android
developers

Android开发教程&笔记

android

文件存取编程基础

文件

- 文件可以用来存储比使用引用更大数量的数据
- Android 提供方法来读、写文件
- 只有本地文件可以被访问
- 优点：可以存储大容量的数据
- 缺点：文件更新或是格式改变可能会导致巨大的编程工作

文件操作

读文件

- Context.openFileInput(String name)打开一个与应用程序联系的私有文件输入流
- 当文件不存在时抛出 FileNotFoundException 异常

```
FileInputStream in = this.openFileInput("test2.txt");//打开文件"test2.txt"
……
in.close();//关闭输入流
```

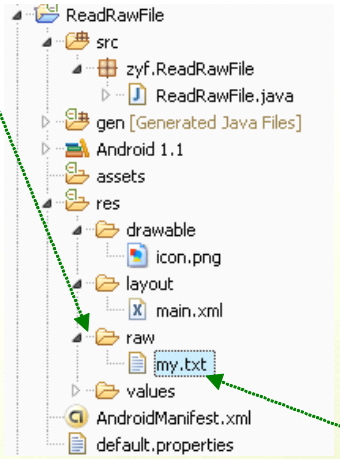
写文件

- Context.openFileOutput(String name,int mode)开启一个与应用程序联系的私有文件输出流
- 当文件不存在时该文件将被创建
- 文件输出流可以在添加模式中打开，这意味新的数据将被添加到文件的末尾

```
FileOutputStream out = this.openFileOutput("test2.txt",MODE_APPEND);
//打开文件"test2.txt"进行写操作 、使用 MODE_APPEND 在添加模式中打开文件
……
out.close();//关闭输出流
```

读取静态文件

- 要打开打包在应用程序中的静态文件，使用 Resources.openRawResource(R.raw.mydatafile)
- 该文件必须放在文件夹 res/raw/中




```
InputStreamin = this.getResources().openRawResource(R.raw.my);
…                          //获得Context资源
in.close();//关闭输入流
```

文件存取示例

创建添加文件内容并保存，打开文件并显示内容

① 新建工程 FileWriteRead

② 修改 main.xml 布局，添加一个 EditText、一个 Button

```
<?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"
    >
<EditText
    android:id="@+id/EditText_Txt"
    android:layout_height="350px"
    android:layout_width="fill_parent">
</EditText>
<Button
    android:layout_height="wrap_content"
    android:id="@+id/Button_Save"
    android:text="保存"
    android:layout_width="80px">
</Button>
</LinearLayout>
```

③ 在 res/layout 中新建一个 open.xml 布局文件，添加一个 TextView、两个 Button("打开","清空")

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
    android:id="@+id/openlayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
<TextView
    android:id="@+id/TextView_showTxt"
    android:layout_width="314px"
    android:layout_height="373px"
    android:layout_x="3px"
    android:layout_y="3px">
</TextView>
<Button
    android:id="@+id/Button_openTxt"
```

```
    android:layout_width="80px"
    android:layout_height="wrap_content"
    android:text="打开"
    android:layout_x="2px"
    android:layout_y="378px">
</Button>
<Button
    android:id="@+id/Button_clean"
    android:layout_width="80px"
    android:layout_height="wrap_content"
    android:text="清空"
    android:layout_x="239px"
    android:layout_y="378px">
</Button>
</AbsoluteLayout>
```

④ 文件存储

```
/*定义IO对象*/
private String Text_of_input;
private OutputStream os;
Text_of_input = inputArt.getText().toString();
//得到用户输入字符
        try {
            os = this.openFileOutput("txtME", MODE_PRIVATE);
            //打开一个文件输出流。名称为txtME，模式为不覆盖
            os.write(Text_of_input.getBytes());
            //把内容写入文件
        } catch (FileNotFoundException e) {
            // TODO Auto-generated catch block
        } catch (IOException e) {
            // TODO Auto-generated catch block
        } finally {
            try {
                //关闭文件输出流
                os.close();
            } catch (IOException e) {
                // TODO Auto-generated catch block
            }
        }
```

⑤ 文件读取

```
/*定义IO对象*/
private String Text_of_output;
private InputStream is;
```

```
    private byte[] b;
            try {

                //打开一个文件输入流。名称为txtME
                is = this.openFileInput("txtME");
                //字节数组声明定义
                b = new byte[1024];
                //读取文件内容放入字节数组
                int length = is.read(b);
                //把字节数组转换成字符串
                Text_of_output = new String(b);
                //显示读取内容长度
                setTitle("文件字数: " + length);
                //显示读取的文件内容
                showmyText.setText(Text_of_output);
            } catch (FileNotFoundException e) {
                // TODO Auto-generated catch block
            } catch (IOException e) {
                // TODO Auto-generated catch block
            }finally {
                try {
                    //关闭文件输入流
                    is.close();
                } catch (IOException e) {
                    // TODO Auto-generated catch block
                }
            }
```

⑥ 修改 mianActivity.java 文件，添加 menu 菜单与操作

