



大话企业级 Android 开发 • 第十四部分

本教程说明及版权声明

- 《大话企业级 Android 开发》是国士工作室为了方便中国 Android 开发者，推动 Android 企业级应用开发，特投入大量心血撰写的书籍，并在网络上免费发布，希望为移动互联网和智能手机时代贡献绵薄之力！所有相关文档版权均属国士工作室所有。
- 本教程是由国士工作室参考官方文档，综合市面相关书籍，经过充分的吸收消化，结合开发实践的一部原创作品，为了本教程及早与广大读者同仁见面、分享，特采用定稿一部分就发布一部分的连载方式发布。读者可以在本博客获取最新内容。
- 未经国士工作室授权，禁止将此文档及其衍生作品以标准（纸质）书籍形式发行。
- 本文档受有关法律的版权保护，对本文档内容的任何未经同意的复制和抄袭行为，将导致相应的法律责任。未经国士工作室同意，任何团体及个人不能用此教程牟利，违者必究。但是：在不收取其他人费用的前提下，您可以自由传播此文档，但必须保证版权信息、文档及其自带标示的完整性。
- 如果对该文档有任何疑问或者建议，请进入官方微博客
<http://www.cnblogs.com/guoshiandroid/>留言或者直接与国士工作室联系（后附联系方式），我们会慎重参考您的建议并根据需要对本文档进行修改，以造福更多开发者！
- 《大话企业级 Android 开发》的最新及完整内容会在国士工作室官方微博客定期更新，请访问国士工作室博客
<http://www.cnblogs.com/guoshiandroid/>获取更多更新内容。





关于国土工作室

我们(国土工作室)是一支专注于 Android 平台企业级应用开发的技术团队，对娱乐多媒体应用有着深刻的理解及研发能力，致力服务于企业用户。为音视频等娱乐多媒体网站、门户网站、SNS、论坛、电子商务等传统网络应用向移动互联网发展提供解决方案和技术支持，为企业提供 Android 培训服务等多种业务。

我们尤其擅长于提供从 Android 客户端到服务端的一站式解决方案和技术支持，服务端可以采用 Java EE，也可以采用轻量级流行的 LAMP 技术体系。目前，研发出了比 KU6、优酷更加强大和完善的 Android 视频网站娱乐多媒体客户端软件，并在持续升级中。

目前，我们正在务实而卓有成效的与音视频等娱乐多媒体网站、门户网站、SNS、论坛、电子商务等传统网络服务商合作，发展迅速，渴望有志之士的加入，和我们一起为成为世界最好的 Android 软件开发和咨询、培训公司而奋斗，为移动互联网和智能手机时代贡献力量！

联系我们

电话:15711060468

Email:guoshiandroid@gmail.com

博客: <http://www.cnblogs.com/guoshiandroid/>





1.1 事务

小安: 到现在我们都没有使用到事务, 但是在有一些情况下, 比如要同时执行

好多条操作数据库的语句, 这些语句一起成功, 或者一起失败, 此时就需

要事务控制。如果我们需要事务, 那么该如何做呢?

大致: 问的很好, 我们可以做一个测试你就会很清楚的明白了。

在原有的业务类PersonService中编写一个saves()方法, 这个方法执行两条SQL语句, 这两条SQL语句都是向表中添加数据, 在不加入事务的情况下, 它的代码如下:

```
public void saves() { //测试事务
    SQLiteDatabase db = databaseHelper.getWritableDatabase();
    db.execSQL("insert into person(name, age) values(?,?)", new Object[]{"张三", 21});
    db.execSQL("insert into person(name, age) values(?,?)", new Object[]{"李四", 22});
}
```

在上述代码中如果第二条SQL语句有错误, 第一条SQL语句也会成功插入数据, 但是加入事务之后就不同了, 代码如下:

```
public void saves() {
    SQLiteDatabase db = databaseHelper.getWritableDatabase();
    db.beginTransaction(); //开始事务
    try
    {
        db.execSQL("insert into person(name, age) values(?,?)", new Object[]{"张三", 21});
        db.execSQL("insert into person(name, age) values(?,?)", new Object[]{"李四", 22});
        db.setTransactionSuccessful(); //设置事务成功标志
    } catch (Exception e)
    {
    }
    db.endTransaction(); //事务结束
}
```

