

[首页](#) [ANDROID](#) [互联网](#) [杂乱无章](#) [科技资讯](#) [程序员人生](#) [程序员笑话](#) [编程技术](#) [网址导航](#)

6.1 数据存储与访问之——文件存储读写

分类 [Android 基础入门教程](#)

Android 基础入门教程(Q群号 : 153836263)

- 1.0 Android基础入门教程
 - 1.0.1 2015年最新Android基...
- 1.1 背景相关与系统架构分析
- 1.2 开发环境搭建
 - 1.2.1 使用Eclipse + ADT + S...
 - 1.2.2 使用Android Studio开...
- 1.3 SDK更新不了问题解决
- 1.4 Genymotion模拟器安装
- 1.5.1 Git使用教程之本地仓...
- 1.5.2 Git之使用GitHub搭建...
- 1.6 .9(九妹)图片怎么玩
- 1.7 界面原型设计
- 1.8 工程相关解析(各种文件...
- 1.9 Android程序签名打包
- 1.11 反编译APK获取代码&...
- 2.1 View与ViewGroup的概念
 - 2.2.1 LinearLayout(线性布局)
 - 2.2.2 RelativeLayout(相对布...
 - 2.2.3 TableLayout(表格布局)
 - 2.2.4 FrameLayout(帧布局)
 - 2.2.5 GridLayout(网格布局)
 - 2.2.6 AbsoluteLayout(绝对...
- 2.3.1 TextView(文本框)详解
- 2.3.2 EditText(输入框)详解
- 2.3.3 Button(按钮)与ImageB...
- 2.3.4 ImageView(图像视图)
- 2.3.5. RadioButton(单选按钮...
- 2.3.6 开关按钮ToggleButton...
- 2.3.7 ProgressBar(进度条)
- 2.3.8 SeekBar(拖动条)
- 2.3.9 RatingBar(星级评分条)
- 2.4.1 ScrollView(滚动条)

本节引言：

嘿嘿，看到这个题目，相信部分读者会问，你前面的Fragment写完了吗？嗯，没写完，因为想例子，需要一点时间，为了提高效率，所以决定像多线程一样，并发的来写教程，这样可能可以加快写教程的进度，到现在为止，刚好写了60篇，离完成入门教程还很远呢，而前面也说过，想在一个半到两个月之内完成这套教程，今天已经9.1号了，要加把劲~好的，废话就这么多，本节给大家介绍的是Android数据存储与访问方式中的一个——文件存储与读写，当然除了这种方式外，我们可以存到SharedPreference，数据库，或者Application中，当然这些后面都会讲，嗯，开始本节内容~

1.Android文件的操作模式

学过Java的同学都知道，我们新建文件，然后就可以写入数据了，但是Android却不一样，因为Android是基于Linux的，我们在读写文件的时候，还需加上文件的操作模式，Android中的操作模式如下：



图1：文件的操作模式

2.文件的相关操作方法

<code>openFileOutput(filename,mode)</code>	打开文件输出流,就是往文件中写入数据,第二个参数是模式
<code>openFileInput(filename)</code>	打开文件输入流,就是读取文件中的信息到程序中
<code>getDir(name,mode)</code>	在app的data目录下获取或创建name对应的子目录
<code>getFileDir()</code>	获得app的data目录的file目录的绝对路径
<code>String[] fileList()</code>	返回app的data目录下的全部文件
<code>deleteFile(filename)</code>	删除app的data目录下的指定文件

ps:Android有自己的一套安全模型,当我们安装apk时,系统会分配给他一个userid,当该应用要去访问其他资源,比如文件时就需要匹配userid,任何app创建的文件,sharedpreferences,数据库文件都是私有的,其他程序无法访问;除非在创建时指定了Context.MODE_WORLD_READABLE或者Context.MODE_WORLD_WRITEABLE ,只有这样其他程序才能正确访问。

