



# Workshop DataStorage

## Contexte :

L'Object est de créer une application nommé « Reporting » qui nous permet de lire et ajouté des articles dans une base de données interne.

## Objectif :

Comment manipuler la Base de données interne(SQLITE)?

Comment manipuler les préférences (SharedPreferences)?

Comment manipuler le variable d'application ?

## Enoncé :

Complétez le code manquant pour créer une base de données SQLITE, une variable d'application et enfin, gérer les préférences.

## Etapas à suivre :

Pour commencer,

MyApp.java

Pour créer une variable d'application :

```
public class MyApp extends Application{  
  
    private boolean connected = false;  
  
    public boolean isConnected() {  
        return connected;  
    }  
  
    public void setConnected(boolean connected) {  
        this.connected = connected;  
    }  
  
}
```

Déclarer la Classe MyApp.java dans le Manifest

```
<application  
    android:name="com.sim.datastorage.utils.MyApp"  
    android:icon="@drawable/icon_app_reports"  
    android:label="@string/app_name" >
```



Affecter une variable d'application

```
MyApp app;
```

```
app = (MyApp) getApplication();  
app.setConnected(true);
```

Utiliser une variable d'application Dans la méthode onCreate() de l'Activity ActivityAddArticle:

```
MyApp app;
```

```
app = (MyApp) getApplication();  
if (app.isConnected()) {  
    getActionBar().setIcon(R.drawable.icon_reports_connected);  
}  
else {  
    getActionBar().setIcon(R.drawable.icon_reports_notconnected);  
}
```

Pour affecter les préférences :

```
public static String PREFERENCE_FILENAME = "reporting_app";
```

```
SharedPreferences reportingPref = getSharedPreferences(PREFERENCE_FILENAME, MODE_PRIVATE);  
SharedPreferences.Editor prefEditor = reportingPref.edit();  
prefEditor.putString("login", editTextLogin.getText().toString());  
prefEditor.putString("password", editTextPassword.getText().toString());  
prefEditor.commit();
```

Pour lire les préférences :

```
SharedPreferences reportingPref = getSharedPreferences(PREFERENCE_FILENAME, MODE_PRIVATE);  
String login = reportingPref.getString("login", "");  
String password = reportingPref.getString("password", "");
```

DBHelper.java

Création de la table Article :

```
@Override  
public void onCreate(SQLiteDatabase db) {  
    db.execSQL(CREATE_ARTICLE);  
}
```



Recréation de la table :

```
@Override
public void onUpgrade(SQLiteDatabase db, int old, int newVersion) {

    db.execSQL("DROP TABLE IF EXISTS " + TABLE_ARTICLE + ";");
    db.execSQL(CREATE_ARTICLE);

}
```

ArticleBDD.java

Ouvrir la base de données :

```
public void open(){
    bdd = dbHelper.getWritableDatabase();
}
```

Fermer la base de données :

```
public void close(){
    bdd.close();
}
```

Insertion d'un objet dans la table article:

```
public long insertTop(Article partis){

    Cursor cursor = bdd.rawQuery
        ("select max(" + DBHelper.ID_ARTICLE + ") from " + DBHelper.TABLE_ARTICLE, null);
    if (cursor.moveToFirst()) partis.setId(cursor.getInt(0)+1);
    else partis.setId(1);

    ContentValues values = new ContentValues();
    values.put(DBHelper.ID_ARTICLE, partis.getId());
    values.put(DBHelper.TITLE_ARTICLE, partis.getTitle());
    values.put(DBHelper.DESC_ARTICLE, partis.getDesc());
    values.put(DBHelper.LOGO_ARTICLE, partis.getLogo());
    return bdd.insert(DBHelper.TABLE_ARTICLE, null, values);

}
```

Supprimer le contenu de la table:

```
public int removeAllarticles(){
    return bdd.delete(DBHelper.TABLE_ARTICLE, null, null);
}
```

Supprimer une ligne de la table article:

```
public int removeArticle(int index){
    return bdd.delete(DBHelper.TABLE_ARTICLE, "" + DBHelper.ID_ARTICLE + "`=?",
        new String[] {String.valueOf(index)});
}
```



Lister le contenu de la table article

```
public List<Article> selectAll() {  
    List<Article> list = new ArrayList<Article>();  
    Cursor cursor = this.bdd.query(DBHelper.TABLE_ARTICLE, new String[] {""},  
        null, null, null, null, null);  
    if (cursor.moveToFirst()) {  
        do {  
            Article article = new Article();  
            article.setId(cursor.getInt(0));  
            article.setTitle(cursor.getString(1));  
            article.setDesc(cursor.getString(2));  
            article.setLogo(cursor.getBlob(3));  
            list.add(article);  
        } while (cursor.moveToNext());  
    }  
    if (cursor != null && !cursor.isClosed()) {  
        cursor.close();  
    }  
    return list;  
}
```

ActivityAddArticle.java

Ajouter un article à la table concerné:

```
articleBDD = new ArticleBDD(getApplicationContext());  
articleBDD.open();  
articleBDD.insertTop(new Article(2, title, desc, Utilities.getBytes(bmp)));  
articleBDD.close();
```

ActivityListArticle.java

Lister les articles dans la base au démarrage de l'activity :

```
articleBDD = new ArticleBDD(getApplicationContext());  
articleBDD.open();  
articleList = articleBDD.selectAll();  
articleBDD.close();
```

Supprimer un article sélectionné (onItemClick()) :

```
articleBDD.open();  
articleBDD.removeArticle(articleList.get(arg2).getId());  
articleBDD.close();
```

Supprimer tous les articles dans la table :

```
articleBDD.open();  
articleBDD.removeAllArticles();  
articleBDD.close();
```

Bon Travail 😊