


# Android 2.0, Eclair

Make your home screen just how you want it. Arrange apps and widgets across multiple screens and in folders. Stunning live wallpapers respond to your touch.



## CHAPITRE 4

### LA CLASSE INTENT ET LES PERMISSIONS

• Mr. MEGHAZI

2018-2019

Cours pour les Master II - GL

## Contenu

- La Classe « Intent »
  - Création
  - Les champs
  - Les infos contenus dans ces champs
- Lancer (Starting) des activités avec les « Intents »
  - Activation **explicite**
  - Activation **implicite** via « Intent Resolution »

## La classe « Intent »

- Une structure de données qui représente :
  - ▣ Une spécification de l'opération à effectuer, ou
  - ▣ Un évènement qui s'est produit (on veut donner une notification à d'autres composants pour cet évènement)

3

## Today's Focus

- Utilisation des « Intents » pour déterminer les opérations à effectuer, pas celles des notification des évènements.
  - ▣ Les « Intents » utilisés pour pour faire démarrer une seule activité
- A expliquer (2<sup>ème</sup> point) en utilisant les « Intents » pour la notification des évènements (broadcast receivers).

4

## La classe « Intent » - (1)

### Les « **Intents** » comme des opérations qu'on souhaite faire

- Les « Intents » fournissent un langage flexible pour spécifier les opérations à effectuer
  - **Exemple:** Choisir un contact, prendre une photo, composer un numéro-tél

5

## La Classe « Intent » (2)

1. Un « Intent » est édifié (construit) par un composant qui veut que certains travaux soient faits.
2. Reçu par une activité qui peut effectuer ces travaux

6

## Les champs de la classe « Intent »

Quels sont les types des données qui sont spécifiées dans un Intent?

- Les Champs de la classe « Intent »
  - Action
  - Data
  - Category
  - Type (MimeType)
  - Component (Cible)
  - Extras
  - Flags

7

## Intent Field – Action (1)

- Une chaîne de caractère représentant l'opération qu'on désire effectuer.
- Exemples:
  - **ACTION\_DIAL** – Composer un numéro
  - **ACTION\_EDIT** – Afficher les données à éditer
  - **ACTION\_SYNC** – Synchroniser les données de l'appareil avec le serveur
  - **ACTION\_MAIN** – Lancer une activité comme l'activité initiale d'une application

8

## Intent Field – Action (2)

- L'implémentation de l'action de l'Intent.

```
Intent newInt= new Intent(Intent.ACTION_DIAL);
```

Ou

```
Intent newInt= new Intent();
```

```
newInt.setAction(Intent.ACTION_DIAL);
```

9

## Intent Field – Data (1)

- Les données associées à l'Intent
  - Écrites (formatées) en un « Uniforme Resource Identifier » (URI)
- Exemples:
  - Données à afficher dans une carte (Map)
    - Uri.parse("geo:0,0?q=1600+Pennsylvania+Ave+Washington+DC")
  - Numéro de téléphone à composer dans le « phone-Dialer »
    - Uri.parse("tel:+15555555555")

10

## Intent Field – Data (2)

- L'injection des données dans l'Intent

```
Intent newInt= new Intent(Intent.ACTION_DIAL,
Uri.parse("tel:+15555555555"));
```

Ou

```
Intent newInt= new Intent(Intent.ACTION_DIAL);
newInt.setData(Uri.parse("tel:+15555555555"));
```

11

## Intent Field – Category

- Des informations supplémentaire sur les composants qui peuvent traiter l'Intent
- Exemples:
  - **Category\_browsable** – peut être invoquer par un navigateur pour afficher des références à des données par un URI
  - **Category\_launcher** – peut être l'activité initiale d'une tâche et être listée dans le plus haut niveau des lanceurs d'applications

12

## Intent Field – Type (1)

- Pour spécifier le MIME Type d'une donnée de l'Intent
- Exemples:
  - IMAGE/\* , IMAGE/PNG , IMAGE/JPG
  - TEXT/PLAIN , TEXT/HTML
- Si on ne va pas spécifier Android va faire des inférences et propose un type.

13

## Intent Field – Type (2)

- Déterminer le type

`Intent.setType(String type);`

Ou

`Intent.setDataAndType(Uri data, String type)`

14

## Intent Field – Component (1)

- Le composant qui doit recevoir cet Intent
- Ce champ est utilisable que si on sait qu'il y a exactement une seule activité qui doit toujours recevoir l'Intent.