图2：文件的相关操作方法

3.文件读写的实现

Android中的文件读写和Java中的文件I/O相同，流程也很简单，下面我们来写个简单的示例：

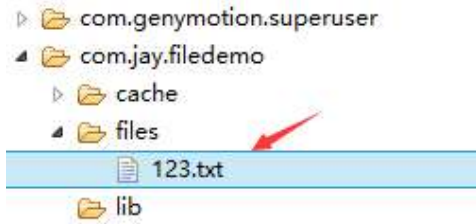
实现效果图：



PS:这里用的是模拟器，因为笔者的N5并没有root，看不到文件的存储目录，

- 2.4.2 Date & Time组件(上)
- 2.4.3 Date & Time组件(下)
- 2.4.4 Adapter基础讲解
- 2.4.5 ListView简单实用
- 2.4.6 BaseAdapter优化
- 2.4.7ListView的焦点问题
- 2.4.8 ListView之checkbox错...
- 2.4.9 ListView的数据更新问题
- 2.5.0 构建一个可复用的自定...
- 2.5.1 ListView Item多布局的...
- 2.5.2 GridView(网格视图)的...
- 2.5.3 Spinner(列表选项框)...
- 2.5.4 AutoCompleteTextVie...
- 2.5.5 ExpandableListView(...
- 2.5.6 ViewFlipper(翻转视图)...
- 2.5.7 Toast(吐司)的基本使用
- 2.5.8 Notification(状态栏通...
- 2.5.9 AlertDialog(对话框)详解
- 2.6.0 其他几种常用对话框基...
- 2.6.1 PopupWindow(悬浮框...
- 2.6.2 菜单(Menu)
- 2.6.3 ViewPager的简单使用
- 2.6.4 DrawerLayout(官方侧...
- 3.1.1 基于监听的事件处理机制
- 3.2 基于回调的事件处理机制
- 3.3 Handler消息传递机制浅析
- 3.4 TouchListener PK OnTo...
- 3.5 监听EditText的内容变化
- 3.6 响应系统设置的事件(Co...
- 3.7 AsyncTask异步任务
- 3.8 Gestures(手势)
- 4.1.1 Activity初学乍练
- 4.1.2 Activity初窥门径
- 4.1.3 Activity登堂入室
- 4.2.1 Service初涉
- 4.2.2 Service进阶
- 4.2.3 Service精通
- 4.3.1 BroadcastReceiver牛...
- 4.3.2 BroadcastReceiver庖...
- 4.4.1 ContentProvider初探

下面我们打开DDMS 的File Exploer可以看到，在data/data/<包名>/file中有我们写入的文件：



我们可以点击右上角的响应图标将文件导入到电脑中，并且打开验证写入的内容：



代码实现：

首先是布局文件:main_activity.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.jay.example.filedemo1.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/nametitle" />

    <EditText
        android:id="@+id/editname"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/detailtitle" />

    <EditText
        android:id="@+id/editdetail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minLines="2" />
```

4.4.2 ContentProvider再探...

4.5.1 Intent的基本使用

4.5.2 Intent之复杂数据的传递

5.1 Fragment基本概述

5.2.1 Fragment实例精讲—...

5.2.2 Fragment实例精讲—...

5.2.3 Fragment实例精讲—...

5.2.4 Fragment实例精讲—...

5.2.5 Fragment实例精讲—...

6.1 数据存储与访问之——文...

6.2 数据存储与访问之——S...

6.3.1 数据存储与访问之——...

6.3.2 数据存储与访问之——...

7.1.1 Android网络编程要学...

7.1.2 Android Http请求头与...

7.1.3 Android HTTP请求方...

7.1.4 Android HTTP请求方...

7.2.1 Android XML数据解析

7.2.2 Android JSON数据解析

7.3.1 Android 文件上传

7.3.2 Android 文件下载 (1)

7.3.3 Android 文件下载 (2)

7.4 Android 调用 WebService

7.5.1 WebView(网页视图)基...

7.5.2 WebView和JavaScip...

7.5.3 Android 4.4后WebVie...

7.5.4 WebView文件下载

7.5.5 WebView缓存问题

7.5.6 WebView处理网页返...

7.6.1 Socket学习网络基础准备

7.6.2 基于TCP协议的Socket...

7.6.3 基于TCP协议的Socket...

7.6.4 基于UDP协议的Socke...

8.1.1 Android中的13种Draw...

8.1.2 Android中的13种Draw...

8.1.3 Android中的13种Draw...

8.2.1 Bitmap(位图)全解析 P...

8.2.2 Bitmap引起的OOM问题

8.3.1 三个绘图工具类详解

8.3.2 绘图类实战示例

