

JDBC : Application à un projet Android

Application JDBC : Développement d'une application Android-Jdbc avec une base de données Oracle.

Voici un exemple d'application que nous souhaitons développer



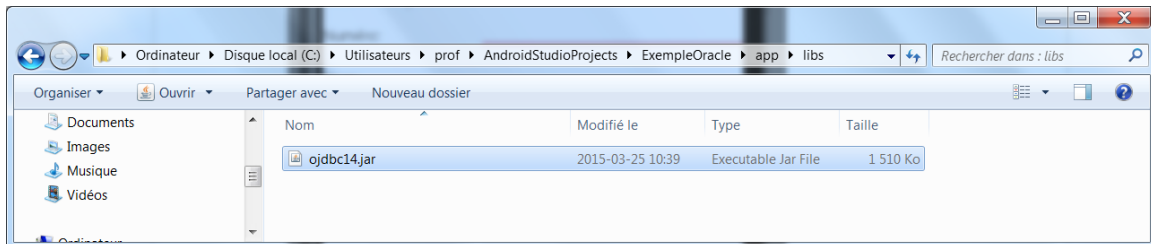
Avant de commencer :

Comme pour tout projet de type JDBC, vous devez inclure votre librairie .JAR qui contient le driver pour accéder à la base de données. Cette librairie est ojdbc14.jar

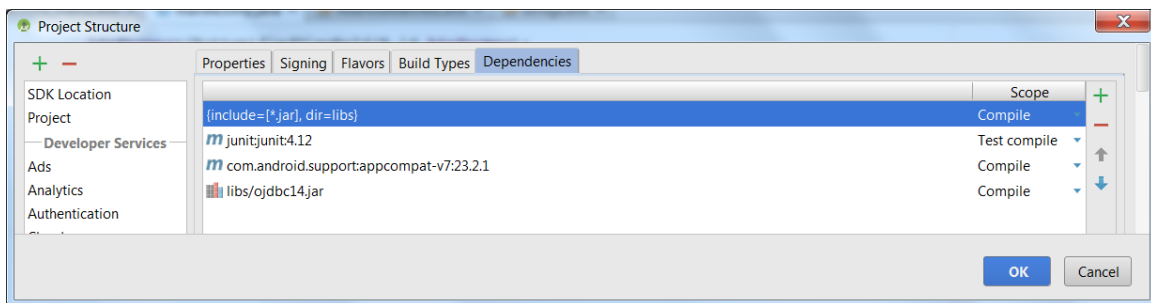
Étape 1 : Ajouter la librairie

Pour ajouter une librairie :

- 1.- Copier le fichier ojdbc14.jar dans le répertoire « libs » de votre projet.



- 2.- Faites « Files » « Project Structure »
- 3.- Sélectionner « Modules »
- 4.- Sélectionner « Dependencies »
- 5.- Appuyer sur +
- 6.- Sélectionner « File dependencie »
- 7.- Sélectionner votre librairie
- 8.- « OK »



Étape 2 : Donner les autorisations nécessaires dans le manifaeste pour que votre application ait accès à l'exterieur.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.prof.exempleoracle">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.INTERNET" />
</manifest>

```

Étape 3 : Important et nécessaire pour faire du JDBC

StrictMode est une classe spéciale pour vérifier que votre application Android ne fait pas des choses comme E / S disque, d'un accès réseau à partir du thread d'interface utilisateur. Il s'agit d'une fonctionnalité de débogage introduit dans Android 2.3.

Cet outil de développeur détecter les choses accidentellement et les apporter à votre attention afin que vous puissiez les corriger afin d'éviter les boîtes de dialogue de l'ANR (activité ne répond pas des dialogues).

Contenu de votre fichier **string.xml** :

```

<resources>
    <string name="app_name">ExempleOracle</string>

    <string name="num">Numéro: </string>
    <string name="nom">Nom: </string>
    <string name="prenom">Prénom: </string>
    <string name="courriel">Courriel: </string>
    <string name="ajouter">Ajouter</string>
    <string name="supprimer">Supprimer</string>
    <string name="modifier">Modifier</string>
    <string name="voir">Voir</string>
    <string name="voir_tout">Premier</string>
    <string name="fermer">Fermer</string>
    <string name="suivant">Suivant</string>
    <string name="precedent">Precedent</string>
    <string name="titre">Gestion des personnes </string>