15

## Intent Field – Component (2)

- Spécifier quel composant :

Intent **newInt**=

Intent(Context packageContext, Class<?> cls);

Ou

Intent **newInt**= new Intent ();

Et une des méthodes:

setComponent(), setClass(), ou setClassName()

16



## Intent Field – Extras (1)

- Des informations additionnelles associer avec l'Intent traitées sous forme de correspondances (une pair Clé - Valeur)

- **Exemple:** Intent.EXTRA\_EMAIL : récipient d'email

```
Intent newInt= new Intent(Intent.ACTION_SEND);
```

```
newInt.putExtra(android.content.Intent.EXTRA_EMAIL,
new String[]{
    "aporter@cs.umd.edu", "ceo@microsoft.com",
    "potus@whitehouse.gov","mozart@musician.org"
});
```

17

## Intent Field – Extras (2)

- Spécifier l'attribue « Extra » :
  - Plusieurs formes, ça dépend des types des données, exemple:

```
putExtra(String name, String value);
```

```
putExtra(String name, float[] value);
```

```
...
```

18

## Intent Field – Flags (1)

- Spécifie comment l'Intent doit être manipuler
- Mettre des « Flags » :

```
Intent newInt= new Intent(Intent.ACTION_SEND);  
newInt.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
```

19

## Intent Field – Flags (2)

- Exemples:

- **FLAG\_ACTIVITY\_NO\_HISTORY** – Ne met pas cette activité dans la pile d'historique
- **FLAG\_DEBUG\_LOG\_RESOLUTION** – Affiche des journaux (Logs) supplémentaires quand cet Intent est traité.

20

## Lancer des activités avec « Intent »

```
startActivity(Intent intent,...)
```

```
startActivityForResult(Intent intent, ...)
```

21

## L'activité cible

- Peut être nommé **explicitement** en l'introduisant dans le champ « Component » de l'Intent.
- Peut être déterminer **Implicitement**.
  - En se basant sur l'intent utilisé, et
  - Sur les propriétés des activités installées sur l'appareil

22

## Activation Explicite

- « **HelloWorldWithLogin** »
- Deux activités:
  - « **LoginActivity** » - vérifie le login et le mot de passe et lance ensuite «**HelloAndroidActivity**»
  - « **HelloAndroidActivity** » - affiche le message « *Hello Android* »

23

## Activation Implicite

- Quand l'activité à activer n'est pas nommée **explicitement**, Android essaye de trouver des activités qui correspondent à l'Intent.
- Ce processus s'appelle «**Intent resolution**».

24

## Le processus de résolution d'Intent (1)

- Quand l'activité à activer n'est pas nommée **explicitement**, Android essaye de trouver des activités qui correspondent à l'Intent.
- Ce processus s'appelle « **Intent resolution** ».

25

## Le processus de résolution d'Intent (2)

Dépend de deux paramètres:

- Un Intent décrivant l'opération qu'on souhaite effectuer
- Des filtres d'Intent qui vont décrire quelles opérations qu'une activité peut supporter
  - Spécifier dans le fichier «AndroidManifest.xml » ou par voie de programmation

26

## Les données de la résolution d'Intent

Une partie des données d'un Intent est ,seulement, utilisée :

- Action
- Data (URI et TYPE)
- Category

27

## Spécification des « IntentFilters » (1)

```
<activity ...>
  <intent-filter ...>
    ...
    <action android:name="actionName" />
    ...
  </intent-filter>
  ...
</activity>
```

28

## Spécification des « IntentFilters » (2)

- Prise en charge de l'Intent « `Intent.Action_DIAL` »

```
<activity ...>
  <intent-filter ...>
    ...
    <action android:name="android.intent.action.DIAL" />
    ...
  </intent-filter>
  ...
</activity>
```

Chaine de caractères



- Ce qui correspond en **JAVA** à : « `Intent.ACTION_DIAL` »

29

## Ajouter des données à l'IntentFilter

```
<intent-filter ...>
  ...
  <data
    android:mimeType="string"
    android:scheme="string"
    android:host="string"
    android:port="string"
    android:path="string"
    android:pathPattern="string"
    android:pathPrefix="string"
  />
  ...
</intent-filter>
```

<http://developer.android.com/guide/components/intents-filters.html>

30

## Prise en charge d'un schéma d'Intent

```
<intent-filter ...>
  ...
  <data android:scheme="geo" />
  ...
</intent-filter>
```

31

## Ajouter une catégorie dans le filtre

```
<intent-filter ...>
  ...
  <category android:name="string" />
  ...
</intent-filter>
```

32



## Exemple: l'application « Maps »

```
<intent-filter >
  <action android:name = "android.intent.action.VIEW" />
  <category android:name = "android.intent.category.DEFAULT" />
  <category android:name =
    "android.intent.category.BROWSABLE"/>
  <data android:scheme = "geo"/>
</intent-filter>
```