```

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
        android:id="@+id/btnsave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/btnwrite" />

    <Button
        android:id="@+id/btnclean"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/btnclean" />

</LinearLayout>

<Button
    android:id="@+id/btnread"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/btnread" />

</LinearLayout>

```

然后我们来写一个文件协助类：**FileHelper.java**

```

/**
 * Created by Jay on 2015/9/1 0001.
 */
public class FileHelper {

    private Context mContext;

    public FileHelper() {
    }

    public FileHelper(Context mContext) {
        super();
        this.mContext = mContext;
    }

    /**
     * 这里定义的是一个文件保存的方法，写入到文件中，所以是输出流
     */
    public void save(String filename, String filecontent) throws Exception {
        //这里我们使用私有模式,创建出来的文件只能被本应用访问,还会覆盖原文件哦
        FileOutputStream output = mContext.openFileOutput(filename, Context.MODE_PRIVATE);
        output.write(filecontent.getBytes()); //将String字符串

```

8.3.3 Paint API之—— Mask...

8.3.4 Paint API之—— Xferm...

8.3.5 Paint API之—— Xferm...

8.3.6 Paint API之—— Xferm...

8.3.7 Paint API之—— Xferm...

8.3.8 Paint API之—— Xferm...

8.3.9 Paint API之—— Color...

8.3.10 Paint API之—— Colo...

8.3.11 Paint API之—— Colo...

8.3.12 Paint API之—— Path...

8.3.13 Paint API之—— Sha...

8.3.14 Paint几个枚举/常量值...

8.3.15 Paint API之——Type...

8.3.16 Canvas API详解(Part 1)

8.3.17 Canvas API详解(Part...

8.3.18 Canvas API详解(Part...

8.4.1 Android动画合集之帧...

8.4.2 Android动画合集之补...

8.4.3 Android动画合集之属...

8.4.4 Android动画合集之属...

9.1 使用SoundPool播放音...

9.2 MediaPlayer播放音频与...

9.3 使用Camera拍照

9.4 使用MediaRecord录音

10.1 TelephonyManager(电...

10.2 SmsManager(短信管理...

10.3 AudioManager(音频管...

10.4 Vibrator(振动器)

10.5 AlarmManager(闹钟服务)

10.6 PowerManager(电源服...

10.7 WindowManager(窗口...

10.8 LayoutInflater(布局服务)

10.9 WallpaperManager(壁...

10.10 传感器专题(1)——相...

10.11 传感器专题(2)——方...

10.12 传感器专题(3)——加...

10.12 传感器专题(4)——其...

10.14 Android GPS初涉

11.0 《2015最新Android基...

以字节流的形式写入到输出流中

```
        output.close();           //关闭输出流
    }

    /*
    * 这里定义的是文件读取的方法
    * */
    public String read(String filename) throws IOException {
        //打开文件输入流
        FileInputStream input = mContext.openFileInput(filename);