</resources>

```

Contenu de votre fichier activity_main2.xml (c'est mon fichier à moi)

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/myLayout"
    android:stretchColumns="0"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        >

        <TableRow
            android:layout_width="fill_parent"
            android:layout_height="fill_parent">
            <TextView android:text="@string/titre"
                android:layout_column="1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
            />
        </TableRow>

        <TableRow
            android:layout_width="fill_parent"
            android:layout_height="fill_parent">
            <TextView android:text="@string/num"
                android:layout_column="1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
            />
            <EditText android:id="@+id/edtNum"
                android:inputType="number"
                android:layout_column="2"
                android:layout_width="150dp"
                android:layout_height="40dp"
            />
        </TableRow>

        <TableRow
            android:layout_width="fill_parent"
            android:layout_height="fill_parent">
            <TextView android:text="@string/nom"
                android:layout_column="1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"/>

            <EditText android:id="@+id/edtNom"
                android:inputType="text"
                android:layout_column="2"
                android:layout_width="150dp"
                android:layout_height="40dp"
            />
        </TableRow>
    </TableLayout>
</RelativeLayout>
```

```

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView android:text="@string/prenom"
        android:layout_column="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />
    <EditText android:id="@+id/edtPrenom"
        android:layout_column="2"
        android:layout_width="150dp"
        android:layout_height="40dp"
    />
</TableRow>

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView android:text="@string/courriel"
        android:layout_column="1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />
    <EditText android:id="@+id/edtCourriel"
        android:layout_column="2"
        android:layout_width="250dp"
        android:layout_height="40dp"
    />
</TableRow>

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

</TableRow>

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button android:id="@+id/btnAjouter"
        android:text="@string/ajouter"
        android:layout_column="1"
        android:layout_width="50dp"
        android:layout_height="40dp"
    />
    <Button android:id="@+id/btnSupprimer"
        android:text="@string/supprimer"
        android:layout_column="2"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_centerVertical="true"
    />
</TableRow>

```

```

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button    android:id="@+id/btnVoirTout"
        android:text="@string/voir_tout"
        android:layout_column="1"
        android:layout_width="50dp"
        android:layout_height="40dp"
    />

    <Button    android:id="@+id/btnFermer"
        android:text="@string/fermer"
        android:layout_column="2"
        android:layout_width="100dp"
        android:layout_height="40dp"
    />

</TableRow>

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button    android:id="@+id/btnModifier"
        android:text="@string/modifier"
        android:layout_column="1"
        android:layout_width="50dp"
        android:layout_height="40dp"
    />
    <Button    android:id="@+id/btnVoir"
        android:text="@string/voir"
        android:layout_width="50dp"
        android:layout_height="40dp"
        android:layout_column="2" />

</TableRow>

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button    android:id="@+id/btnSuivant"
        android:text="@string/suivant"
        android:layout_column="1"
        android:layout_width="100dp"
        android:layout_height="40dp"
    />
    <Button    android:id="@+id/btnPrecedent"
        android:text="@string/precedent"
        android:layout_width="100dp"
        android:layout_height="40dp"
        android:layout_column="2" />

</TableRow>

</TableLayout>
</RelativeLayout>

```

Code de Main MainActivity.java

```
package com.example.prof.exempleoracle;

import android.app.AlertDialog;
import android.database.Cursor;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.view.View.OnClickListener;
import android.app.AlertDialog.Builder;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class MainActivity extends AppCompatActivity implements
OnClickListener {

    ResultSet rst=null;
    public static Connection conn_ = null;
    private EditText edtNum, edtNom, edtPrenom, edtCourriel;
    private Button
    btnAjouter, btnSupprimer, btnModifieur, btnVoir, btnVoirTout,
    btnFermer, btnSuivant, btnPrecedent;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        StrictMode.setThreadPolicy(new
StrictMode.ThreadPolicy.Builder()
                .detectDiskReads()
                .detectDiskWrites()
                .detectNetwork()
                .penaltyLog() // Enregistre un message à
logcat
                .build());
        StrictMode.setVmPolicy(new StrictMode.VmPolicy.Builder()
                .detectLeakedSqlLiteObjects()
                .penaltyLog()
                .penaltyDeath() //l'application se bloque, fonctionne à
//la fin de toutes les sanctions permises
                .build());
```

