

Departamento de Engenharia Informática

Minds-On



SQLite Database and ListView Adapter

Paulo Baltarejo Sousa
pbs@isep.ipp.pt
2016

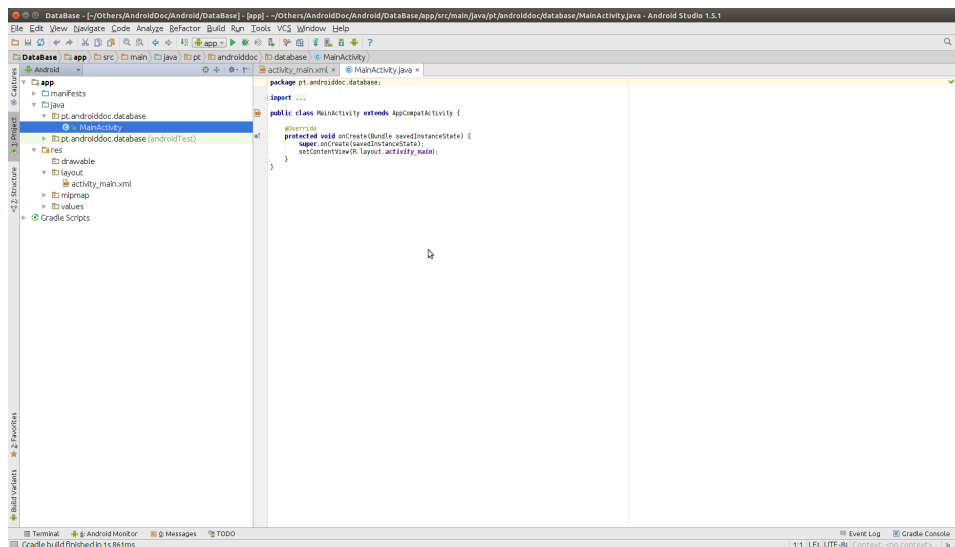
1 Implementing a simple Android SQLite app

This app shows how to use SQLite Database as well as the custom `ListView` adapter. It creates a basic list of people app.

1.1 Create a project

Start a new Android project.

1. Select **File > New > New Project....**
2. Fill in the project details with the following values:
 - New Project: Application name: DataBase
 - Target Android Devices: Phone and Tablet
 - Target Android Devices: Minimum SDK: API10
 - Add an Activity to Mobile: Empty Activity
 - ...



1.2 Changing the activity_main.xml file layout

From Android studio, open `activity_main.xml` file and change it content to:

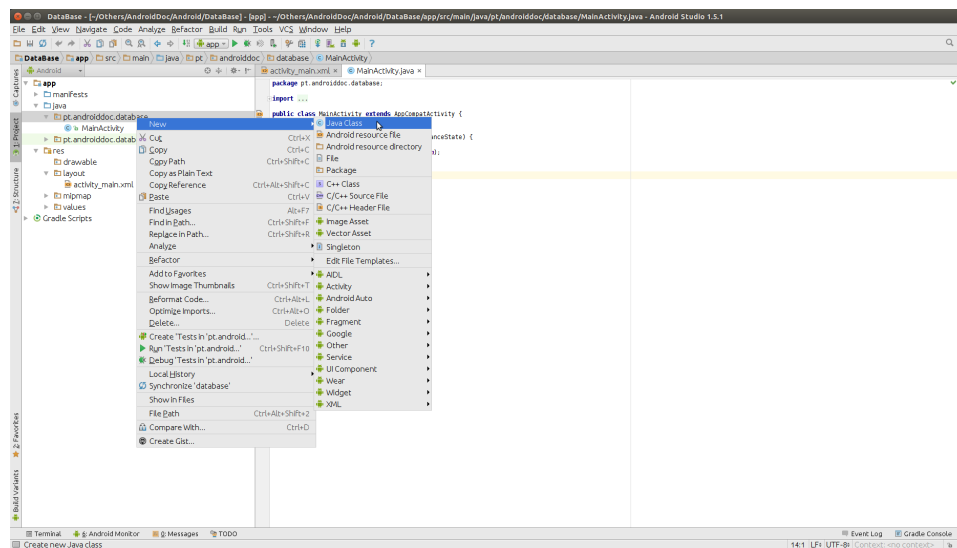
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="pt.androiddoc.database.MainActivity">