加入事务之后, 如果第二条SQL语句有错误, 变为:

```
db.execSQL("insert into persondddf(name, age) values(?,?)", new Object[]{"李四
```





, 22})

其中表名“person”变为了“persondddf”，在PersonServiceTest测试类中加入testSaves()方法，测试是否可以加入数据，代码如下：

```
public void testSaves() throws Throwable //测试事务
{
    PersonService personService = new PersonService(this.getContext());
    //传入上下文
    personService.saves();
}
```

执行testSaves()方法测试，结果显示如图3-27



图3-27

然后执行testGetScrollData()方法，查看是否有数据“张三”，如图3-28

Time	pid	tag	Message
03-27 21:24...	I 1...	PersonServiceTest	Person [age=21, id=1, name=Tom0]
03-27 21:24...	I 1...	PersonServiceTest	Person [age=21, id=2, name=Tom1]
03-27 21:24...	I 1...	PersonServiceTest	Person [age=21, id=3, name=Tom2]
03-27 21:24...	I 1...	PersonServiceTest	Person [age=21, id=4, name=Tom3]
03-27 21:24...	I 1...	PersonServiceTest	Person [age=21, id=5, name=Tom4]
03-27 21:24...	I 1...	PersonServiceTest	Person [age=21, id=6, name=Tom5]
03-27 21:24...	I 1...	PersonServiceTest	Person [age=21, id=7, name=Tom6]
03-27 21:24...	I 1...	PersonServiceTest	Person [age=21, id=8, name=Tom7]
03-27 21:24...	I 1...	PersonServiceTest	Person [age=21, id=9, name=Tom8]
03-27 21:24...	I 1...	PersonServiceTest	Person [age=21, id=10, name=Tom9]



图3-28

我们发现“张三”的数据没有被插入数据库。

有关数据库的基本操作，现在我们已经基本完成了，掌握前面的所有内容之后，基本就可以解决今后在项目中有数据库的所有业务需求。





1.2 使用 ListView 显示表中的数据

小安：大致博士，您把数据库的操作和测试，都讲完了，但是数据如何在手机上显示出来呢？有没有像列表那类东西？

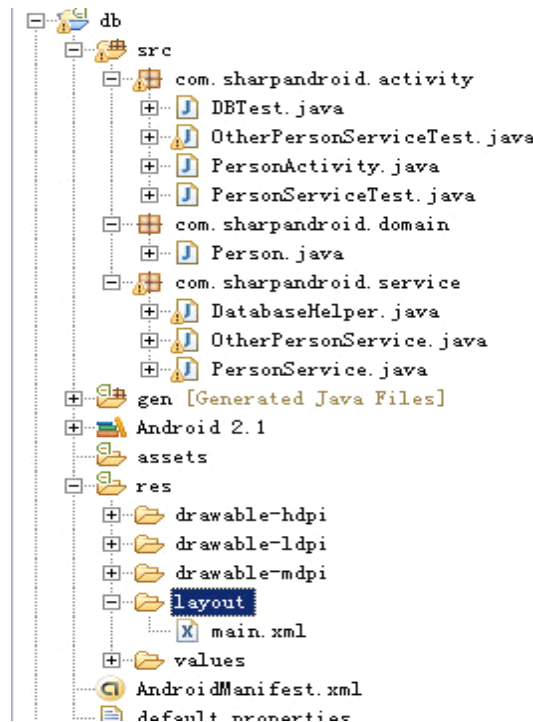
大致：孺子可教。对于数据的显示，我们可以这个时候就会用到 ListView 显示控件，它的显示风格如图 3-29



图3-29

要实现上图的效果，我们需要完成两个内容，第一个内容是每一行的显示风格，ListView 的每一行有三个元素，这个显示风格需要用做界面来完成。它的风格和排版是我们自己定义的，ListView 并不管，它只管将数据按照我们定义的风格去一行行显示。每一行的风格要怎么做呢？我们可以使用布局文件来实现，布局文件所在的文件夹，如图3-30





