

# Shared Preferences

Tutoriel Android



# Introduction

Dans toutes applications, il est souvent nécessaire d'utiliser des variables qui doivent être gardé en mémoire même suite à une fermeture.

## **Une solution : les Shared Preferences**

# Objectifs du tp

- Comprendre le principe des “shared preferences”.
  - Connaître les différentes méthodes et fonctions.
  - Analyse d'un exemple d'utilisation.
- Savoir stocker et récupérer des données avec cette méthode.
  - Exercice d'application.
- Découvrir les “preferences” et le lien avec les “shared preferences”.
  - Démonstration d'un exemple de settings.

# Shared Preferences ???

- C'est un moyen de stocker et de récupérer des données.
- Les données sont stockées sous la forme : clé -> valeur.
- Données stockées dans des fichiers XML.

nameKey	Jean
emailKey	jean.dupont@gmail.com
pseudoKey	JeanD

*Exemple de données stockées en Shared Preferences*

# Accéder à des préférences dans un fichier

- Avec la méthode `getSharedPreferences(String name, int mode)`.
  - **name** : nom du fichier;
  - **mode** : mode utilisé, `MODE_PRIVATE` par défaut.
- Retourne une instance de `SharedPreferences`.

```
SharedPreferences sharedPreferences = getSharedPreferences("MyFile", Context.MODE_PRIVATE);
```

Code pour accéder à des préférences dans un fichier

# Récupérer la valeur d'une préférence

- Plusieurs méthodes disponibles selon le type de la donnée :

```
Map<String, ?> getAll();
```

```
String getString(String key, String defValue);
```

```
boolean getBoolean(String key, boolean defValue);
```

```
int getInt(String key, int defValue);
```

```
float getFloat(String key, float defValue);
```

```
long getLong(String key, long defValue);
```

```
SharedPreferences sharedPreferences =  
    getSharedPreferences("MyFile", Context.MODE_PRIVATE);  
  
String name = sharedPreferences.getString("nameKey", "Anonyme");  
int age = sharedPreferences.getInt("ageKey", "0");
```

*Exemple de récupération de valeurs dans un fichier de préférences*

# Stocker une valeur ou la modifier

1. Appeler la méthode `edit()` sur l'objet ***SharedPreferences***.
  - Retourne une instance de l'interface ***SharedPreferences.Editor***.
2. Placer la ou les valeurs à l'aide des méthodes suivantes selon le type :
  - `putString()`, `putInt()`, `putBoolean()`, `putFloat()`... (prennent toutes en paramètres la clé et la valeur).
  - `clear()`: indique la suppression de toutes les valeurs .
  - `remove(String key)` : supprime l'entrée qu'après avoir fait appel à la méthode `commit()`.
3. Appeler la méthode `apply()` pour ajouter ces changements.
  - `apply()` écrit de manière asynchrone. Il est possible d'utiliser `commit()` mais cette méthode est déconseillée car elle écrit de manière synchrone.

# Stocker une valeur ou la modifier - suite

```
SharedPreferences sharedPreferences = getSharedPreferences("MyFile", Context.MODE_PRIVATE);  
// Appel de la méthode edit()  
SharedPreferences.Editor editor = sharedPreferences.edit();  
// Création ou modification de valeurs  
editor.putString("name", "Jean");  
editor.putInt("age", 42);  
// Application des changements  
editor.apply();
```

*Exemple d'ajouts/modifications de valeurs dans un fichier de préférences*



# Exemple d'utilisation

Accès au fichier de préférence  
*MyPrefsFile*

Récupération de la valeur  
*silentMode*

Enregistrement de la valeur  
*silentMode*

- 1 . Appel de la méthode *edit()*.
- 2 . On place le contenu de *mSilentMode* dans la valeur *silentMode*.
- 3 . Ajout des changements dans le fichier de préférence.

```
public class Calc extends Activity {  
    public static final String PREFS_NAME = "MyPrefsFile";
```

```
    @Override  
    protected void onCreate(Bundle state) {  
        super.onCreate(state);  
        ...
```

```
        // Restore preferences
```

```
        SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);  
        boolean silent = settings.getBoolean("silentMode", false);  
        setSilent(silent);  
    }
```

```
    @Override  
    protected void onStop() {  
        super.onStop();
```

```
        // We need an Editor object to make preference changes.
```

```
        // All objects are from android.context.Context
```

```
        SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);  
1 → SharedPreferences.Editor editor = settings.edit();  
2 → editor.putBoolean("silentMode", mSilentMode);
```

```
        // Commit the edits!
```

```
3 → editor.commit();  
    }
```

# Exercice d'application

Sauvegarde d'informations de  
préférences (pseudo, volume).

The screenshot shows an Android application interface with a blue header bar containing the title "MyTpSharedPreferences". The status bar at the top right shows "4G", signal strength, battery, and the time "1:22". The main content area is white and contains two sections. The first section is labeled "Pseudo :" and has a text input field with the placeholder text "Pseudo". The second section is labeled "Volume :" and contains a slider control. Below the slider is a toggle switch labeled "Actif". At the bottom of the form is a grey button with the text "ENREGISTRER".

