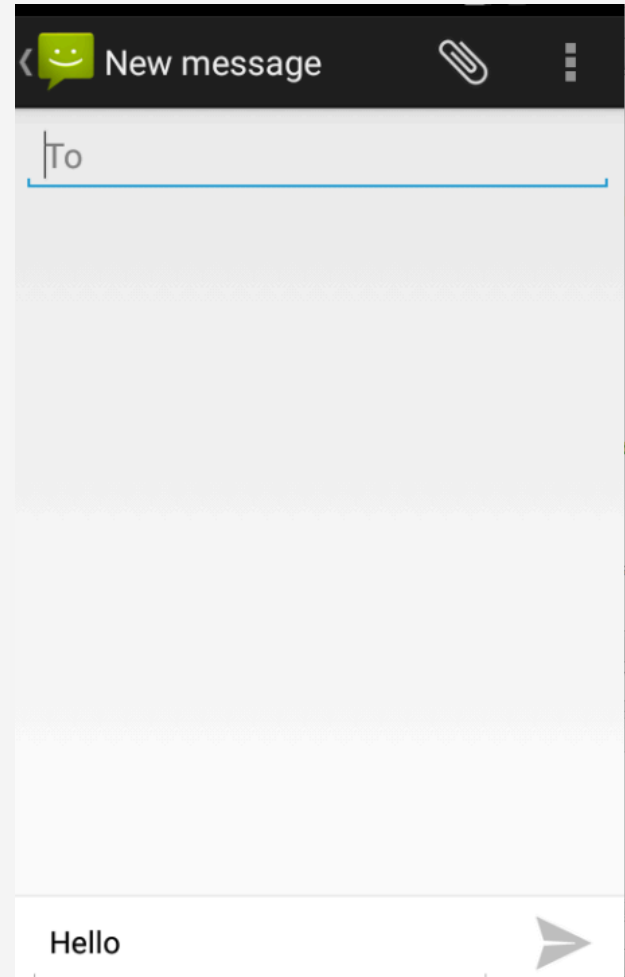
Action	Donnée	Description
ACTION_DIAL	tel:123	Affiche le numéroteur téléphonique avec le numéro (123) rempli
ACTION_VIEW	http://www.google.com	Affiche la page Google dans un navigateur.
ACTION_EDIT	content://contacts/people/2	Edite les informations sur la personne dont l'identifiant est 2 (de votre carnet d'adresse)
ACTION_VIEW	content://contacts/people/2	Utilisé pour démarrer une activité qui affiche les données du contact numéro 2
ACTION_VIEW	content://contacts/people	Affiche la liste des contacts, que l'utilisateur peut parcourir. La sélection d'un contact permet de visualiser ses détails dans un nouvel Intent.

Intent Implicite

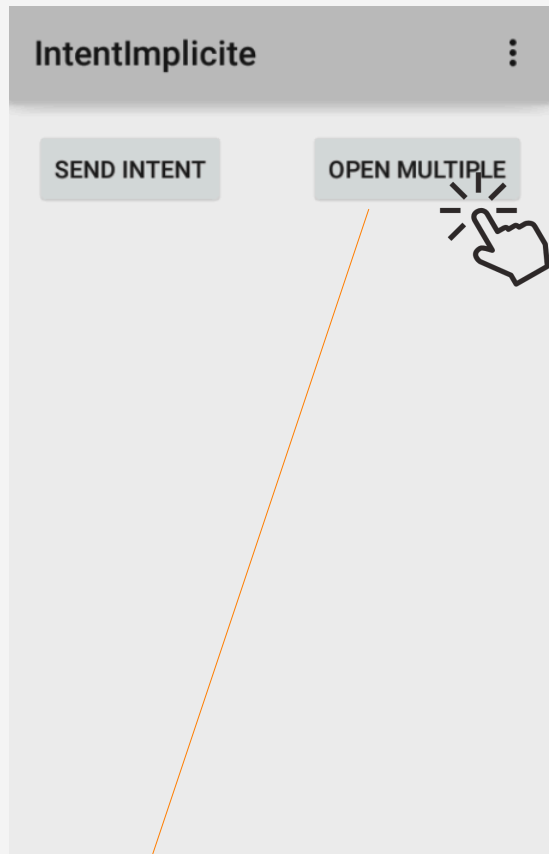
```
public void send(View v){  
    // Create the text message with a string  
    Intent sendIntent = new Intent();  
    sendIntent.setAction(Intent.ACTION_SEND);  
    sendIntent.putExtra(Intent.EXTRA_TEXT, "Hello");  
    sendIntent.setType("text/plain");  
  
    // Verify that the intent will resolve to an activity  
    if (sendIntent.resolveActivity(getPackageManager()) != null) {  
        startActivity(sendIntent);  
    }else{  
        Toast.makeText(this,"The send action could not be performed!",Toast.LENGTH_SHORT).show();  
    }  
}
```