编写personitem.xml文件

创建一个布局文件personitem.xml, 在layout文件下, 它的作用就是定义ListView每一行的风格和布局, 如图3-31



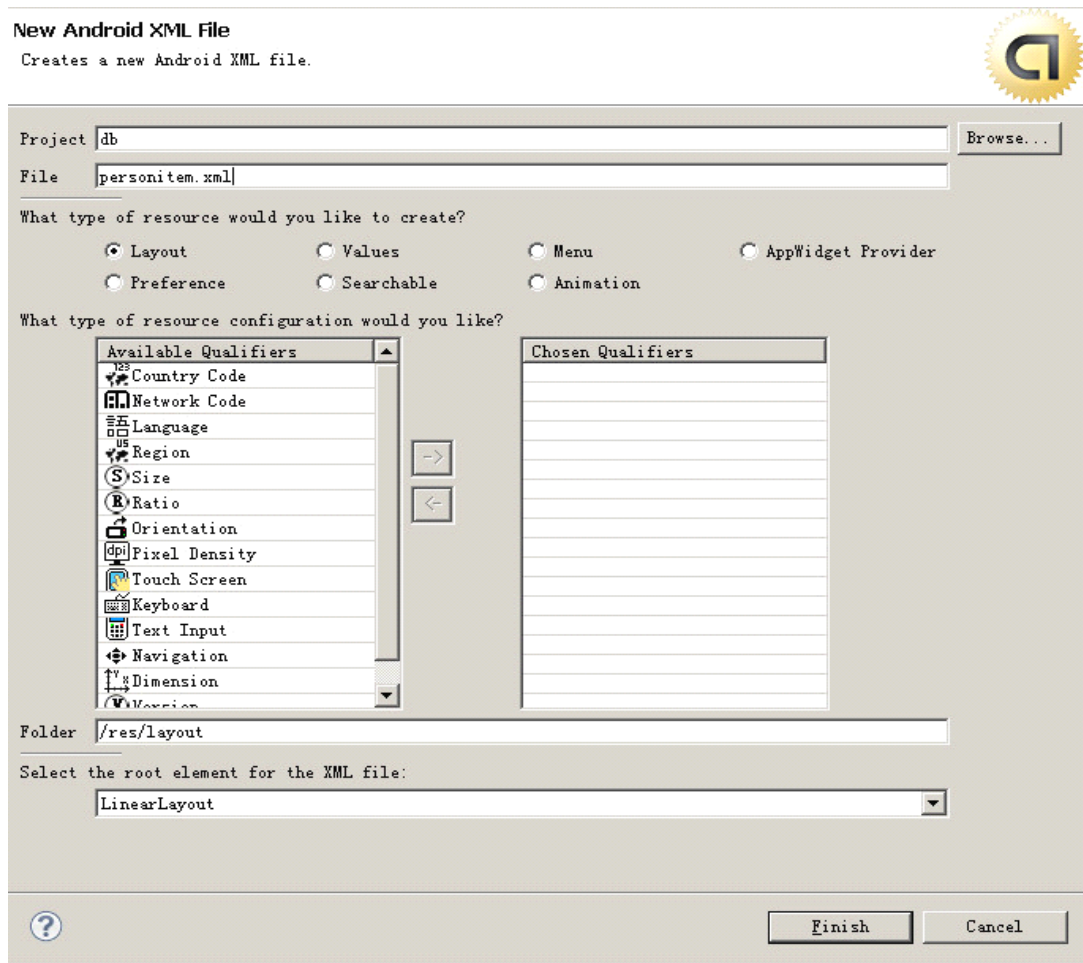


图3-31

创建成功之后，它的代码如下：

```
Personitem.xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="60px"
        android:layout_height="wrap_content"
        android:id="@+id/personid"
    >
</TextView>
    <TextView
        android:layout_width="160px"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@id/personid "
```





```

        android:layout_alignTop="@id/personid "
        android:gravity="center_horizontal"
        android:id="@+id/name"
    >
</TextView>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@id/name"
    android:layout_alignTop="@id/name"
    android:id="@+id/age"
    >
</TextView>
</RelativeLayout>

```

编写main.xml文件

因为PersonActivity中，它显示的主界面为main.xml，所以我们打开main.xml，为它加入一个ListView组件，代码如下：

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<ListView
    android:id="@+id/personList"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    ></ListView>
</LinearLayout>

```

这个时候我们就需要为ListView绑定数据，ListView中有一个setAdapter()方法设置适配器，也就是说数据是通过适配器传入ListView的。在Android系统中我们常用的适配器有ArrayAdapter，我们这次用到的适配器SimpleAdapter，以及采用查询结果集作为数据来源的适配器SimpleCursorAdapter。实例化一个SimpleAdapter，然后为它绑定数据，代码如下：

PersonActivity.java

