



Cours 5 : fournisseurs de contenu

Hadjila Fethallah

Maître de Conférences au
Département d'Informatique

F_hadjila@mail.univ-lemcen.dz

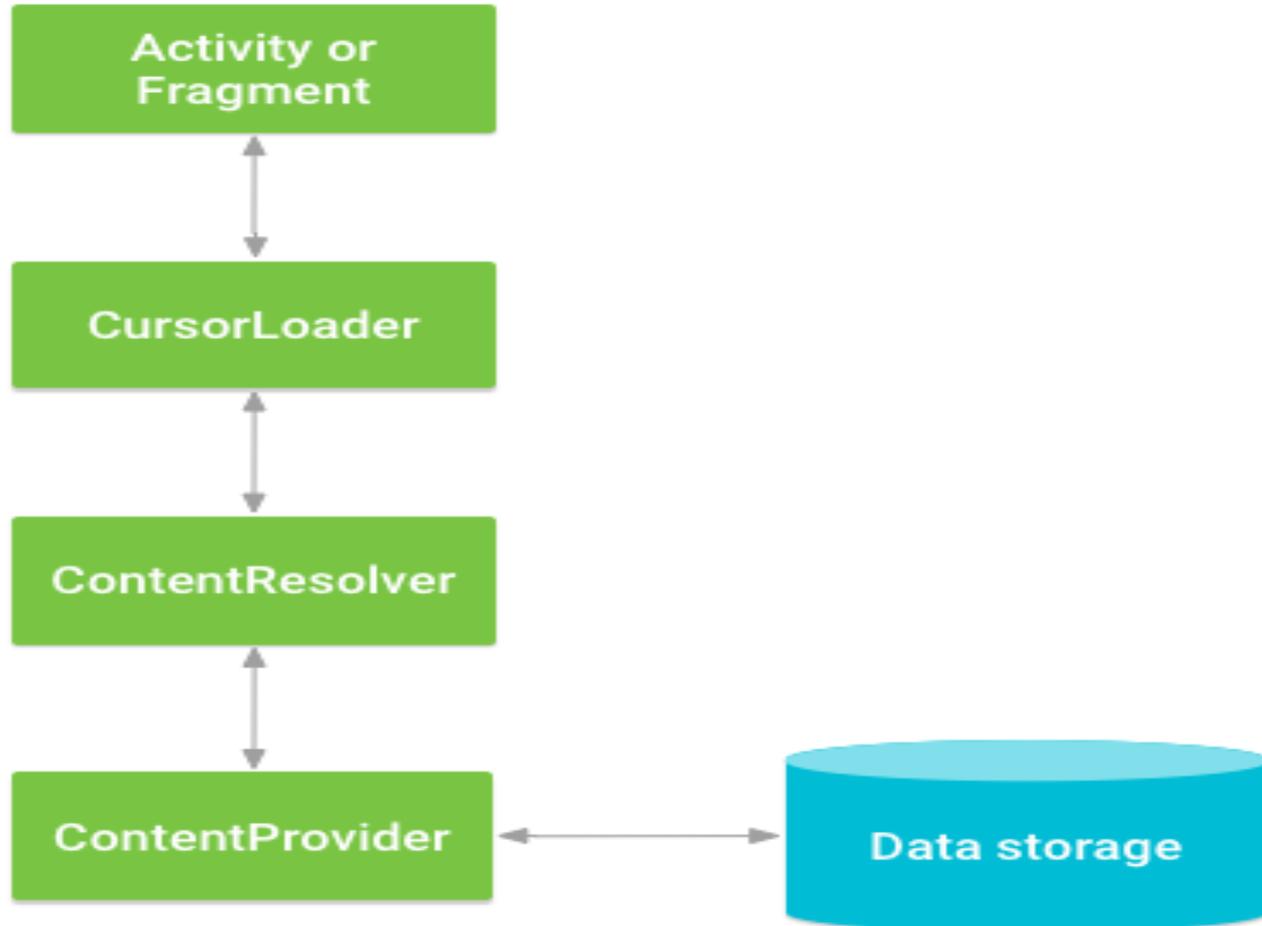
Fournisseur de contenu (CP)

- C'est un composant d'une application qui n'a pas d'interface utilisateur UI
- Il encapsule et partage des données structurées
- Il constitue l'approche officielle d'android pour partager les données entre processus
- Encapsuler les données= offrir une interface standard pour
 - insérer, modifier, rechercher, supprimer les données
- La méthode de stockage des données est en dehors de la responsabilité du fournisseur de contenu.
- Le concepteur de l'application doit choisir sa propre technologie pour stocker les données de manière permanente (SGBD SQLITE, fichiers XML,...)