```
package zyf.FileWrite;
/*导入要使用的包*/
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
```

```
public class FileWrite extends Activity implements Button.OnClickListener {
    /** Called when the activity is first created. */
    /*要使用的对象、变量声明*/
    /*保存部分*/
    private EditText inputArt;/*编辑框，输入用户字符串*/
    private Button saveButton;/*按钮，保存*/
    private String Text_of_input;/*字符串，用户输入的字符串*/
    private OutputStream os;/*文件输出流，保存文件流*/
    /*读取部分*/
    private TextView showmyText;/*TextView，显示读取文件内容*/
    private Button openTxt,cleanTxt;/*按钮，打开文件*/
    private String Text_of_output;/*字符串，从文件中读取到得 字符串*/
    private InputStream is;/*文件输入流，读取文件流*/
    private byte[] b;/*字节数组，用来读取文件内容*/

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setLayoutShow(R.layout.main);/*设置主屏布局*/
        UIinit("main");/*初始化UI元素方法*/
        Logic("main");/*添加事件逻辑方法*/
    }
    /*设置主屏*/
    private void setLayoutShow(int layoutID) {
        // TODO Auto-generated method stub
        setContentView(layoutID);/*设置当前主屏布局*/
    }
    private void UIinit(String mainROopen) {
        /*初始化UI*/
        if (mainROopen.equals("main")) {
            inputArt = (EditText) findViewById(R.id.EditText_Txt);
            saveButton = (Button) findViewById(R.id.Button_Save);
        } else if (mainROopen.equals("open")) {
            showmyText = (TextView) findViewById(R.id.TextView_showTxt);
            openTxt = (Button) findViewById(R.id.Button_openTxt);
            cleanTxt=(Button)findViewById(R.id.Button_clean);
        }
    }
    private void Logic(String string) {
        // TODO Auto-generated method stub
        /*为按钮添加事件处理*/
        if (string.equals("main")) {
            saveButton.setOnClickListener(this);
```

```
    } else if (string.equals("open")) {
        openTxt.setOnClickListener(this);
        cleanTxt.setOnClickListener(this);
    }
}

@Override
public void onClick(View v) {
    /*根据ID判断按钮事件*/
    switch (v.getId()) {
    case R.id.Button_Save: {
        /*提示*/
        NoteDebug("文件保存");
        // TODO Auto-generated method stub
        /*获得用户输入的字符串*/
        Text_of_input = inputArt.getText().toString();
        try {
            /*打开文件输出流，名称txtME，以不覆盖模式打开*/
            os = this.openFileOutput("txtME", MODE_PRIVATE);
            /*把字符串转换成字节数组，写入文件中*/
            os.write(Text_of_input.getBytes());
        } catch (FileNotFoundException e) {
            /*文件未找到，异常*/
            // TODO Auto-generated catch block
            NoteDebug("文件关闭失败" + e);
        } catch (IOException e) {
            /*文件写入错误*/
            // TODO Auto-generated catch block
            NoteDebug("文件写入失败" + e);
        } finally {
            try {
                /*关闭文件输出流*/
                os.close();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                NoteDebug("文件关闭失败" + e);
            }
        }
        /*输入框清空*/
        inputArt.setText("");
    }
        break;
    case R.id.Button_openTxt: {
        NoteDebug("文件打开");
```

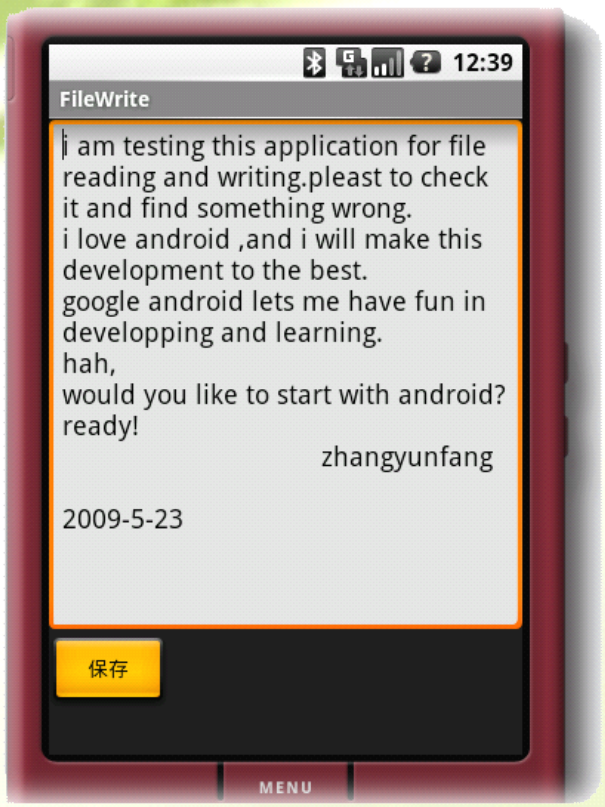
```
            try {
                /*打开文件输入流，名称txtME*/
                is = this.openFileInput("txtME");
                /*初始化字节数组*/
                b = new byte[1024];
                /*从文件输入流中读取内容到字节数组中，返回内容长度*/
                int length = is.read(b);
                /*把字节数组转换成字符串*/
                Text_of_output = new String(b);
                /*设置标题，显示文件内容长度*/
                setTitle("文件字数: " + length);
                /*显示文件内容*/
                showmyText.setText(Text_of_output);
            } catch (FileNotFoundException e) {
                /*文件未找到，异常*/
                // TODO Auto-generated catch block
                NoteDebug("文件打开失败" + e);
            } catch (IOException e) {
                /*文件读取错误，异常*/
                // TODO Auto-generated catch block
                NoteDebug("文件读取失败" + e);
            }