```

public class PersonActivity extends Activity {
    /** Called when the activity is first created. */
    private final static String TAG="PersonActivity";

```





```
private ListView listView;
private PersonService personService;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    listView = (ListView) findViewById(R.id.personList);
    List<HashMap<String, String>> data = new ArrayList<HashMap<String, String>>();
    HashMap<String, String> title = new HashMap<String, String>();
    title.put("personid", "编号");
    title.put("name", "姓名");
    title.put("age", "年龄");
    data.add(title);
    SimpleAdapter adapter = new SimpleAdapter(PersonActivity.this,
        data, R.layout.personitem, new String[]{"personid", "name", "age"},
        new int[]{R.id.personid, R.id.name, R.id.age});
    listView.setAdapter(adapter);
}
```

在实例化SimpleAdapter中的参数：

第一个参数：表示上下文。

第二个参数：表示内部是继承map结构的list。

第三个参数：表示资源的id。

第四个参数：表示根据title中的key获取title中的值。

第五个参数：表示title中的数据显示所对应的TextView, 如图3-32

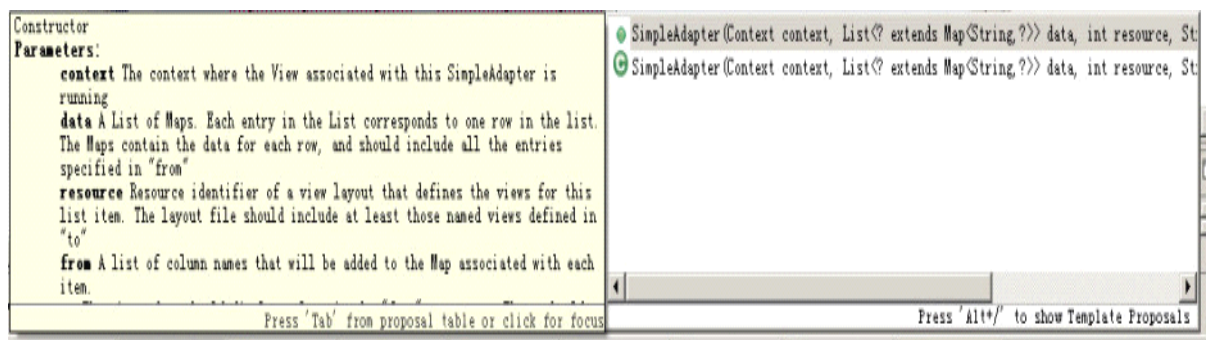


图 3-32

运行PersonActivity的界面显示为图3-33





图 3-33

图3-33只是显示标题，并没有绑定数据，如何显示内容呢？那就得用到业务类 PersonService，调用它的数据分页getScrollData()方法来得到数据，然后用迭代将数据都加入到data中，代码如下：

PersonActivity.java

```
public class PersonActivity extends Activity {
    /** Called when the activity is first created. */
    private final static String TAG="PersonActivity";
    private ListView listView;
    private PersonService personService;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        listView = (ListView)findViewById(R.id.personList);
        personService = new PersonService(this);
        List<Person> persons = personService.getScrollData(0, 10); //前十条数据
        List<HashMap<String, String>> data = new
        ArrayList<HashMap<String, String>>();
        HashMap<String, String>title = new HashMap<String, String>();
        title.put("personid", "编号");
        title.put("name", "姓名");
        title.put("age", "年龄");
        data.add(title); //标题
        for(Person person : persons) {
            HashMap<String, String> map = new HashMap<String, String>();
            map.put("personid", String.valueOf(person.getId()));
```





```
map.put("name", person.getName());
map.put("age", String.valueOf(person.getAge()));
data.add(map);    //显示各个数据
}
SimpleAdapter adapter = new SimpleAdapter(PersonActivity.this,
    data, R.layout.personitem, new String[]{"personid", "name",
"age"},
    new int[]{R.id.personid, R.id.name, R.id.age});
listView.setAdapter(adapter);
}
}
```

运行界面效果如图3-34



图 3-34

通过图3-34发现，数据已经列出了，这个列表的每一行是可以点击的如图3-34





图 3-35

但是，我们怎么知道点击的是哪行呢？当我们点击ListView每一行的时候，会触发一个事件 `setOnItemClickListener`，代码如下：

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view,  
        int position, long id) {  
        // TODO Auto-generated method stub  
    }  
});
```

