android

An Open Handset Alliance Project

Persistance des données

O.Legrand
G. Seront

Persistance

- Plusieurs moyens sur le mobile:
  - Système de fichiers local;
  - Bases de données SQLite;
  - *Shared Preferences* pour sauver les préférences de l'utilisateur;
  - *Activity.onSaveInstanceState()* pour sauver les données d'une instance d'activité.
- Stockage externe :
  - sur un serveur de l'entreprise, dans le cloud,…

developer.android.com/guide/topics/data/data-storage.htmll

Fichiers en local

- Utilisation du système de fichiers local pour sauver les données;
- En utilisant :
  - les classes et méthodes Java I/O standard;
  - ou *Context.openFileInput()* et *Context.openFileOutput()* disponibles sous Android;

developer.android.com/reference/android/content/Context.html

```
public class Android_TestFichier extends Activity {

    private static final String NOM_FICHIER = "mesDonnees.tmp";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        FileOutputStream fos=null;
        FileInputStream fis=null;
        try {

            // écriture dans un nouveau fichier
            fos = openFileOutput(NOM_FICHIER, Context.MODE_PRIVATE);
            for (byte i = 0; i < 10; i++) {
                fos.write(i);
            }
            fos.close();

            // lecture du fichier
            fis = openFileInput(NOM_FICHIER);
            for (byte i = 0; i < 10; i++) {
                String s = "" + fis.read();
                Log.i("MonLog", s);
            }
            fis.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Fichiers en local

- Remarques :
  - Accès uniquement aux fichiers contenus dans le dossier de l'application courante;
  - *MODE_PRIVATE* : accès réservé à l'application;
  - Pour un accès ouvert à toutes les applications :

    MODE_WORLD_READABLE *ou* *MODE_WORLD_WRITEABLE*;
  - ou *Content Provider* (méthode standard).

developer.android.com/reference/android/content/Context.html

Shared Preferences

- Les *Shared Preferences* permettent de sauver d'une session à l'autre :
  - les préférences des utilisateurs;
  - les paramètres de l'application;
  - l'état de l'interface utilisateur;
- Par défaut, elles ne sont partagées que par les composants de l'application *(MODE_PRIVATE)*;
- Elles seront accessibles par les composants des autres applications si : *MODE_WORLD_READABLE* ou *MODE_WORLD_WRITEABLE* ou *MODE_MULTI_PROCESS.*

developer.android.com/guide/topics/data/data-storage.html#pref

Sauvegarder les préférences

- Pour créer ou modifier une *Shared Preferences* :
  - appelez la méthode *getSharedPreferences()* sur le context en lui passant le nom de la préférence en paramètre;
  - demandez la référence de l'éditeur : *edit()* ;
  - appelez les méthodes *put<type>* sur l'éditeur pour ajouter les données (clef-valeur);
  - exécutez un *commit()* pour sauvegarder.

developer.android.com/guide/topics/data/data-storage.html#pref

Sauvegarder les préférences

- Exemple :

```
private void sauverLesPreferences() {
    // obtenir la référence des préférences partagées
    SharedPreferences preferences = this.getSharedPreferences(
            "mesPreferences", MODE_PRIVATE);

    // récupère un éditeur pour modifier les préférences
    Editor editor = preferences.edit();

    // modifie les préférences
    editor.putBoolean("unBooléen", true);
    editor.putFloat("unRéel", 4.5f);
    editor.putInt("unEntier", 8);
    editor.putString("uneChaine", "Salut toi!");

    // enregistre les préférences
    editor.commit();
}
```

developer.android.com/guide/topics/data/data-storage.html#pref

Récupérer les préférences

- Pour récupérer les préférences sauvegardées :
  - appelez la méthode *getSharedPreferences()* sur le context en lui passant le nom de la préférence en paramètre;
  - appelez les méthodes *get<type>* pour extraire les valeurs sauvegardées;
  - Passez aux getters 2 paramètres :
    - une clef et une valeur par défaut

developer.android.com/guide/topics/data/data-storage.html#pref

Récupérer les préférences

- Exemple :