Modèle de données

- Les données sont exposées sous forme de **tables (similaires aux BDD relationnelles)**
- Un CP peut gérer plusieurs tables
- Chaque table contient une colonne **_ID** qui assure un identifiant unique de chaque enregistrement
- Quelques tables prédefinies :
 - UserDictionary.Words: ayant comme colonnes: id,word,frequency...
 - Contacts: ayant comme colonnes: id,display_name,photo...
 - CallLog.Calls

L'accès aux fournisseurs de contenu



Classes impliquées

- package: android.content
- **classe abstraite ContentProvider**
 - Encapsule les données
- **classe abstraite ContentResolver**
 - C'est une classe intermediaire entre le composant client et le CP (elle appartient au processus client)
- **classe Uri du package: android.net**
 - Fournit un moyen d'identification des tables et des lignes
- **Classe contrat**

URI

- **Uniform Resource Identifier**

- Décrite dans la norme RFC 2396

- **Syntaxe:**

- Nom du Schema** : vaut “content “ pour les CP
- “://”
- Authorité** : dans ce cas c’est l’identifiant du CP
- Chemin (Path)**: chemin ou adresse de la table
- ID** : c’est un parametre optionnel qui designe le numero de la ligne dans une table

Exemple d'uri (1)

- Tous les URIs des CP commencent par content://
- URI d'un CP:
 - content:// com.example.cpetudiant
- URI d'une table (appelée etudiant):
 - content:// com.example.cpetudiant/etudiant
- URI d'une seule ligne (ligne avec _ID=24):
 - content:// com.example.cpetudiant/etudiant/24

Exemple d'uri (2)

- `content://com.android.contacts/contacts`
 - représente une liste d'instances de contacts
- `content://com.android.contacts/contacts/5`
 - représente le contact dont l'identifiant est 5.

Implementation d'un CP

- Étendre la classe `ContentProvider` impose l'implémentation des six méthodes suivantes :
- `onCreate`
- `getType`
- `query`
- `insert`
- `update`
- `delete`
- le système android s'en charge d'instancier le CP (et pas l'utilisateur)

Implementation d'un CP

- Hériter de la classe `ContentProvider`
- Ajouter le composant crée dans le fichier manifest :

```
<manifest ... >
```

```
...
```

```
<application ... >
```

```
<provider
```

```
android:name="com.example.cpetudiant.etudiantprovider"
```

```
android:authorities="com.example.cpetudiant"
```

```
android:exported="true"/>
```

```
...
```

```
</application>
```

```
</manifest>
```

Classe contentprovider

- **boolean onCreate()**
- Initialise la BDD du CP
- Appelée lorsque le premier composant client sollicite le CP
- **String getType(Uri uri)**
- Retourne le type MIME d'une table ou d'une ligne (tuple), la table ou le tuple sont identifiés avec l'uri.

Classe contentprovider

- `int update(Uri uri, ContentValues values, String selection, String[] selectionArgs)`
- Met à jours toutes les lignes qui répondent aux critères de matching (pour une table ou un tuple identifié par l'argument uri)
- Les nouvelles valeurs sont placées dans la structure values.
- La fonction retourne le nombre des lignes modifiées

Classe contentprovider

- **Uri insert(Uri uri, ContentValues values)**
- Elle insere une nouvelle ligne dans la table identifiée par l'argument uri.
- Elle retourne l'URI de la nouvelle ligne inserée
- **int delete(Uri uri, String selection,**
- **String[] selectionArgs)**
- Supprime toutes les lignes qui repondent aux criteres de selection (pour une table ou un tuple identifié par l'argument uri).
- Retourne le nombre de lignes supprimées

Classe contentprovider

- **Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder)**
- Exécute une requête sur les lignes qui répondent aux critères de sélection (la table ou le tuple est déjà identifié par l'argument uri)
- La projection contient une liste de colonnes qui composent le résultat.
- le résultat de la requête est retourné sous forme d'un objet "cursor"

ContentResolver

- Les requetes du composant client ne sont pas executées directement par le CP mais acheminées tout d'abord à un objet ContentResolver
- L'objet ContentResolvers est crée grace à la fonction **getContentResolver()** (à partir d'une activité ou un autre composant)
- La classe ContentResolver contient les memes methodes que ContentProvider: insert(), update(), delete(), query(), ...
- Elle permet aussi de notifier les observateurs enregistrés lorsque les données du CP changent

Classe statique (ou la classe de contrat)

- Elle définit l'uri de l'autorité
- Pour chaque table du CP , elle englobe une sousclasse qui définit:
 - les noms des colonnes
 - son uri (ie l'uri de la table en cours)
 - son type mime
 - le type mime de ses tuples

Type mime (suite à classe de contrat)

- La syntaxe générale d'un type mime est:
- " type/soustype"
- dans le cas d'une table:
"vnd.android.cursor.**dir**/vnd.cpetudiant.con
tratetudiant_etudiant"
- Dans le cas d'une ligne(tuple)
- "vnd.android.cursor.**item**/vnd.cpetudiant.c
ontratetudiant_etudiant"

Remarques

- La classe CursorLoader permet l'exécution des requêtes dans un thread worker (séparé du main thread).
- Oncreate est appelée une seule fois lors de l'instanciation du CP, les autres fonctions sont appelées potentiellement plusieurs fois.
- Les fonctions d'un CP doivent être synchronisées pour garder la cohérence de la BDD.
- Ex: **public synchronized String getType(Uri uri)**



Exemple détaillé

- Voir le TP7



FIN du Cours5