当我们点击ListView某一项之后，系统会调用 `onItemClick()` 方法，它的 `parent` 参数就是上面的ListView，`View` 参数表示每一项最顶级的那个元素，如图3-36所选的内容，`position` 参数表示每一项数据在adapter中的位置编号，`id` 参数表示每一项在ListView中的位置编号。



```

1<?xml version="1.0" encoding="utf-8"?>
2<RelativeLayout
3    xmlns:android="http://schemas.android.com/apk/res/android"
4    android:layout_width="fill_parent"
5    android:layout_height="wrap_content">
6    <TextView
7        android:layout_width="60px"
8        android:layout_height="wrap_content"
9        android:id="@+id/personid"
10    >
11    </TextView>
12    <TextView
13        android:layout_width="160px"
14        android:layout_height="wrap_content"
15        android:layout_toRightOf="@id/personid"
16        android:layout_alignTop="@id/personid"
17        android:gravity="center_horizontal"
18        android:id="@+id/name"
19    >
20    </TextView>
21    <TextView
22        android:layout_width="wrap_content"
23        android:layout_height="wrap_content"
24        android:layout_toRightOf="@id/name"
25        android:layout_alignTop="@id/name"
26        android:id="@+id/age"
27    >
28    </TextView>
29</RelativeLayout>
30

```

图3-36

它的文档说明不是很清楚，我们通过打印和比较，可以测试一下参数position和id是否相等，在onItemClick()方法中加入代码：

```
Log.i("position and id", "result="+( position==id));
```

加入过滤器“position and id”，单击列表的一项如图3-35，运行结果如图3-37

Log position and id			
Time	pid	tag	Message
03-28 10:34...	I 309	position and id	result=true

图3-37

Position和id两个元素在本次应用中是相等的，如何获取单击那项的内容呢？在onItemClick()方法中加入代码如下：

```

public void onItemClick(AdapterView<?> parent, View view,
    int position, long id) {
    // TODO Auto-generated method stub
    ListView listView = (ListView)parent;
    HashMap<String, String> itemData = (HashMap<String,
    String>)listView.getItemAtPosition(position);
    String personid = itemData.get("personid");
    String name = itemData.get("name");

```





```

String age = itemData.get("age");
Log.i(TAG, "className="+ view.getClass().getName());
//打印 view 的类名
Log.i(TAG, "personid="+ personid+ "name="+name + "age"+ age);
Log.i(TAG, "result="+ (position==id));

    }
});