            finally {
                try {
                    /*关闭文件输入流*/
                    is.close();
                } catch (IOException e) {
                    // TODO Auto-generated catch block
                    NoteDebug( "文件关闭失败"+e);
                }
            }
        }
            break;
        case R.id.Button_clean:{
            /*清空*/
            showmyText.setText("");
            NoteDebug("清空");
        }
            break;
        default:
            break;
        }
    }
```

```
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // TODO Auto-generated method stub
        /*添加三个菜单项目，并设置图片*/
        menu.add(0, 1, 1, "Edit").setIcon(R.drawable.ic_menu_edit);
        menu.add(0, 2, 2, "Open").setIcon(R.drawable.ic_menu_agenda);
        menu.add(0, 3, 3, "Exit").setIcon(R.drawable.ic_lock_power_off);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // TODO Auto-generated method stub
        switch (item.getItemId()) {
        case 1:
            /*显示main.xml为主屏布局*/
            setLayoutShow(R.layout.main);
            UIinit("main");
            Logic("main");
            NoteDebug("编辑文件Layout");
            break;
        case 2:
            /*显示open.xml为主屏布局*/
            setLayoutShow(R.layout.open);
            UIinit("open");
            Logic("open");
            NoteDebug( "打开文件Layout");
            break;
        case 3:
            /*退出*/
            finish();
            NoteDebug( "Byebye");
            break;
        default:
            break;
        }
        return super.onOptionsItemSelected(item);
    }
    private void NoteDebug(String showString){
        /*显示Toast提示*/
        Toast.makeText(this,showString, Toast.LENGTH_SHORT).show();
    }
}
```

⑦ 结果


12:39
FileWrite
i am testing this application for file reading and writing.pleast to check it and find something wrong.
i love android ,and i will make this development to the best.
google android lets me have fun in developping and learning.
hah,
would you like to start with android?
ready!
zhangyunfang
2009-5-23
保存
MENU


12:39
FileWrite
Edit
Open
Exit
MENU


12:39
文件字数：370
I am testing this application for file reading and writing.pleast to check it and find something wrong.
i love android ,and i will make this development to the best.
google android lets me have fun in developping and learning.
hah,
would you like to start with android?
ready!
zhangyunfang
2009-5-23
文件打开
打开
清空
MENU


12:39
文件字数：370
清空
打开
清空
MENU


12:39
文件字数：370
Edit
Open
Exit
MENU

数据库编程基础

SQLite 数据库

- 在某些情况下，文件不是有效的
  - 如果多线程数据访问是相关的
  - 如果应用程序处理可能变化的复杂数据结构
  - 等等
- 因此，Android 带来了内置 SQLite 数据库支持
- 数据库对于创建它们的包套件是私有的
- 数据库不应该用来存贮文件
- 提示：在 SDK 中的 samples/NotePad 下可以找到关于如何使用数据库的例子
- SQLite 是一个轻量级的软件库
- 实现了一个完全适应严峻环境的数据库
  - 原子量性
  - 坚固性
  - 独立性
  - 耐久性
- 体积大小只用几千字节
- 一些 SQL 的指令只是部分支持，例如：ALTER、TABLE
- 参阅 http://www.sqlite.org 获取更多信息

创建数据库

- Context.createDatabase(String name,int version ,int mode,CursorFactory factory)创建一个新的数据库并返回一个 SQLiteDatabase 对象
- 假如数据库不能被创建，则抛出 FileNotFoundException 异常
- 新创建 SQLite 数据库方法

