

原 最简单的基于FFmpeg的移动端例子附件：Android 自带播放器

2015年07月25日 20:00:30 阅读数：22409

=====

最简单的基于FFmpeg的移动端例子系列文章列表：

[最简单的基于FFmpeg的移动端例子：Android HelloWorld](#)

[最简单的基于FFmpeg的移动端例子：Android 视频解码器](#)

[最简单的基于FFmpeg的移动端例子：Android 视频解码器-单个库版](#)

[最简单的基于FFmpeg的移动端例子：Android 推流器](#)

[最简单的基于FFmpeg的移动端例子：Android 视频转码器](#)

[最简单的基于FFmpeg的移动端例子附件：Android 自带播放器](#)

[最简单的基于FFmpeg的移动端例子附件：SDL Android HelloWorld](#)

[最简单的基于FFmpeg的移动端例子：IOS HelloWorld](#)

[最简单的基于FFmpeg的移动端例子：IOS 视频解码器](#)

[最简单的基于FFmpeg的移动端例子：IOS 推流器](#)

[最简单的基于FFmpeg的移动端例子：IOS 视频转码器](#)

[最简单的基于FFmpeg的移动端例子附件：IOS自带播放器](#)

[最简单的基于FFmpeg的移动端例子：Windows Phone HelloWorld](#)

=====

前几篇文章记录的都是安卓平台下基于FFmpeg的多媒体程序。作为对比，本文记录一个基于安卓自带多媒体框架OpenCore的视频播放器。

安卓自带的多媒体框架的使用方法相对移植FFmpeg来说简单太多了，只需要调用几个接口函数就能控制视频的播放。不再过多记录它的使用方法。

□

源代码

项目的目录结构如图所示。Java源代码位于src目录。

□

Android程序Java端代码位于src\com\leixiaohua1020\sandroidplayer\MainActivity.java，如下所示。

```
[java]  1.  /**
2.      *
3.      *  最简单的安卓视频播放器
4.      *  Simplest Android Player
5.      *
6.      *  雷霄骅 Lei Xiaohua
7.      *  leixiaohua1020@126.com
8.      *  中国传媒大学/数字电视技术
9.      *  Communication University of China / Digital TV Technology
10.     *  http://blog.csdn.net/leixiaohua1020
11.     *
12.     *  本程序是基于安卓接口的视频播放器。
13.     *
14.     *  Video player based on Android API.
15.     *
16.     */
17.     package com.leixiaohua1020.sandroidplayer;
18.
19.     import android.os.Bundle;
20.     import android.os.Environment;
21.     import android.app.Activity;
22.     import android.util.Log;
23.     import android.view.Menu;
24.     import android.view.View;
25.     import android.view.View.OnClickListener;
```

```

26. import android.widget.Button;
27. import android.widget.EditText;
28. import android.widget.VideoView;
29.
30. public class MainActivity extends Activity {
31.
32.     @Override
33.     protected void onCreate(Bundle savedInstanceState) {
34.         super.onCreate(savedInstanceState);
35.         setContentView(R.layout.activity_main);
36.
37.         final VideoView videoView = (VideoView) findViewById(R.id.videoView1);
38.
39.         Button setButton = (Button) this.findViewById(R.id.button_set);
40.         Button startButton = (Button) this.findViewById(R.id.button_start);
41.         Button pauseButton = (Button) this.findViewById(R.id.button_pause);
42.         Button stopButton = (Button) this.findViewById(R.id.button_stop);
43.         final EditText urlEditText= (EditText) this.findViewById(R.id.input_url);
44.
45.
46.         setButton.setOnClickListener(new OnClickListener() {
47.             public void onClick(View arg0){
48.                 String folderurl=Environment.getExternalStorageDirectory().getPath();
49.                 String urltext=urlEditText.getText().toString();
50.                 //Small FIX, Avoid '/'
51.                 if(urltext.charAt(0)=='/'){
52.                     urltext=urltext.substring(1);
53.                 }
54.                 String inputurl=folderurl+"/"+urltext;
55.                 Log.e("url",inputurl);
56.
57.                 videoView.setVideoPath(inputurl);
58.                 //videoView.setMediaController(new MediaController(MainActivity.this));
59.             }
60.         });
61.
62.         startButton.setOnClickListener(new OnClickListener() {
63.             public void onClick(View arg0){
64.                 videoView.start();
65.             }
66.         });
67.
68.         pauseButton.setOnClickListener(new OnClickListener() {
69.             public void onClick(View arg0){
70.                 videoView.pause();
71.             }
72.         });
73.
74.         stopButton.setOnClickListener(new OnClickListener() {
75.             public void onClick(View arg0){
76.                 videoView.stopPlayback();
77.             }
78.         });
79.
80.     }
81.
82.
83.     @Override
84.     public boolean onCreateOptionsMenu(Menu menu) {
85.         // Inflate the menu; this adds items to the action bar if it is present.
86.         getMenuInflater().inflate(R.menu.main, menu);
87.         return true;
88.     }
89.
90. }

```

运行结果

App在手机上运行后的结果如下图所示。单击“Set”、“Start”、“Pause”、“Stop”几个按钮可以对视频做不同的操作。

注意需要把等待播放的视频文件拷贝至存储卡相应的目录中。例如对于下面截图的情况，需要将sintel.mp4拷贝至存储卡的根目录中。

□

下载

simplest ffmpeg mobile

项目主页

Github：https://github.com/leixiaohua1020/simplest_ffmpeg_mobile

开源中国：https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_mobile

SourceForge：<https://sourceforge.net/projects/simplestffmpegmobile/>

CSDN工程下载地址：<http://download.csdn.net/detail/leixiaohua1020/8924391>

本解决方案包含了使用FFmpeg在移动端处理多媒体的各种例子：

[Android]

simplest_android_player: 基于安卓接口的视频播放器

simplest_ffmpeg_android_helloworld: 安卓平台下基于FFmpeg的HelloWorld程序

simplest_ffmpeg_android_decoder: 安卓平台下最简单的基于FFmpeg的视频解码器

simplest_ffmpeg_android_decoder_onelib: 安卓平台下最简单的基于FFmpeg的视频解码器-单库版

simplest_ffmpeg_android_streamer: 安卓平台下最简单的基于FFmpeg的推流器

simplest_ffmpeg_android_transcoder: 安卓平台下移植的FFmpeg命令行工具

simplest_sdl_android_helloworld: 移植SDL到安卓平台的最简单程序

[IOS]

simplest_ios_player: 基于IOS接口的视频播放器

simplest_ffmpeg_ios_helloworld: IOS平台下基于FFmpeg的HelloWorld程序

simplest_ffmpeg_ios_decoder: IOS平台下最简单的基于FFmpeg的视频解码器

simplest_ffmpeg_ios_streamer: IOS平台下最简单的基于FFmpeg的推流器

simplest_ffmpeg_ios_transcoder: IOS平台下移植的ffmpeg.c命令行工具

simplest_sdl_ios_helloworld: 移植SDL到IOS平台的最简单程序

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/47056733>

文章标签：[Android](#) [OpenCore](#) [视频播放器](#) [Java](#) [FFmpeg](#)

个人分类：[FFMPEG](#) [Android多媒体](#)

所属专栏：[FFmpeg](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushide@163.com