Denis Brunet, Saliha Yacoub.

```
//appel de la connexion
OracleConnexion() ;

//Déclarations

edtNum=(EditText) findViewById(R.id.edtNum) ;
edtNom=(EditText) findViewById(R.id.edtNom) ;
edtPrenom=(EditText) findViewById(R.id.edtPrenom) ;
edtCourriel=(EditText) findViewById(R.id.edtCourriel) ;

btnAjouter=(Button) findViewById(R.id.btnAjouter) ;
btnAjouter.setOnClickListener(this) ;

btnSupprimer=(Button) findViewById(R.id.btnSupprimer) ;
btnSupprimer.setOnClickListener(this) ;

btnVoir=(Button) findViewById(R.id.btnVoir) ;
btnVoir.setOnClickListener(this) ;

btnVoirTout=(Button) findViewById(R.id.btnVoirTout) ;
btnVoirTout.setOnClickListener(this) ;

btnFermer=(Button) findViewById(R.id.btnFermer) ;
btnFermer.setOnClickListener(this) ;

btnSuivant=(Button) findViewById(R.id.btnSuivant) ;
btnSuivant.setOnClickListener(this) ;

btnPrecedent=(Button) findViewById(R.id.btnPrecedent) ;
btnPrecedent.setOnClickListener(this) ;

} // fin du onCreate
```


//Remarquez que la connexion est dans un Thread

```
private void OracleConnexion() {  
  
    Thread t= new Thread() {  
        @Override  
        public void run() {  
            try  
  
                {  
                    Class.forName("oracle.jdbc.OracleDriver");  
                } catch (   
                    ClassNotFoundException e  
                    )  
  
                {  
                    Toast.makeText(MainActivity.this, "Driver manquant." +  
e.getMessage().toString(), Toast.LENGTH_LONG).show();  
                }  
  
                String jdbcURL = "jdbc:oracle:thin:@222.222.344.241:1521:ORCL";  
                String user = "patoche";  
                String passwd = "blablaba";  
  
                try  
  
                    {  
conn_ = DriverManager.getConnection(jdbcURL,user,passwd);  
  
                } catch (   
                    java.sql.SQLException se  
                    )  
  
                    {  
                        Toast.makeText(MainActivity.this, "Connexion au serveur  
impossible." + se.getMessage().toString(), Toast.LENGTH_LONG).show();  
                    }  
  
                } // fin du run  
  
    }; // fin du thread  
        t.start();  
  
    } // fin de OracleConnexion
```

```

@Override
public void onClick(View view)
{
    if(view==btnFermer)
    {
        try {
            conn_.close();
            AfficheMessage("Connexion","connexion fermé");
        }
        catch (SQLException fe){}
    }

    if(view==btnVoir)
    {
        try {
            String sql2="select * from Personnes where num = ?" ;
            PreparedStatement pstmt = conn_.prepareStatement(sql2);
            int n = Integer.parseInt(edtNum.getText().toString());
            pstmt.setInt(1, n);

            ResultSet rst1 = pstmt.executeQuery();
            if (rst1.next()) {
                edtNom.setText(rst1.getString(2));
                edtPrenom.setText(rst1.getString(3));
                edtCourriel.setText(rst1.getString(4));
            }
        } catch (SQLException se) {
            Log.d("MainActivity", se.getMessage());
        }

        } //Fin if bouton voir

//Bouton ajouter
    if(view==btnAjouter)
    {
        try {
            String sqlins = "insert into personnes (num,
nom,prenom, courriel) values (sqpers.nextval,?,?,?)";

            PreparedStatement pstmtins =
conn_.prepareStatement(sqlins);
            pstmtins.setString(1,edtNom.getText().toString());
            pstmtins.setString(2,edtPrenom.getText().toString());
            pstmtins.setString(3,edtCourriel.getText().toString());
            pstmtins.executeUpdate();
            videTexte();
        } catch (SQLException seinst) {
            Toast.makeText(MainActivity.this, "liste." +
seinst.toString(), Toast.LENGTH_LONG).show();
            Log.d("MainActivity", seinst.getMessage());
        }

    } // Fin ajouter