Eviter que l'application crash si l'activité
appelée n'existe pas

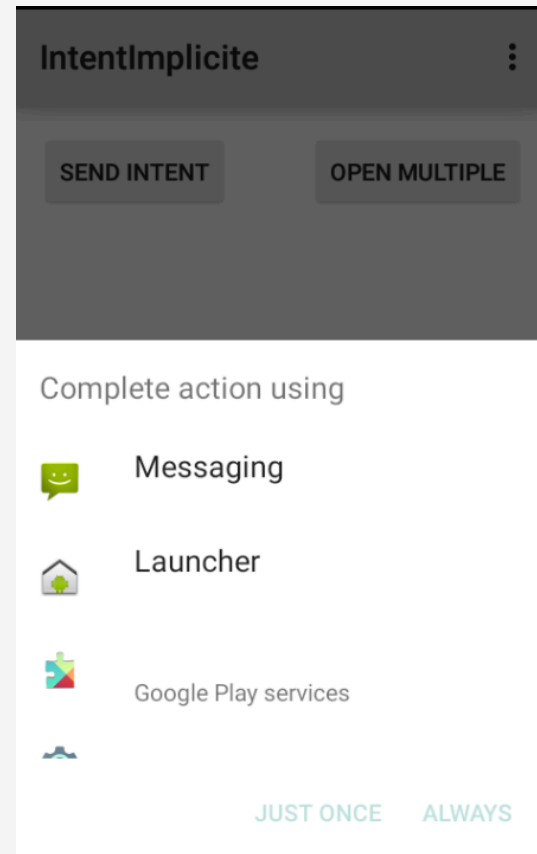
Intent Implicite



Intent Implicite: Plusieurs Activités Possibles



```
Intent sendIntent = new Intent();  
sendIntent.setAction(Intent.ACTION_MAIN);
```



Intents

INTENTS EXPLICITES

Arguments et Utilisation

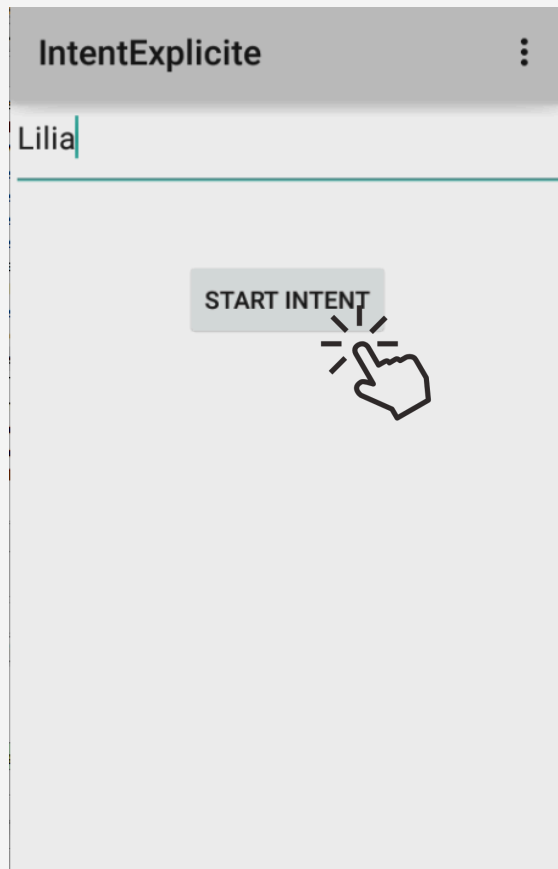
Intents Explicites

- Les principaux arguments d'un Intent explicite sont :
 - Le contexte déclenchant l'Intent (en général *this*, si on le lance à partir de l'activité de départ, ou bien *<Activity_class_name>.this*)
 - La classe destination (en général *<Activity_class_name>.class*)
- Il est typiquement appelé comme suit:

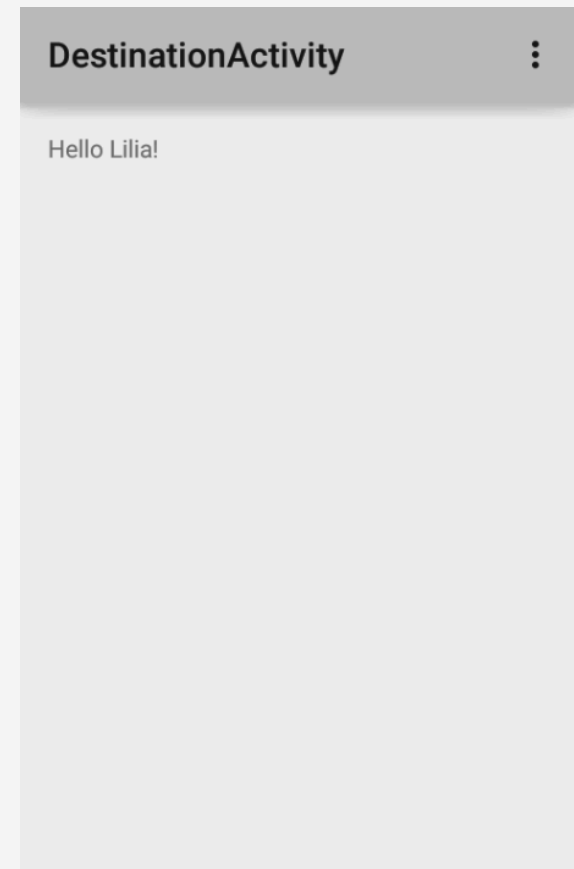
```
Intent myActivityIntent = new Intent (StartClass.this, EndClass.class) ;
startActivity (myActivityIntent) ;
```

Intent Explicite

```
public void start(View v){  
    Intent i = new Intent(this, DestinationActivity.class);  
    i.putExtra("name", name.getText().toString());  
    startActivity(i);  
}
```



```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_destination);  
  
    tv = (TextView)findViewById(R.id.show);  
  
    String name = getIntent().getStringExtra("name");  
    tv.setText("Hello " + name + "!");  
}
```



Démarrer une Activité avec Résultat

Intents Explicites

- Il est possible d'établir un lien bidirectionnel entre deux activités grâce à un Intent
- Pour recevoir un résultat à partir d'une autre activité, appeler *startActivityForResult* au lieu de *StartActivity*
- L'activité destination doit bien sûr être conçue pour renvoyer un résultat une fois l'opération réalisée
- Le résultat est envoyé sous forme d'Intent
- L'activité principale le recevra dans un *onActivityResult*

Intent avec Résultat

```
public void start(View v){
    Intent i = new Intent(this, DestinationActivity.class);
    i.putExtra("val", Integer.valueOf(texte.getText().toString()));
    startActivityForResult(i, REQUEST_CODE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {

    if (requestCode == REQUEST_CODE) {
        if (resultCode == RESULT_OK){
            Toast.makeText(this, "Resultat = "+intent.getIntExtra("resultat",0), Toast.LENGTH_LONG).show();
        }
        if (resultCode == RESULT_CANCELED) {
            Toast.makeText(this, "Pas de Resultat !", Toast.LENGTH_LONG).show();
        }
    }
}
```

Activité 1

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_destination);

    tv = (TextView) findViewById(R.id.show);
    mult = (EditText) findViewById(R.id.multEdit);

    valueReceived = getIntent().getIntExtra("val",0);
    tv.setText("Multiply "+valueReceived+" by: ");
}

public void retour (View v){
    Intent returnIntent = new Intent();
    if (mult.getText() != null) {
        int result = valueReceived * Integer.valueOf(mult.getText().toString());
        returnIntent.putExtra("resultat", result);
        setResult(RESULT_OK, returnIntent);
    } else {
        setResult(RESULT_CANCELED, returnIntent);
    }
    finish();
}
```

Activité 2

Intent avec Résultat

