



Préférences partagées sous android



Plan

I.	INTRODUCTION	1
II.	LES PREFERENCES PARTAGEES	2
II.1.	SHARED PREFERENCES	2
II.2.	EDITOR	2
II.3.	EXEMPLE D'ECRITURE	2
II.4.	EXEMPLE DE LECTURE	2

I. Introduction

En programmation, la gestion de la persistance des données (en anglais : persistence) et parfois des états d'un programme réfère au mécanisme responsable de la sauvegarde et de la restauration des données. Ces mécanismes font en sorte qu'un programme puisse se terminer sans que ses données et son état d'exécution ne soient perdus. Ces informations de reprise peuvent être enregistrées sur disque, éventuellement sur un serveur distant (un serveur de bases de données relationnelles, par exemple). [1]

La persistance des données peut être locale ou distante. La persistance de données locale consiste à stocker les données dans la mémoire interne du terminal mobile. La persistance de données distante consiste à stocker les données dans une base de données externe au terminal mobile. Android ne peut pas communiquer directement avec la BD, cependant plusieurs solutions sont possibles pour manipuler les données de la BD.

Parmi les solutions de persistance locale :

- L'utilisation des préférences partagées,
- L'utilisation des fichiers,
- L'utilisation de SQLite.

Parmi les solutions de persistance distante :

- L'utilisation d'une application web qui appelle le SGBD distant,
- L'utilisation d'une application hybride qui appelle une application web qui appelle le SGBD distant,
- L'utilisation d'une application native qui appelle une application web qui appelle le SGBD distant,
- L'utilisation d'une application hybride qui appelle un service web qui appelle le SGBD distant.

Le choix de la solution à utiliser dépend de la nature des données à stocker, de leur quantité, du nombre d'utilisateur des données et de la disponibilité d'une connexion Internet :

- Si les données ne sont pas structurées alors on utilise les préférences partagées ou les fichiers.
- Si les données sont structurées et leur quantité est importante alors on utilise un SGBD.
- Si seulement le propriétaire du terminal mobile utilise les données alors on utilise une persistance locale sinon on fait recours à une persistance distante.
- Si aucune connexion internet n'est existante alors on utilise une persistance locale.
- Si la quantité de données est très importante alors on utilise une persistance distante pour ne pas saturer la mémoire du terminal mobile.

Exemples :

- Enregistrement des paramètres de connexion.
- Médecin qui se déplace à ces patients et enregistre un ensemble de données de diagnostic.
- Saisie et consultation des notes des étudiants.
- Enregistrement de la variation de l'accélération en utilisant l'accéléromètre.

II. Les préférences partagées

II.1. SharedPreferences

SharedPreferences est une classe du package android.content, elle permet de gérer les préférences partagées sous forme d'un fichier xml qui contient un ensemble de paires clé/valeur.

Pour initialiser un SharedPreferences on peut utiliser l'une de ces deux solutions:

- `SharedPreferences p=PreferenceManager.getDefaultSharedPreferences (getApplicationContext());`
- `SharedPreferences p=getSharedPreferences("conn", Context.MODE);`

Dans la première solution, le fichier crée aura un nom qui dépend du package de l'application et son chemin complet est : `"/data/data/[application package name]/shared_prefs/[application package name]_preferences.xml"`, par contre dans la deuxième solution, le nom du fichier sera `conn.xml` et son chemin complet est : `"/data/data/[application package name]/shared_prefs/conn.xml"`.

Pour la deuxième solution le paramètre `MODE` peut être [2] :

- `MODE_PRIVATE` : le mode par défaut, seul l'application qui a créé les préférences peut y accéder.
- `MODE_APPEND` : permet d'ouvrir les préférences partagées en mode ajout.

Pour lire une valeur à partir des préférences partagées, utiliser une de ces méthodes suivant le type de la valeur :

- `boolean getBoolean(String key, boolean defValue)`
- `float getFloat(String key, float defValue)`
- `int getInt(String key, int defValue)`
- `long getLong(String key, long defValue)`
- `String getString(String key, String defValue)`

Pour ajouter une paire clé/valeur il faut passer par la classe `Editor` :

- `SharedPreferences.Editor edit()`

II.2. Editor

`Editor` est une interface interne de la classe `SharedPreferences`, elle permet d'ajouter des paires clé/valeur aux préférences partagées.

Pour ajouter une valeur aux préférences partagées, utiliser une de ces méthodes suivant le type de la valeur :

- `Editor putBoolean(String key, boolean value)`
- `Editor putFloat(String key, float value)`
- `Editor putInt(String key, int value)`
- `Editor putLong(String key, long value)`
- `Editor putString(String key, String value)`

Pour effacer tous les paires utiliser :

- `Editor clear()`

Pour valider le changement des valeurs utiliser :

- `Boolean commit()`

II.3. Exemple d'écriture

```
SharedPreferences p=getSharedPreferences("conn", Context.MODE_PRIVATE);
Editor ed= p.edit();
ed.putString("utilisateur", "Mohamed");
ed.commit();
```

II.4. Exemple de lecture

```
SharedPreferences p=getSharedPreferences("conn", Context.MODE_PRIVATE);
String utilisateur = p.getString("utilisateur", "");
```

Références

[1] [https://fr.wikipedia.org/wiki/Persistance_\(informatique\)](https://fr.wikipedia.org/wiki/Persistance_(informatique))

[2] <http://androidopentutorials.com/android-sharedpreferences-tutorial-and-example/>