        byte[] temp = new byte[1024];
        StringBuilder sb = new StringBuilder("");
        int len = 0;
        //读取文件内容:
        while ((len = input.read(temp)) > 0) {
            sb.append(new String(temp, 0, len));
        }
        //关闭输入流
        input.close();
        return sb.toString();
    }
}
```

最后是MainActivity.java，我们在这里完成相关操作：

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    private EditText editname;
    private EditText editdetail;
    private Button btnsave;
    private Button btnclean;
    private Button btnread;
    private Context mContext;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mContext = getApplicationContext();
        bindViews();
    }

    private void bindViews() {
        editdetail = (EditText) findViewById(R.id.editdetail);
        editname = (EditText) findViewById(R.id.editname);
        btnclean = (Button) findViewById(R.id.btnclean);
        btnsave = (Button) findViewById(R.id.btnsave);
        btnread = (Button) findViewById(R.id.btnread);
    }
}
```

```

        btnclean.setOnClickListener(this);
        btnsave.setOnClickListener(this);
        btnread.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btnclean:
                editdetail.setText("");
                editname.setText("");
                break;
            case R.id.btnsave:
                FileHelper fHelper = new FileHelper(mContext);
                String filename = editname.getText().toString();

                String filedetail = editdetail.getText().toString();

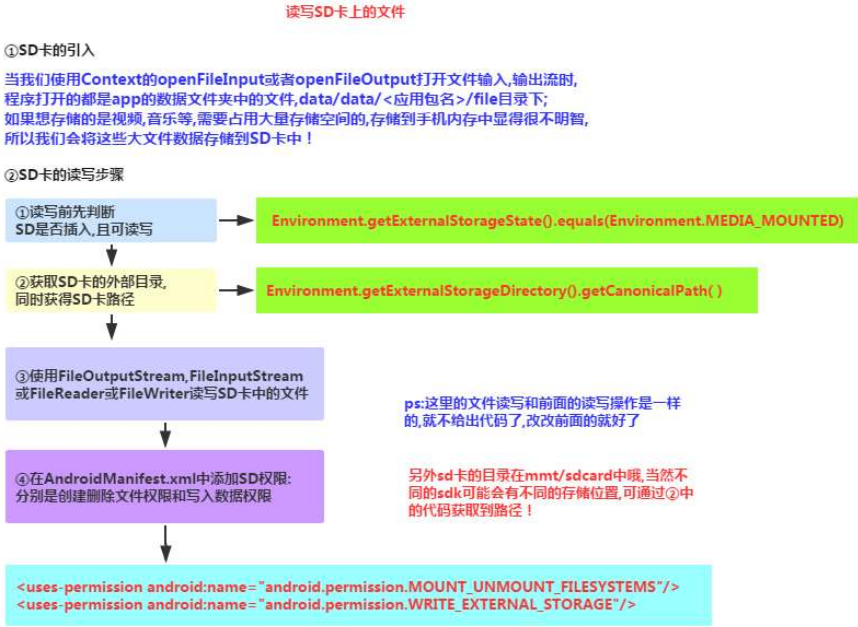
                try {
                    fHelper.save(filename, filedetail);
                    Toast.makeText(getApplicationContext(), "数据写入成功", Toast.LENGTH_SHORT).show();
                } catch (Exception e) {
                    e.printStackTrace();
                    Toast.makeText(getApplicationContext(), "数据写入失败", Toast.LENGTH_SHORT).show();
                }
                break;
            case R.id.btnread:
                String detail = "";
                FileHelper fHelper2 = new FileHelper(getApplicationContext());
                try {
                    String fname = editname.getText().toString();

                    detail = fHelper2.read(fname);
                } catch (IOException e) {
                    e.printStackTrace();
                }
                Toast.makeText(getApplicationContext(), detail, Toast.LENGTH_SHORT).show();
                break;
        }
    }
}

```

4.读取SD卡上的文件

读取流程图：



代码示例：

运行效果图：



同样打开DDMS的File Explorer，在旧版本的系统上我们可以直接在mmt\sdcard上找到，但是新版本 的就可能需要我们自己找找了，首先我们来到这个路径下：

mnt	2015-09-01	09:06	drwxrwxr-x	
USB	2015-09-01	09:06	d-----	
asec	2015-09-01	09:06	drwxr-xr-x	
obb	2015-09-01	09:06	drwxr-xr-x	
sdcard	2015-09-01	09:06	lrwxrwxrwx	-> /storage/emulated/legacy

点开sdcard，但是没东西，我们继续找唠叨后面这

个/storage/emulated/legacy下找：

storage	2015-09-01	09:06	d---r-x---	
emulated	2015-09-01	09:06	dr-xr-xr-x	-> /mnt/shell/emulated/0
legacy	2015-09-01	09:06	lrwxrwxrwx	-> /storage/emulated/legacy
sdcard0	2015-09-01	09:06	lrwxrwxrwx	