```

运行本应用的方法，如图3-38，

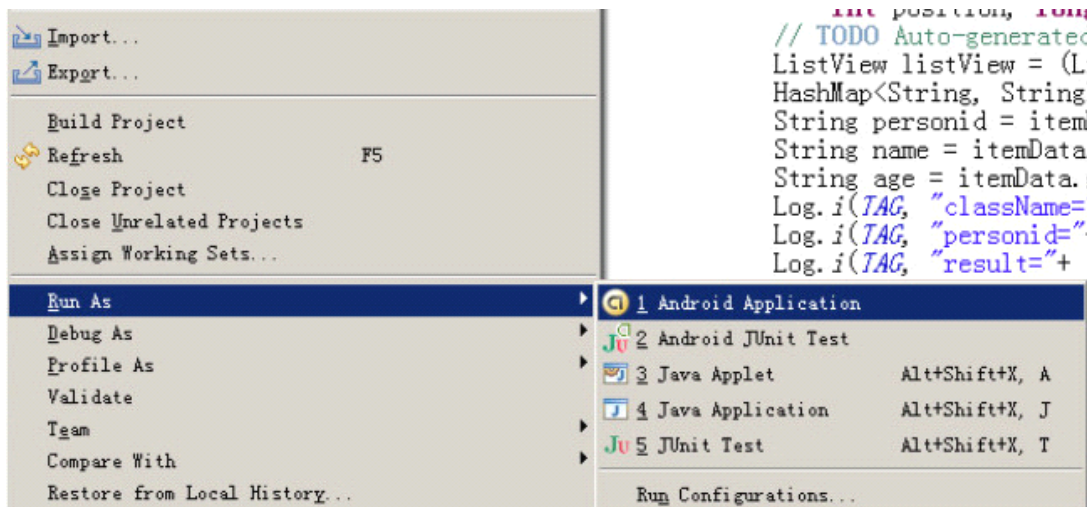


图3-38

运行成功之后，如图3-35点击列表的一项，出现的结果如图3-39

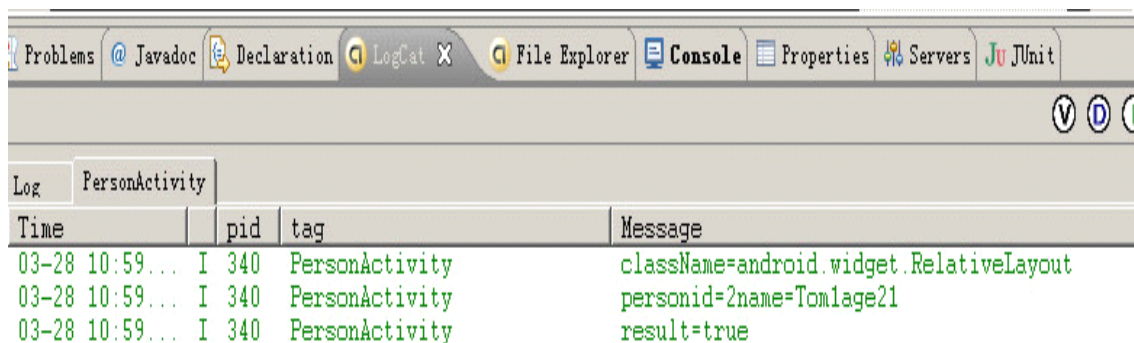


图3-39

通过运行结果，我们得到了被单击元素的值和视图的布局RelativeLayout，并且知道position和id相等。上述ListView的使用方法，在实际应用中会经常用到，所以大家一定要





掌握。

如图3-40，所选择的内容不属于数据项，它只是一个标题，我们可以将它固定放到 main.xml 文件中，代码如下：

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/positionid"
            android:id="@+id/idTitle"
            >
        </TextView>
        <TextView
            android:layout_width="200px"
            android:layout_height="wrap_content"
            android:layout_toRightOf="@id/idTitle"
            android:layout_alignTop="@id/idTitle"
            android:gravity="center_horizontal"
            android:text="@string/name"
            android:id="@+id/nameTitle"
            >
        </TextView>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_toRightOf="@id/nameTitle"
            android:layout_alignTop="@id/nameTitle"
            android:text="@string/age"
            android:id="@+id/ageTitle"
            >
        </TextView>
    </RelativeLayout>
</LinearLayout>
```





```

        android:id="@+id/personList"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    ></ListView>
</LinearLayout>
String.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, PersonActivity!</string>
    <string name="app_name">数据库应用</string>
    <string name="positionid">编号</string>
    <string name="name">姓名</string>
    <string name="age">年龄</string>
</resources>

```



编号	姓名	年龄
1	Tom0	21
2	Tom1	21
3	Tom2	21
4	Tom3	21
5	Tom4	21
6	Tom5	21
7	Tom6	21
8	Tom7	21
9	Tom8	21
10	Tom9	21

图3-40

然后，把PersonActivity中加入标题的代码去掉，就变为如下代码：

PersonActivity.java

```

public class PersonActivity extends Activity {
    /** Called when the activity is first created. */
    private final static String TAG="PersonActivity";
    private ListView listView;
    private PersonService personService;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        listView = (ListView)findViewById(R.id.personList);
    }
}

```





```

        personService = new PersonService(this);
        List<Person> persons = personService.getScrollData(0, 10); //前十条数据
        List<HashMap<String, String>> data = new
ArrayList<HashMap<String, String>>();
        for(Person person : persons){
            HashMap<String, String> map = new HashMap<String, String>();
            map.put("personid", String.valueOf(person.getId()));
            map.put("name", person.getName());
            map.put("age", String.valueOf(person.getAge()));
            data.add(map); //显示各个数据
        }
        SimpleAdapter adapter = new SimpleAdapter(PersonActivity.this,
            data, R.layout.personitem, new String[]{"personid", "name",
"age"},
            new int[]{R.id.personid, R.id.name, R.id.age});
        listView.setAdapter(adapter);