    <ListView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/listView"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
</RelativeLayout>
```

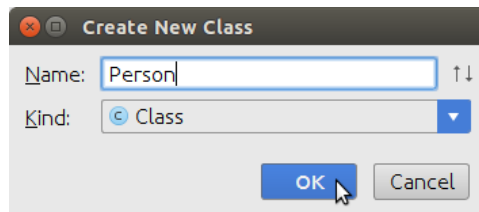
1.3 Adding a java class to model a singular person

From Android studio, right-click on pt.androiddoc.database package:

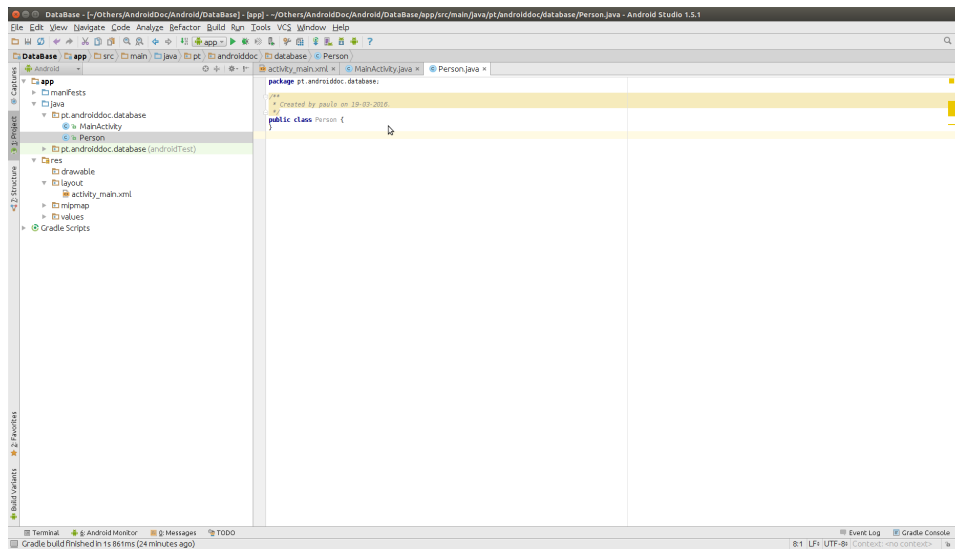
1. Select New > Java Class



2. Write Person for the java class name .



3. Click OK



4. Change Person.java content to:

```

public class Person {
    private int id;
    private String firstName;
    private String lastName;

    public Person(int id, String first,String last) {
        this.id = id;
        this.firstName = first;
        this.lastName = last;
    }
    public void setId(int id) {
        this.id = id;
    }
    public int getId() {
        return id;
    }
    public void setFirstName(String name) {
        this.firstName = name;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setLastName(String name) {
        this.lastName = name;
    }
    public String getLastName() {
        return lastName;
    }
}

```

1.4 SQLite database

1.4.1 Adding DBHelper java class

Create a new java class called DBHelper and change its content to:

```

public class DBHelper extends SQLiteOpenHelper {
    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "PeopleDB";
    private String createStatement = "";
    private String tableName;
    private Context context;

    public DBHelper(Context context, String tableName, String fields) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        this.createStatement = "CREATE TABLE IF NOT EXISTS ";
        this.createStatement += tableName + " (";
        this.createStatement += fields + ");";
        this.tableName = tableName;
        this.context = context;
    }

    @Override
    public void onCreate(SQLiteDatabase arg0) {
        arg0.execSQL(this.createStatement);
    }

    @Override
    public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2) {
        arg0.execSQL("DROP TABLE IF EXISTS " + this.tableName);
        onCreate(arg0);
    }
}

```

1.4.2 Adding DBAdapter java class

Create a new java class called DBAdapter and change its content to:

```

public class DBAdapter {
    private DBHelper dbHelper;

    private static final String TABLE = "People";
    private static final String ID = "id";
    private static final String FIRST_NAME = "firstName";
    private static final String LAST_NAME = "lastName";

    private static final String SELECT_PEOPLE = "SELECT * FROM " + TABLE;

    public DBAdapter(Context context) {
        dbHelper = new DBHelper(context, TABLE, ID + " INTEGER PRIMARY KEY
            AUTOINCREMENT,"
            + FIRST_NAME + " TEXT," + LAST_NAME + " TEXT");
    }

    public ArrayList<Person> getPeople() {
        ArrayList<Person> people = new ArrayList<Person>();
        SQLiteDatabase sqLiteDB = dbHelper.getReadableDatabase();
        Cursor cursor = sqLiteDB.rawQuery(SELECT_PEOPLE, null);
        cursor.moveToFirst();
        for (int i = 0; i < cursor.getCount(); i++){
            people.add(new Person(cursor.getInt(0),
                cursor.getString(1),cursor.getString(2)));
            cursor.moveToNext();
        }
        cursor.close();
        sqLiteDB.close();
        return people;
    }