33

## Recevoir des Intents « Implicites »

### Remarque:

- A fin de recevoir des Intents implicites une activité doit spécifier un filtre d'Intent avec la catégorie:

« android.intent.category.DEFAULT »

34

## Les priorités

- **android: priority** – transmise au composant père à la prise en charge du processus faisant la correspondance entre les Intents.
- Ce qui entraîne que Android préfère une activité à une autre
- La valeur doit être entre **-10000** et **10000**
- Des valeurs plus grandes représentent des priorités plus grandes

35

## Examiner les filtres d'Intents

```
% adb shell dumpsys package
```

36

# Les permissions

37

## Contenu

- Les permissions dans Android
- Définir et utiliser les permissions d'une Application
- Les permissions des composants et les permissions relatives aux APIs

38

## Les permissions (1)

- Android protège les ressources et les données avec les permissions.
- Utilisées pour limiter l'accès à:
  - Informations de l'User - **Ex:** Contactes
  - Cost-Sensitive APIs - **Ex:** SMS/MMS
  - Ressources Système - **Ex:** Camera

39

## Les permissions (2)

- Les permissions sont représentées en tant que chaînes de caractères [Strings]
- Dans «AndroidManifest.xml », les applications déclarent les permissions:
  - Qu'elles utilisent elle-même
  - Requises par autres composants (utilisant l'Appli).

40

## Utilisation des permissions (1)

- Les applications spécifient les permissions qu'elles utilisent à travers un « Tag » : **<uses-permission>**
- Les utilisateurs doivent accepter ces permissions avant qu'une application soit installée
  - Sinon, des erreurs et des messages d'échecs vont apparaître

41

## Utilisation des permissions (2)

```
<manifest ... >
...
<uses-permission android:name= "android.permission.CAMERA"/>
<uses-permission android:name= "android.permission.INTERNET"/>
<uses-permission
  android:name="android.permission.ACCESS_FINE_LOCATION"/>
...
</manifest >
```

*A voir dans le site officiel d'android sous la section: « Manifest.permission »*

42

## Exemple

### MapLocationFromContacts

- Sélectionner un contact de la base de données des contacts
- Afficher une carte pointant sur l'adresse du contact choisi

43

## Définir Les permissions (1)

Les applications peuvent aussi **définir** et **renforcer** leurs propres **permissions**

- Supposer que votre application effectue une opération privilégié/dangereuse
- Peut être que vous ne voulez pas autoriser n'importe quelle application d'invoquer certaines opérations
- **Alors** vous devez définir et renforcer vos propres permissions

44

## Définir Les permissions (2)

### Exemple:

- « **PermissionExempleBoom** »
- Une simple Application qui effectue (prétend d'effectuer) une **action dangereuse**

45

## Définir et renforcer les permissions

- Si vous ne voulez pas que n'importe qui exécute «**PermissionExempleBoom** »
- Définir et renforcer une permission spécifique à l'application.

46

## Utiliser la nouvelle permission

Les applications qui veulent utiliser « **PermissionExempleBoom** »  
doivent, maintenant, acquérir la bonne permission.

47

## Utilisation des permissions

- Une application peut **DÉCLARER/ACCEPTER** les permissions d'une application qu'elle utilise.

48



## Les permissions des composants

- Les composants individuels peut eux aussi définir leurs propres permissions, mettant, ainsi, une restriction aux composants qui peuvent leurs accéder.
- Les permissions des composants prennent le niveau des permissions des applications qu'ils les invoquent.

49

## 1- Les permissions des Activités

- Met une restriction sur les composants qui peuvent démarrer l'activité associé.
- Vérifié lors de l'exécution de:
  - `startActivity()`
  - `startActivityForResult()`
- Throws `SecurityException` (lors d'un échec de permission)

50

## 2- Les permissions des services

- Met une restriction sur les composants qui peuvent démarrer ou connecter au service associé.
- Vérifié lors de l'exécution de:
  - `Context.startService()`
  - `Context.stopService()`
  - `Context.bindService()`
- Throws `SecurityException` (lors d'un échec de permission)

51

## 3- Les permissions des « BroadcastReceivers »

- Met une restriction sur les composants qui peuvent envoyer et recevoir des « Broadcasts »
- Vérifié à plusieurs endroits
  - A Voir dans le chapitre des « BroadcastReceivers ».

52

## 4- Les permissions des « ContentProviders »

- Met une restriction sur les composants qui peuvent lire et écrire dans le « ContentProvider »
  - A Voir dans le chapitres des « ContentProviders ».