# Les widgets

- TextView (*id : txt\_pseudo*)

Pseudo

---

- Button (*id : btn\_save*)

ENREGISTRER

- Switch (*id : sw\_volume*)

Actif :



- SeekBar (*id : sb\_volume*)



À vous de jouer ...  
( ~5 mins | Voir le premier exercice du pdf )

1. Récupérer les valeurs dans le fichier de préférence PREFS\_NAME.
2. Placer les valeurs dans les différents widgets associés.

```
private void initPrefs(SharedPreferences prefs) {  
    // TODO 1  
    String pseudo = prefs.getString("pseudo", "");  
    boolean volume_mode = prefs.getBoolean("volume_actif", true);  
    int volume_val = prefs.getInt("volume_value", 5);  
  
    // TODO 2  
    this.pseudo.setText(pseudo);  
    this.volume_actif.setChecked(volume_mode);  
    this.volume_value.setProgress(volume_val);  
}
```

Enregistrer les valeurs dans le fichier de préférence PREFS\_NAME.

```
private void onSavePrefs() {
    this.btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            SharedPreferences prefs = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);
            SharedPreferences.Editor editor = prefs.edit();

            // On place les valeurs des widgets
            editor.putString("pseudo", pseudo.getText().toString());
            editor.putBoolean("volume_actif", volume_actif.isChecked());
            editor.putInt("volume_value", volume_value.getProgress());
            // On stocke les valeurs dans le fichier de préférences
            editor.apply();
            Toast.makeText(MainActivity.this,
                "Preferences saved !",
                Toast.LENGTH_LONG).show();
        }
    });
}
```

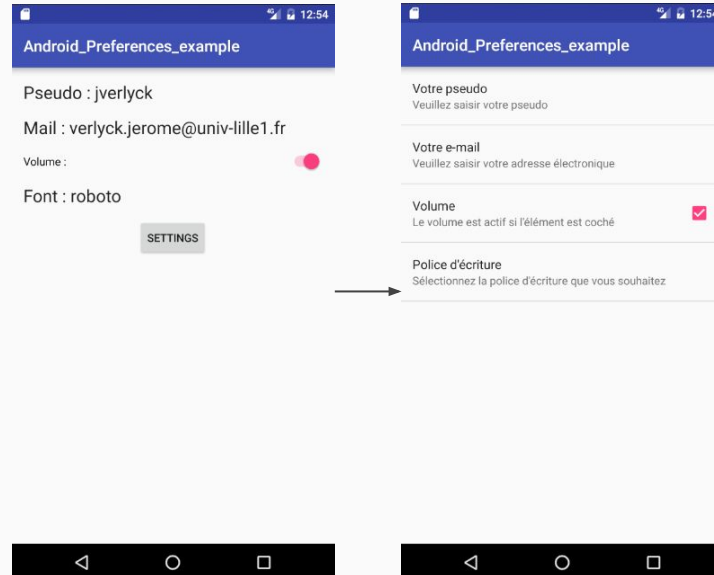
# Code final

<https://drive.google.com/open?id=0B7N3r33cPwHuZzZUTnhHNjBCc0E>

# Shared Preferences <--> Preferences

## Preferences :

APIs permettant la construction d'une interface pour la gestion des paramètres d'expériences utilisateur de notre application ou d'autres applications si on le souhaite.





# Utilisation d'un fichier .xml dans un dossier *res/xml*

Lors d'un clique :

Affiche une zone de saisie

Coche ou décoche la case

Affiche une liste de choix sous forme de radio boutons

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
    <EditTextPreference android:title="Votre pseudo"
        android:key="pseudo"
        android:summary="Veuillez saisir votre pseudo">
    </EditTextPreference>
    <EditTextPreference android:title="Votre e-mail"
        android:key="mail"
        android:summary="Veuillez saisir votre adresse électronique">
    </EditTextPreference>
    <CheckBoxPreference android:title="Volume"
        android:defaultValue="true"
        android:summary="Le volume est actif si l'élément est coché"
        android:key="volume" />
    <ListPreference android:title="Police d'écriture"
        android:summary="Sélectionnez la police d'écriture que vous souhaitez"
        android:key="font"
        android:defaultValue="arial"
        android:entries="@array/listArray"
        android:entryValues="@array/listValues" />
</PreferenceScreen>
```

# L'exemple complet

<https://drive.google.com/open?id=0B7N3r33cPwHuQS11Z1lwZ2JRVk0> (version déprécié depuis API level 11)

<https://drive.google.com/open?id=0B7N3r33cPwHuZGZSNzljVjRrRzA> (utilisation fragment => à essayer plus tard)

# Des questions ?

[jerome.verlyck@etudiant.univ-lille1.fr](mailto:jerome.verlyck@etudiant.univ-lille1.fr)

# Liens utiles

- Lien de la documentation sur l'interface SharedPreferences :
  - <https://developer.android.com/reference/android/content/SharedPreferences.html>
- Lien de la documentation sur l'objet Preference
  - <https://developer.android.com/reference/android/preference/Preference.html>
- Lien du guide officiel sur les Settings
  - <https://developer.android.com/guide/topics/ui/settings.html>