    public Person getPerson(int _id) {
        SQLiteDatabase sqLiteDB = dbHelper.getReadableDatabase();
        String s = "SELECT * FROM " + TABLE + " WHERE " + ID + "=" + id;
        Cursor cursor = sqLiteDB.rawQuery(s, null);
        cursor.moveToFirst();
        Person person = new Person(cursor.getInt(0),
            cursor.getString(1),cursor.getString(2));
        cursor.close();
        sqLiteDB.close();
        return person;
    }

    public boolean insertPerson( String firstName, String lastName) {
        try {
            SQLiteDatabase db = dbHelper.getWritableDatabase();
            ContentValues initialValues = new ContentValues();
            initialValues.put(FIRST_NAME,firstName);
            initialValues.put(LAST_NAME, lastName);
            db.insert(TABLE, null, initialValues);
            db.close();
        } catch (SQLException sqlerror) {
            Log.v("Insert into table error", sqlerror.getMessage());
            return false;
        }
        return true;
    }
}

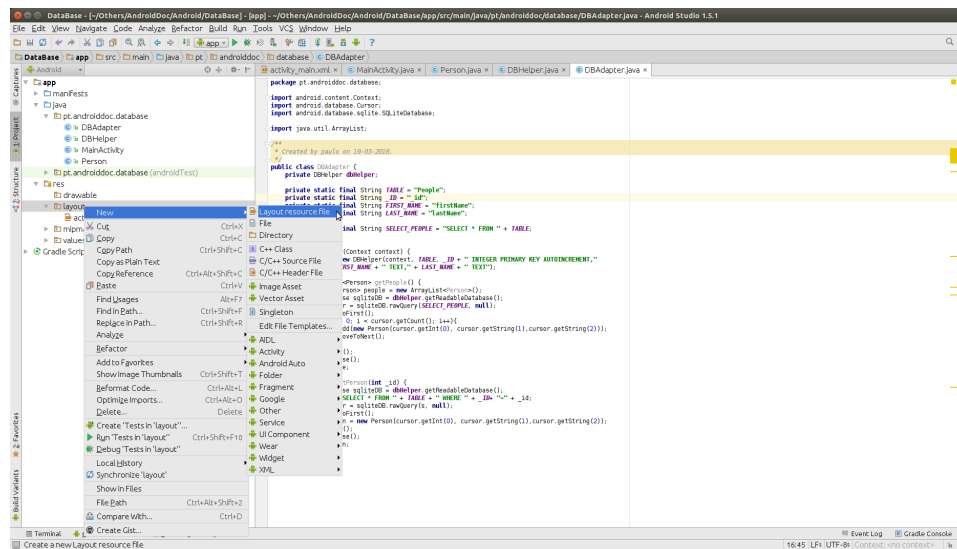
```

1.5 ListView adapter

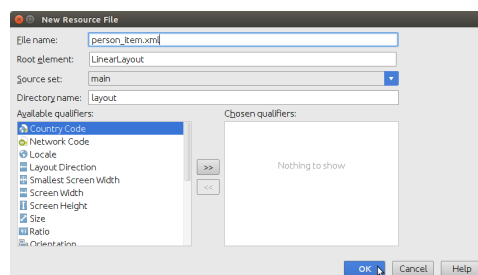
1.6 Creating a ListView item layout

From Android studio, right-click on `res/layout` directory:

1. Select **New > Layout resource file**



2. Write `person_item.xml` for the file name .



3. Click OK

4. Change `person_item.xml` content to:


```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
    android"
    android:orientation="horizontal" android:layout_width="
        match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:id="@+id/personId" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:id="@+id/firstName" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:id="@+id/lastName" />
</LinearLayout>

```

1.7 Creating a ListView adapter

Create a new java class called LVAdapter and change its content to:

```

public class LVAdapter extends BaseAdapter {
    Context context;
    int layout_id;
    private final List<Person> items;

    public LVAdapter(final Context context, final int layout_id,
final List<Person> items) {
        this.context = context;
        this.layout_id = layout_id;
        this.items = items;
    }
    public int getCount() {
        return this.items.size();
    }

    public Object getItem(int arg0) {
        return this.items.get(arg0);
    }

    public long getItemId(int arg0) {
        return 0;
    }

    public View getView(int arg0, View arg1, ViewGroup arg2) {
        final Person row = this.items.get(arg0);
        View itemView = null;
        if (arg1 == null) {
            LayoutInflater inflater = (LayoutInflater)
                context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            itemView = inflater.inflate(layout_id, null);
        } else {
            itemView = arg1;
        }