```
private void chargerLesPreferences() {
    // obtenir la référence des préférences partagées
    SharedPreferences preferences = this.getSharedPreferences(
            "mesPreferences", MODE_PRIVATE);

    // récupère les valeurs sauvegardées
    boolean unBooléen = preferences.getBoolean("unBooléen", false);
    float unRéel = preferences.getFloat("unRéel", 0);
    int unEntier = preferences.getInt("unEntier", 0);
    String uneChaine = preferences.getString("uneChaine", "vide");

    Log.i("MonLog", "" + unBooléen + "   " + unRéel + "   " + unEntier
            + "  " + uneChaine);
}
```

developer.android.com/guide/topics/data/data-storage.html#pref

Préférences d'une activité

- Toute activité possède sa propre *Shared Preferences* unique qui porte son nom;
- Par défaut, elle ne la partage qu'avec les autres composants de l'application *(MODE_PRIVATE)*;
- Elle sera accessible par les composants des autres applications si : *MODE_WORLD_READABLE ou MODE_WORLD_WRITEABLE ou MODE_MULTI_PROCESS.*
- Pour y accéder : *getPreferences() ;*
- Elle s'utilise comme toute préférence partagée.

developer.android.com/guide/topics/data/data-storage.html#pref

Préférences d'une activité

```
private void sauverLesPreferencesDeActivité() {
    // obtenir la référence des préférences de l'activité
    SharedPreferences preferences = this.getPreferences(MODE_PRIVATE);

    // récupère un éditeur pour modifier les préférences
    Editor editor = preferences.edit();

    // modifie les préférences
    editor.putString("uneChaine", "préférence de l'activité");

    // enregistre les préférences
    editor.commit();
}

private void chargerLesPreferencesDeActivité() {
    // obtenir la référence des préférences de l'activité
    SharedPreferences preferences = this.getPreferences(MODE_PRIVATE);

    // récupère la valeur sauvegardée
    String uneChaine = preferences.getString("uneChaine", "vide");

    Log.i("MonLog", "" + uneChaine);
}
```

developer.android.com/guide/topics/data/data-storage.html#pref

Sauver l'état d'une activité

- Pour sauver l'état d'une activité, *onSaveInstanceState()* est appelée par le système:
  - avant *onStop()*, parfois avant, parfois après *onPause()*;
  - elle sauve le contenu des vues identifiées *(@+id/);*
  - ces contenus seront restaurés au redémarrage de l'activité par un appel automatique à *onRestoreInstanceState()*

developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)

onSaveInstanceState()

- Cette méthode sera appelée si :
  - l'activité est tuée par le système;
  - la configuration a changé :
    - l'écran change de position (portrait-paysage);
    - le clavier physique (s'il existe) est activé,…
- Elle ne sera pas appelée lors :
  - d'un appui par l'utilisateur sur le bouton de retour;
  - d'un appel à la méthode *finish()*.

developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)

onSaveInstanceState()

- Cette méthode peut être redéfinie pour sauver d'autres variables d'instance de l'activité;
- Sauvez les données dans le *Bundle* passé en paramètre;
- Utilisez les méthodes *put<type>* comme pour les *Shared Preferences*.

```
@Override
protected void onSaveInstanceState(Bundle bundle) {
    super.onSaveInstanceState(bundle);
    bundle.putString("uneChaine", "sauvé");
}
```

developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)

Restaurer l'état de l'activité

- Lorsque l'activité redémarre, le bundle sauvé par *onSaveInstanceState()* est passé en paramètre aux méthodes :
  - *onCreate(b:Bundle);*
  - *onRestoreInstanceState(b:Bundle);* appelée après *onStart()*
- Utilisez les méthodes *get<type>* pour extraire les données du bundle.

```
@Override
protected void onRestoreInstanceState(Bundle bundle) {
    super.onRestoreInstanceState(bundle);
    Log.i("MonLog", bundle.getString("uneChaine"));
}
```

developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)