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> parent, View view,
                int position, long id) {
                // TODO Auto-generated method stub
                ListView listView = (ListView)parent;
                HashMap<String, String> itemData = (HashMap<String,
String>)listView.getItemAtPosition(position);
                String personid = itemData.get("personid");
                String name = itemData.get("name");
                String age = itemData.get("age");
                Log.i(TAG, "className="+ view.getClass().getName()); //打印view
的类名

                Log.i(TAG, "personid="+ personid+ "name="+name + "age"+ age);
                Log.i(TAG, "result="+ (position==id));

            }

        });
    }
}

```

运行本应用，结果如图3-41





编号	姓名	年龄
1	Tom0	21
2	Tom1	21
3	Tom2	21
4	Tom3	21
5	Tom4	21
6	Tom5	21
7	Tom6	21
8	Tom7	21
9	Tom8	21
10	Tom9	21

图3-41

数据和标题已经分开，标题已经不能被单击选中了。

1.3 使用 SimpleCursorAdapter 绑定数据

除了前面使用 SimpleAdapter 绑定数据，我们还可以使用 SimpleCursorAdapter 绑定，使用 SimpleCursorAdapter 更适合绑定数据库的记录集，在业务类中查询只返回 Cursor 就可以，然后将 Cursor 赋给 SimpleCursorAdapter，下面我们用 SimpleCursorAdapter 去实现数据的绑定。

首先，在业务类 PersonService 中加入返回游标的方法 getRawScrollData()，代码如下：

PersonService.java

```
public Cursor getRawScrollData(int startResult, int maxResult) {
    List<Person> persons = new ArrayList<Person>();
    SQLiteDatabase database = databaseHelper.getWritableDatabase();
    return database.rawQuery("select personid , name, age from person limit ?,?",
        new String[] {String.valueOf(startResult),
            String.valueOf(maxResult)});
}
```

然后在 PersonActivity 类中使用 SimpleCursorAdapter 绑定数据，代码如下：

PersonActivity.java

```
public class PersonActivity extends Activity {
    /** Called when the activity is first created. */
    private final static String TAG="PersonActivity";
    private ListView listView;
    private PersonService personService;
```





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

    listView = (ListView)findViewById(R.id.personList);
    personService = new PersonService(this);
    /* 使用 SimpleCursorAdapter 绑定数据*/
    Cursor cursor = personService.getRawScrollData(0, 10); //得到游标
    SimpleCursorAdapter adapter = new SimpleCursorAdapter(this,
R.layout.personitem, cursor,
        new String[]{"personid", "name", "age"}, new int[]{R.id.personid,
R.id.name, R.id.age});

    listView.setAdapter(adapter);

    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

        @Override
        public void onItemClick(AdapterView<?> parent, View view,
            int position, long id) {
            // TODO Auto-generated method stub
            ListView listView = (ListView)parent;
            HashMap<String, String> itemData = (HashMap<String,
String>)listView.getItemAtPosition(position);
            String personid = itemData.get("personid");
            String name = itemData.get("name");
            String age = itemData.get("age");
            Log.i(TAG, "className="+ view.getClass().getName()); //打印view
的类名

            Log.i(TAG, "personid="+ personid+ "name="+name + "age"+ age);
            Log.i(TAG, "result="+ (position==id));

        }
    });
}
```