好吧，他又跳到别的地方去了，我们继续找/storage/shell/emulated/0

shell	2015-09-01	09:06	drwx-----	
emulated	1970-01-01	00:00	drwxrwxrwx	
0	2015-09-01	11:05	drwxrwxrwx	
Alarms	2015-08-04	02:17	drwxrwxrwx	
Android	2015-08-27	07:18	drwxrwxrwx	
DCIM	2015-08-04	02:17	drwxrwxrwx	
Download	2015-08-27	07:16	drwxrwxrwx	
Movies	2015-08-04	02:17	drwxrwxrwx	
Music	2015-08-04	02:17	drwxrwxrwx	
Notifications	2015-08-04	02:17	drwxrwxrwx	
Pictures	2015-08-27	07:18	drwxrwxrwx	
Podcasts	2015-08-04	02:17	drwxrwxrwx	
Ringtones	2015-08-04	02:17	drwxrwxrwx	
Tencent	2015-08-27	07:18	drwxrwxrwx	
baidu	2015-08-27	07:19	drwxrwxrwx	
test.txt	2015-09-01	11:05	-rwxrwxrwx	
xtuone	2015-08-27	07:18	drwxrwxrwx	
legacy	2015-08-04	02:17	drwxrwxrwx	
obb	2015-08-04	02:17	drwxrwxrwx	

果然找到了，我们在SD卡里生成的test.txt！导出到电脑看下里面的内容：



嘿嘿，果然读写SD卡成功~接下来我们看下代码是怎么写的：

代码实现：

main_activity.xml:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.jay.example.filedemo2.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="请输入文件名" />

    <EditText
        android:id="@+id/edittitle"
```



```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="文件名" />
```

<TextView

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="请输入文件内容" />
```

<EditText

```
android:id="@+id/editdetail"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="文件内容" />
```

<Button

```
android:id="@+id/btnsave"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="保存到SD卡" />
```

<Button

```
android:id="@+id/btnclean"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="清空" />
```

<Button

```
android:id="@+id/btnread"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="读取sd卡中的文件" />
```

```
</LinearLayout>
```

接着我们来写一个SD操作类：**SDFileHelper.java**

```
/**
 * Created by Jay on 2015/9/1 0001.
 */
public class SDFileHelper {

    private Context context;

    public SDFileHelper() {
    }

    public SDFileHelper(Context context) {
        super();
        this.context = context;
    }
}
```

```

//往SD卡写入文件的方法
public void saveFileToSD(String filename, String filecontent) throws Exception {
    //如果手机已插入sd卡,且app具有读写sd卡的权限
    if (Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED)) {
        filename = Environment.getExternalStorageDirectory().getCanonicalPath() + "/" + filename;
        //这里就不要用openFileOutput了,那个是往手机内存中写数据的

        FileOutputStream output = new FileOutputStream(filename);

        output.write(filecontent.getBytes());
        //将String字符串以字节流的形式写入到输出流中
        output.close();
        //关闭输出流
    } else Toast.makeText(context, "SD卡不存在或者不可读写", Toast.LENGTH_SHORT).show();
}

//读取SD卡中文件的方法
//定义读取文件的方法:
public String readFromSD(String filename) throws IOException {
    StringBuilder sb = new StringBuilder("");
    if (Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED)) {
        filename = Environment.getExternalStorageDirectory().getCanonicalPath() + "/" + filename;
        //打开文件输入流
        FileInputStream input = new FileInputStream(filename);

        byte[] temp = new byte[1024];

        int len = 0;
        //读取文件内容:
        while ((len = input.read(temp)) > 0) {
            sb.append(new String(temp, 0, len));
        }
        //关闭输入流
        input.close();
    }
    return sb.toString();
}
}

```

接着MainActivity.java实现相关逻辑：

```

public class MainActivity extends AppCompatActivity implements
View.OnClickListener{