```

```

// code du bouton Personne
// ce qui est en commentaire
//est pour afficher tous les enregistrements.
if(view==btnVoirTout)
{
    try {
        String sqltout="select * from Personnes " ;
        Statement pstmt =
conn .createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
        rst = pstmt.executeQuery(sqltout);
        // ResultSet rst2 = pstmt.executeQuery(sqltout);
        if(rst.first()) {
            edtNum.setText(((Integer)
rst.getInt(1)).toString());
            edtNom.setText(rst.getString(2));
            edtPrenom.setText(rst.getString(3));
            edtCourriel.setText(rst.getString(4));
        }
        /* StringBuffer buffer=new StringBuffer();
        while(rst2.next()) {
            buffer.append("num:" + rst2.getInt(1) + "\n");
            buffer.append("nom:" + rst2.getString(2)+ "\n");
            buffer.append("prenom:" + rst2.getString(3)+ "\n");
            buffer.append("courriel:" + rst2.getString(4)+
"\n");
            buffer.append("-----\n");
        }*/
        //AfficheMessage("Personne", buffer.toString());
    } catch (SQLException sel) {
        // Toast.makeText(MainActivity.this, "liste." +
sel.toString(), Toast.LENGTH_LONG).show();
        Log.d("MainActivity", sel.getMessage());
    }
}

if(view==btnSuivant)
{
    try {

        if (rst.next()) {

edtNum.setText(((Integer) rst.getInt(1)).toString());
            edtNom.setText(rst.getString(2));
            edtPrenom.setText(rst.getString(3));
            edtCourriel.setText(rst.getString(4));
        }
    } catch (SQLException se2) {
        Log.d("MainActivity", se2.getMessage());
    }
}

```

```

        if (view==btnPrecedent)
        {
            try {

                if (rst.previous()) {

                    edtNum.setText(((Integer)rst.getInt(1)).toString());
                    edtNom.setText(rst.getString(2));
                    edtPrenom.setText(rst.getString(3));
                    edtCourriel.setText(rst.getString(4));
                }
            } catch (SQLException se3) {

                Log.d("MainActivity", se3.getMessage());

            }

        }

    }

}

} //fin de méthode onclick

public void videTexte()
{
    edtNum.setText("");
    edtNom.setText("");
    edtPrenom.setText("");
    edtCourriel.setText("");
    edtNom.requestFocus();
}

public void AfficheMessage(String titre,String message)
{
    AlertDialog.Builder builder=new AlertDialog.Builder(this);
    builder.setCancelable(true);
    builder.setTitle(titre);
    builder.setMessage(message);
    builder.show();
}

} //Fin de classe

```

Travail final : Partie 1

Pondération 7%

Date de remise le 19 mai avant 16h10.

Mise en contexte :

L'agence Tourism est une agence qui œuvre dans le domaine du tourisme. Une de ces tâches consiste à faire visiter à ces clients de monuments historiques en leur proposant des circuits

Un circuit à une ville début, une ville d'arrivée un prix, un nom unique et la durée total pour le visiter. Il y a aussi le nombre maximum de clients par circuit.

- 1- Dans SQLDeveloper, créez la table Circuits dont la structure est la suivante :

Colonnes	Types et contrainte
IdCircuit	Number, primary key. Doit être fait par une séquence.
VilleDepart	Varchar2(30) not null
VilleArivee	Varchar2(30) not null
Prix	Number(4,2) not null
Duree	Number not null

- 2- Créer l'application JDBC-Android qui permet :
 - a. Ouvrir une connexion sur le serveur Oracle
 - b. D'ajouter un enregistrement
 - c. De supprimer un enregistrement
 - d. Rechercher un circuit selon la ville de départ. S'il y a plusieurs circuits on doit pouvoir se déplacer d'un enregistrement à l'autre par des boutons suivant et précédent.
 - e. De fermer la connexion.