```
SQLiteDatabase mydataBase=SQLiteDatabase.create(new CursorFactory(){
//创建一个数据库
//工厂类，一个可选工厂类，当查询时调用来实例化一个光标
    @Override
    public Cursor newCursor(SQLiteDatabase db,
            SQLiteCursorDriver masterQuery, String editTable,
            SQLiteQuery query) {
        // TODO Auto-generated method stub
        return null;
    }

});
```

```
SQLiteDatabase myDataBase=this.openOrCreateDatabase("myDataBase.db",
                                                    MODE_PRIVATE, new CursorFactory(){
//创建新的数据库，名称myDatabase，模式MODE_PRIVATE，鼠标工厂
//工厂类，一个可选工厂类，当查询时调用来实例化一个光标
    @Override
    public Cursor newCursor(SQLiteDatabase db,
            SQLiteCursorDriver masterQuery, String editTable,
            SQLiteQuery query) {
        // TODO Auto-generated method stub
        return null;
    }

});
```

删除数据库

- Context.deleteDatabase(String name)删除指定名称的数据库
- 假如数据库成功删除则返回 true，失败则为 false(例如数据库不存在)

```
//删除指定名称的数据库
this.deleteDatabase("myDatabase.db");
```

打开数据库

- Context.openDatabase(String file,CursorFactory factory) 打开一个存在的数据库并返回一个 SQLiteDatabase 对象
- 如果数据库不存在则抛出 FileNotFoundException 异常

```
//创建一个名为：myDataBase的数据库，后缀为.db
SQLiteDatabase my_DataBase=this.openOrCreateDatabase("myDateBase.db",
                                                    MODE_PRIVATE, null);
my_DataBase.close();//不要忘记关闭数据库
```

非查询 SQL 指令

- SQLiteDatabase.execSQL(String sql)可以用来执行非查询 SQL 指令，这些指令没有结果
- 包括：CREATE TABLE / DROP TABLE / INSERT 等等
- 例如：

① 创建一个名为"test"并带两个参数的表

```
//创建一个名为"test"并带两个参数的表
my_DataBase.execSQL("CREATE TABLE test (_id INTEGER PRIMARY KEY,
                                        someNumber INTERGER);");
```

② 在数据库中插入一个元组

```
//在数据库中插入一个元组
my_DataBase.execSQL("INSERT INTO test (_id,someNumber) values(1,8);");
```

③ 删除表

```
//删除表
my_DataBase.execSQL("DROP TABLE test");
```

查询 SQL 指令-游标 Cursors

- Android 使用游标(Cursors)来导航浏览查询结果
- 游标(Cursors)被 android.database.Cursor 对象来描述
- 一个游标(Cursors)是一个简单的指针，它从查询结果的一个元组跳到下一个元组(或是前一个或是第一个或是……)
- 游标(Cursors)在它定位位置的那一刻返回元组数据

表 test

	_id	someNumber
▶	1	8
	2	10
	3	2

```
//为了创建一个Cursor(游标)，必须执行一个查询，要么通过SQL使用rawQuery()方法
//或是更精心设计的方法，像query()方法
Cursor cur=my_DataBase.rawQuery("SELECT * FORM test", null);
if(cur!=null){//游标不为空
    //返回给定名称的列的基于0开始的index，如果该属性列不存在则返回-1
    //通过它们的index来检索属性值
    int numColumn=cur.getColumnIndex("someNumber");
    if(cur.moveToFirst()){
    //cur.moveToFirst()让游标指向第一行，如果游标指向第一行，则返回true
        do {
            int num=cur.getInt(numColumn);//获得当前行该属性的值
            /*Cursor提供了不同的方法来回索不同的数据类型
            例如getInt(int index)/getString(int index)等等*/
            /*做一些事情*/
        } while (cur.moveToNext());
        /*游标移动到下一行，如果游标已经通过了结果集中的最后，
          即没有行可以移动时，则返回false*/
        //其他可能移动的是 previous() 和first()方法
    }
}
```

android
developers

android