在实例化SimpleCursorAdapter的时候的参数如图3-42



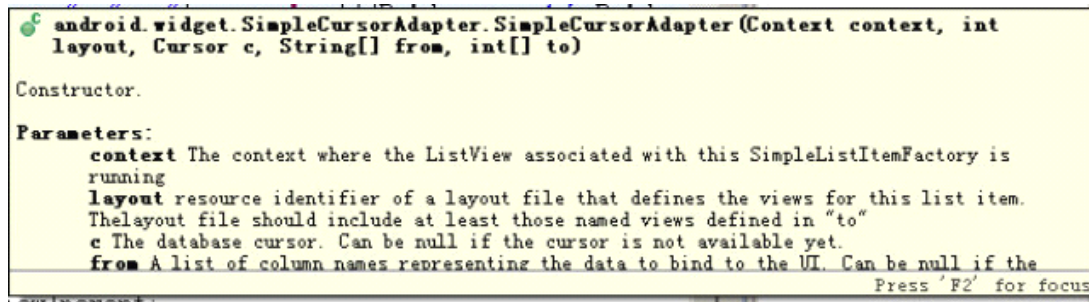


图3-42

第一个参数context：表示上下文。

第二个参数layout：表示资源的id。

第三个参数c：表示要绑定的游标。

第四个参数from：表示绑定的数据库中的列名。

第五个参数to：表示与数据字段名对应的TextView。

从常识来说，我们上述代码是没有问题的，但是运行会出现错误，如图3-43

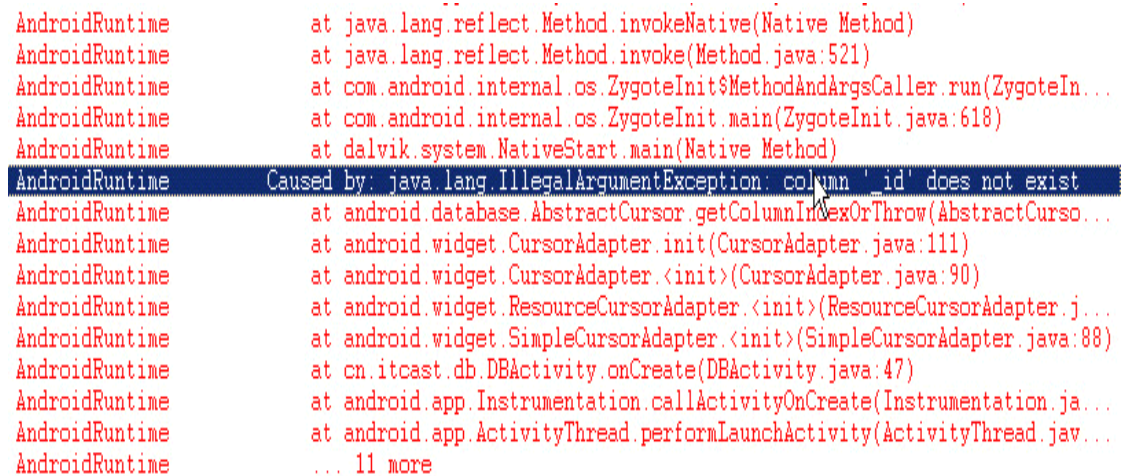


图3-43

错误的意思是：非法参数例外，列“_id”不存在。但是，我们并没有指定列名“_id”，怎么会错误提示不存在呢？原来在Android的文档中有一个规定，我们打开CursorAdapter的源码，就会发现有一段注释，如图3-44





```

/**
 * Adapter that exposes data from a {@link android.database.Cursor Cursor} to a
 * {@link android.widget.ListView ListView} widget. The Cursor must include
 * a column named " id" or this class will not work.
 */
public abstract class CursorAdapter extends BaseAdapter implements Filterable,
    CursorFilter.CursorFilterClient {
    /**
     * This field should be made private, so it is hidden from the SDK.
     * {@hide}
     */

```

图3-44

意思是说：这个游标必须要包含一个列名为“_id”，否则这个类将不会工作。这个是Android的规定，这一点必须要特别注意。还有一点也要注意，“_id”不可以使用大写。

我们为了符合这项规定，把业务类PersonService的getRawScrollData()方法改为如下代码：

```

public Cursor getRawScrollData(int startResult, int maxResult) {
    List<Person> persons = new ArrayList<Person>();
    SQLiteDatabase database = databaseHelper.getWritableDatabase();
    return database.rawQuery("select personid as _id, name, age from person
    limit ?,?",
        new String[]{String.valueOf(startResult),
String.valueOf(maxResult)}); // 查询结果的"personid"列名改为 "_id"
}

```

运行本应用成功，显示结果和使用之前的一样，都如图3-41。

有关用ListView显示数据到现在就基本完成了。使用ListView你可以把数据直观的呈现给用户。

小安：呵呵，您今天讲了太多知识点，我得好好消化一下了。

大致：嗯，SQLite 数据库在你今后做应用开发的时候，一定是很重要的，是必须掌握的知识。

小安：必须的！