```

```

private EditText editname;
private EditText editdetail;
private Button btnsave;
private Button btnclean;
private Button btnread;
private Context mContext;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mContext = getApplicationContext();
    bindViews();
}

private void bindViews() {
    editname = (EditText) findViewById(R.id.edittitle);
    editdetail = (EditText) findViewById(R.id.editdetail);
    btnsave = (Button) findViewById(R.id.btnsave);
    btnclean = (Button) findViewById(R.id.btnclean);
    btnread = (Button) findViewById(R.id.btnread);

    btnsave.setOnClickListener(this);
    btnclean.setOnClickListener(this);
    btnread.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btnclean:
            editdetail.setText("");
            editname.setText("");
            break;
        case R.id.btnsave:
            String filename = editname.getText().toString();
            String filedetail = editdetail.getText().toString();

            SDFileHelper sdHelper = new SDFileHelper(mContext);

            try {
                sdHelper.savaFileToSD(filename, filedetail);

                Toast.makeText(getApplicationContext(), "数据写入成功", Toast.LENGTH_SHORT).show();
            } catch (Exception e) {
                e.printStackTrace();
                Toast.makeText(getApplicationContext(), "数据写入失败", Toast.LENGTH_SHORT).show();
            }
            break;
    }
}

```

```

        case R.id.btnread:
            String detail = "";
            SdFileHelper sdHelper2 = new SdFileHelper(mContext);

            try
            {
                String filename2 = editname.getText().toString();

                detail = sdHelper2.readFromSD(filename2);
            }
            catch (IOException e) {e.printStackTrace();}

            Toast.makeText(getApplicationContext(), detail,
            Toast.LENGTH_SHORT).show();

            break;
        }
    }
}

```

最后别忘记在**AndroidManifest.xml**写上读写SD卡的权限哦！

```

<!-- 在SDCard中创建与删除文件权限 -->
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_
_FILESYSTEMS"/>
<!-- 往SDCard写入数据权限 -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_
_STORAGE"/>

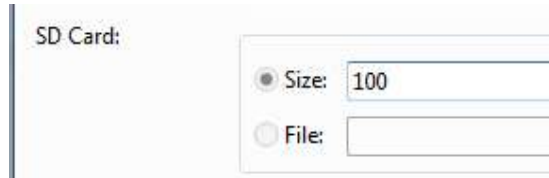
```

5.关于原生模拟器SD卡的问题

如果是真机调试的话通常都是可以的,对于原生虚拟机的话就问题多多了,再我们前面使用

`Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED)`

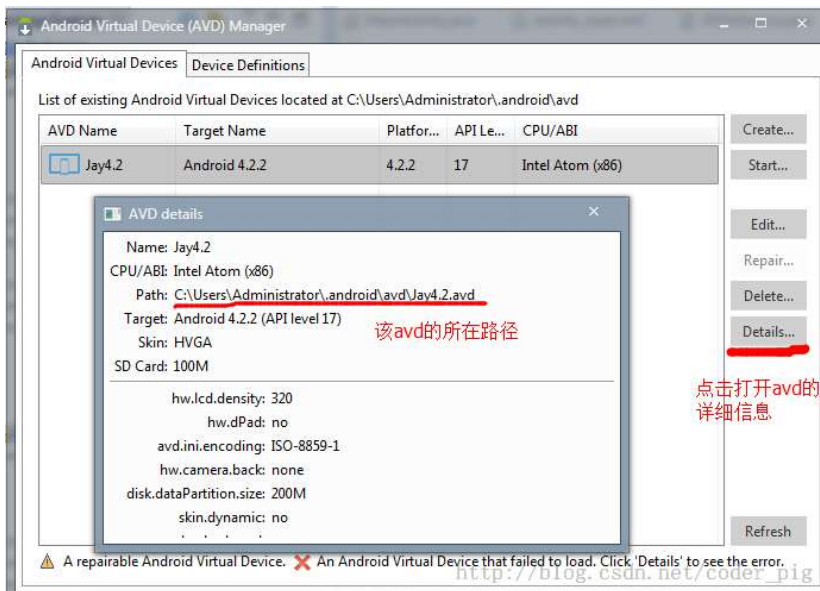
可能一直返回的是false,就是SD卡不存在,这个是主要的问题,现在新版本的SDK都会在 创建AVD的时候会同时申请一块SD卡的存储区域的



对于旧版本的sdk或者其他原因可能需要手动关联下sd卡,设置如下:

①找到创建好的avd的镜像的路径:

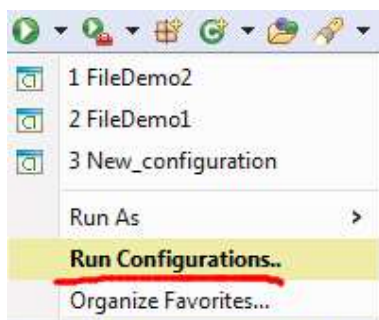
点击打开avd界面,点击detail,查看avd镜像的目录下



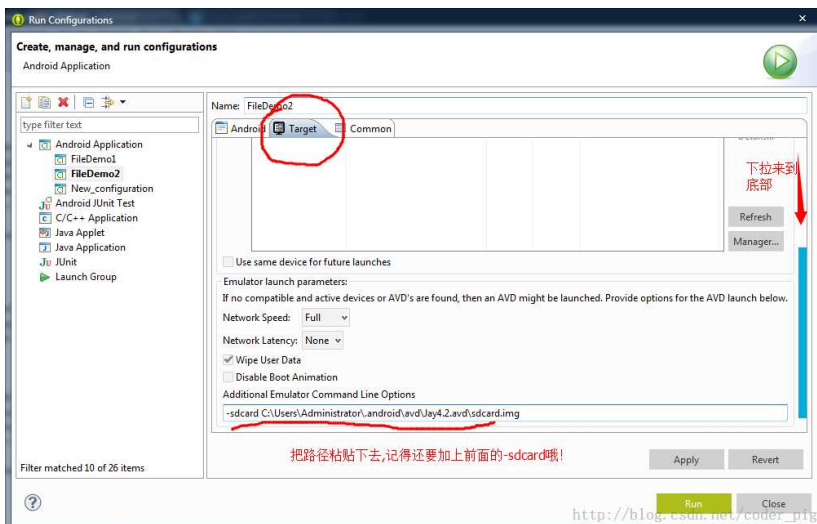
②来到avd镜像所在的路径下,复制sdcard.img的路径:

比如我的:-sdcard

C:\Users\Administrator.android\avd\Jay4.2.avd\sdcard.img



③接着点击 来到以下界面:



最后apply以下,然后Run就可以了!

6.读取raw和assets文件夹下的文件

相信大家对两个文件夹并不陌生,如果我们不想自己的文件被编译成二进制文件的话,我们可以把文件放到这两个目录下,而两者的区别如下:

res/raw: 文件会被映射到R.java文件中,访问的时候直接通过资源ID即可访问,而且他不能有目录结构,就是不能再创建文件夹

assets: 不会映射到R.java文件中,通过AssetManager来访问,能有目录结

构，即，可以自行创建文件夹

读取文件资源：

res/raw：

```
InputStream is =getResources().openRawResource(R.id.filename);
```

assets：

```
AssetManager am = getAssets();
InputStream is = am.open("filename");
```

代码下载：

- FileDemo.zip：[下载 FileDemo.zip](#)
- FileDemo2.zip：[下载 FileDemo2.zip](#)

本节小结：

好的，关于Android的数据存储与访问的第一节——文件读写就到这里，如果在学习本文中 遇到什么问题，或者觉得有些纰漏的地方，欢迎提出，万分感激，谢谢~



在线实例

- [HTML 实例](#)
- [CSS 实例](#)
- [JavaScript 实例](#)
- [Ajax 实例](#)
- [jQuery 实例](#)
- [XML 实例](#)
- [Java 实例](#)

字符集&工具

- [HTML 字符集设置](#)
- [HTML ASCII 字符集](#)
- [HTML ISO-8859-1](#)
- [HTML 实体符号](#)
- [HTML 拾色器](#)
- [JSON 格式化工具](#)

最新更新

- [Vue.js 入门教程](#)
- [CSS 下拉菜单](#)
- [WordPress Nginx...](#)
- [TIOBE 2015 年12...](#)
- [Swift 正式开源](#)
- [PHP 7 正式发布](#)
- [Shell 编程快速入门](#)

站点信息

- [意见反馈](#)
- [免责声明](#)
- [关于我们](#)
- [文章归档](#)

关注微信



Reserved. 备案号：闽ICP备
15012807号-1