        TextView txtId = (TextView) itemView.findViewById(R.id.personId);
        txtId.setText(Integer.toString(row.getId()));

        TextView firstName = (TextView) itemView.findViewById(R.id.firstName
        );
        firstName.setText(row.getFirstName());

        TextView lastName = (TextView) itemView.findViewById(R.id.lastName);
        lastName.setText(row.getLastName());

        return itemView;
    }
}

```

1.8 Using SQLite database and ListView Adapters in an Activity

From Android studio, open MainActivity.java file and change it content to:

```
public class MainActivity extends AppCompatActivity {
    ListView lv;
    DBAdapter dbAdapter;
    LVAdapter lvAdapter;
    ArrayList<Person> listPerson;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        dbAdapter = new DBAdapter(this);
        lv = (ListView) findViewById(R.id.listView);
        listPerson = new ArrayList<Person>();
        listPerson.addAll(dbAdapter.getPeople());
        lvAdapter = new LVAdapter(this, R.layout.person_item, listPerson);
        lv.setAdapter(lvAdapter);
    }
}
```

1.9 Adding some Data

1. Create a new java class called PeopleData and change its content to:

```

public class PeopleData {

    public PeopleData() { }

    ArrayList<Person> getPeopleData()
    {
        String strJson= "{\"people\": [ " +
            "{\"id\":\"1\",\"fname\": \"Maria\",\"lname\": \"Antonia\" }," +
            "{\"id\":\"2\",\"fname\": \"Marco\", \"lname\": \"Janota\"}," +
            "{\"id\":\"3\",\"fname\": \"Manuel\",\"lname\": \"Ribeiro\"}," +
            "{\"id\":\"4\",\"fname\": \"Ana\",\"lname\": \"Pereira\"}" +
            "]}";
        ArrayList<Person> people = new ArrayList<Person>();
        try {
            JSONObject jsonRootObject = new JSONObject(strJson);
            JSONArray jsonArray = jsonRootObject.optJSONArray("people");
            for(int i=0; i < jsonArray.length(); i++){
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                int id = Integer.parseInt(jsonObject.optString("id").toString());
                String fname = jsonObject.optString("fname").toString();
                String lname = jsonObject.optString("lname").toString();
                people.add(new Person(id,fname,lname));
            }

        } catch (JSONException e) {e.printStackTrace();}

        return people;
    }
}

```

2. Open DBHelper java class file and change its content to:

```

public class DBHelper extends SQLiteOpenHelper {
    ...
    @Override
    public void onCreate(SQLiteDatabase arg0) {
        arg0.execSQL(this.createStatement());

        insertPeopleData(arg0);
    }

    ...
    public void insertPeopleData(SQLiteDatabase arg0){
        String sql;
        ArrayList<Person> list;
        PeopleData data = new PeopleData();
        list = data.getPeopleData();
        for(Person elem : list){
            sql = "INSERT INTO " + this.tableName +
                " VALUES (" +elem.getId()+", '"+ elem.getFirstName() + "','"+
                    +
                    elem.getLastName() + "')";
            arg0.execSQL(sql);
            sql="";
        }
    }
}

```

2 Running

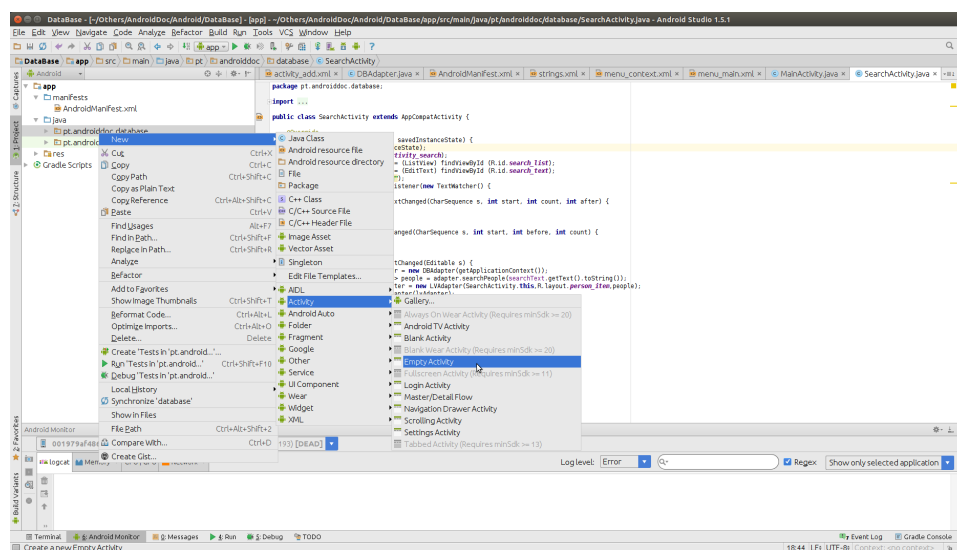
1. To run the app from Android Studio: Click Run from the toolbar.

DataBase		
1	Maria	Antonia
2	Marco	Janota
3	Manuel	Ribeiro
4	Ana	Pereira

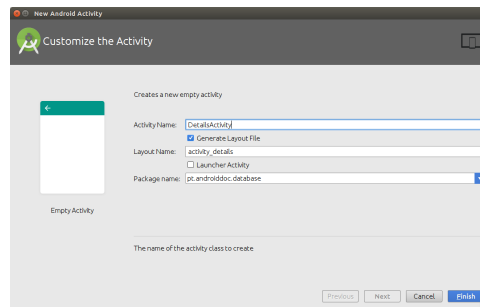
3 Presenting Person Details

From Android studio, right-click on `pt.androiddoc.database` package:

1. Select **New > Activity > Empty Activity**



2. Write `DetailsActivity` for the Activity name.



3. Click **Finish**

4. Change `activity_details.xml` content to:

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/
    android"
    xmlns:tools="http://schemas.android.com/tools" android:
        layout_width="match_parent"
        android:layout_height="match_parent" android:paddingLeft="@dimen/
            activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
        android:paddingBottom="@dimen/activity_vertical_margin"
        tools:context=".DetailsActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:id="@+id/textView2"
        android:layout_below="@+id/textView"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:id="@+id/textView3"
        android:layout_below="@+id/textView2"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
</RelativeLayout>

```

5. Change DetailsActivity.java content to:


```

public class DetailsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_details);
        int id = getIntent().getIntExtra("PERSON_ID", 0);
        DBAdapter adapter = new DBAdapter(getApplicationContext());
        Person person = adapter.getPerson(id);
        if(person != null){
            TextView person_id = (TextView) findViewById(R.id.textView
                );
            person_id.setText(Integer.toString(person.getId()));
            TextView first_name = (TextView) findViewById(R.id.
                textView2);
            first_name.setText(person.getFirstName());
            TextView last_name = (TextView) findViewById(R.id.
                textView3);
            last_name.setText(person.getLastName());
        }
    }
}

```

6. Open MainActivity.java file and change the onCreate method content to:

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

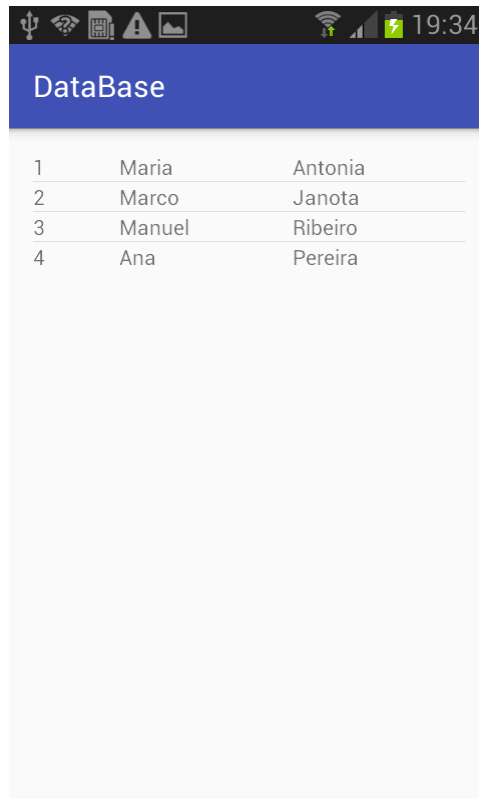
    dbAdapter = new DBAdapter(this);
    lv = (ListView) findViewById(R.id.listView);
    listPerson = new ArrayList<Person>();
    listPerson.addAll(dbAdapter.getPeople());
    lvAdapter = new LVAdapter(this, R.layout.person_item,
        listPerson);
    lv.setAdapter(lvAdapter);

    lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> arg0, View arg1,
            int arg2, long arg3) {
            Intent intent = new Intent(MainActivity.this,
                DetailsActivity.class);
            Person p = (Person)lvAdapter.getItem(arg2);
            intent.putExtra("PERSON_ID", p.getId());
            startActivity(intent);
        }
    });
}

```

4 Running Again

1. To run the app from Android Studio: Click **Run** from the toolbar.



2. Click on an Item List

