[**Android MediaPlayer和VideoView的使用**](http://blog.csdn.net/huaxun66/article/details/53333747)

2016-11-25 14:14 587人阅读 [评论](http://blog.csdn.net/huaxun66/article/details/53333747#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/huaxun66/article/details/53333747#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

技术关注（2） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录[(?)[+]](http://blog.csdn.net/huaxun66/article/details/53333747)

MediaPlayer

MediaPlayer类是Androd多媒体框架中的一个重要组件，通过该类，我们可以以最小的步骤来获取，解码和播放音视频。它支持三种不同的媒体来源：

* 本地资源
* 内部URI，比如你可以通过ContentResolver来获取
* 外部URL(流)

对于**[Android](http://lib.csdn.net/base/android" \o "Android知识库" \t "_blank)**支持的媒体格式列表，可见：[Supported Media Formats文档](http://blog.sina.com.cn/s/blog_53ad73d70101c6om.html)

在播放网络上的视频流时，Android原生的MediaPlayer支持两种协议，HTTP和RTSP，这两种协议最大的不同是，RTSP协议支持实时流媒体的播放，而HTTP协议不支持。因为VideoView的底层实现是MediaPlayer，因此VideoView也支持以上两种协议。   
但是Android原生MediaPalyer支持的协议（不支持RTMP、MMS等）和封装格式实在太有限了，如果我们想播放那些它不支持的视频，这时候就需要第三方播放器了，很多第三方播放器的底层实现都是基于FFmpeg，后续我会专门来讲讲第三方播放器和FFmpeg的使用。

相关方法详解

（1）获得MediaPlayer实例：   
可以直接new或者调用create方法创建：

MediaPlayer mp = new MediaPlayer();

MediaPlayer mp = MediaPlayer.create(this, R.raw.test); //无需再调用setDataSource

* 1
* 2
* 1
* 2

另外create还有这样的形式：

create(Context context, Uri uri, SurfaceHolder holder)

* 1
* 1

通过Uri和指定 SurfaceHolder 【抽象类】 创建一个多媒体播放器。

（2）设置播放文件：

MediaPlayer.create(this, R.raw.test); //①raw下的资源

mp.setDataSource("/sdcard/test.mp3"); //②本地文件路径

mp.setDataSource("http://www.xxx.com/music/test.mp3");//③网络URL文件

* 1
* 2
* 3
* 1
* 2
* 3

另外setDataSource()方法有多个，里面有这样一个类型的参数：FileDescriptor   
在使用这个API的时候，需要把文件放到res文件夹平级的assets文件夹里，然后使用下述代码设置DataSource：

AssetFileDescriptor fileDescriptor = getAssets().openFd("rain.mp3");

mediaPlayer.setDataSource(fileDescriptor.getFileDescriptor(),fileDescriptor.getStartOffset(), fileDescriptor.getLength());

* 1
* 2
* 1
* 2

（3）其他方法

getCurrentPosition( )：得到当前的播放位置

getDuration() ：得到文件的时间

getVideoHeight() ：得到视频高度

getVideoWidth() ：得到视频宽度

isLooping()：是否循环播放

isPlaying()：是否正在播放

pause()：暂停

prepare()：准备(同步)

prepareAsync()：准备(异步)

release()：释放MediaPlayer对象

reset()：重置MediaPlayer对象

seekTo(int msec)：指定播放的位置（以毫秒为单位的时间）

setAudioStreamType(int streamtype)：指定流媒体的类型

setDisplay(SurfaceHolder sh)：设置用SurfaceHolder来显示多媒体

setLooping(boolean looping)：设置是否循环播放

setOnBufferingUpdateListener(MediaPlayer.OnBufferingUpdateListener listener)：网络流媒体的缓冲监听

setOnCompletionListener(MediaPlayer.OnCompletionListener listener)：网络流媒体播放结束监听

setOnErrorListener(MediaPlayer.OnErrorListener listener)：设置错误信息监听

setOnVideoSizeChangedListener(MediaPlayer.OnVideoSizeChangedListener listener)：视频尺寸监听

setScreenOnWhilePlaying(boolean screenOn)：设置是否使用SurfaceHolder显示

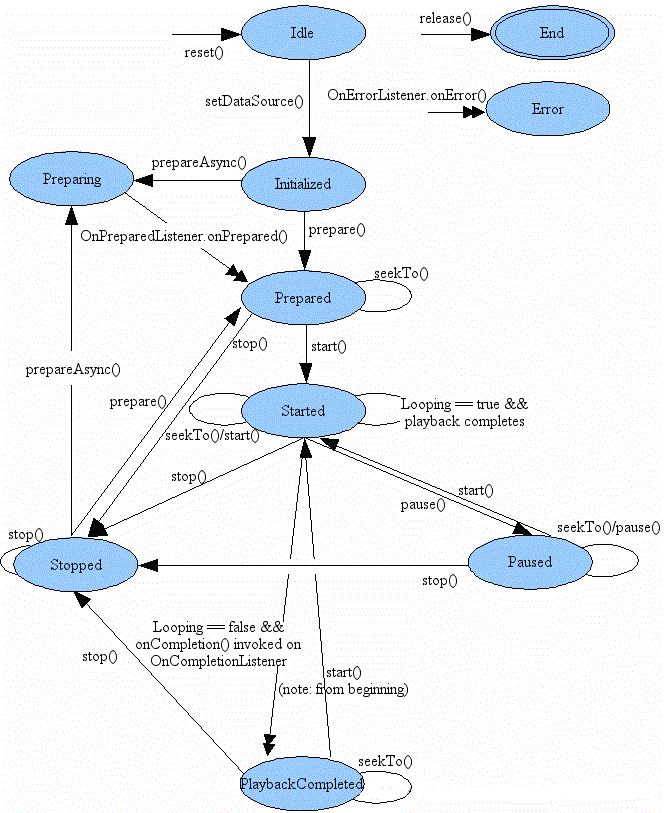
setVolume(float leftVolume, float rightVolume)：设置音量

start()：开始播放

stop()：停止播放

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23

生命周期



* 状态1：Idel(空闲)状态   
  当 mediaplayer创建或者执行reset()方法后处于这个状态。
* 状态2：Initialized(已初始化)状态   
  当调用mediaplayer的setDataResource()方法给mediaplayer设置播放的数据源后，mediaplayer会处于该状态。
* 状态3：Prepared(准备就续)状态   
  设置完数据源后，调用mediaplayer的prepare()方法，让mediaplayer准备播放。值得一提的是，这里除了prepare()方法，还有prepareAsnyc()方法，此方法是异步方法，一般用于网络视频的缓冲。当缓冲完毕后，就会触发准备完毕的事件。我们要做的就是监听该事件(OnPreparedListener)，当缓冲完成时，执行相应的操作。在此状态上，我们可以调用seekTo()方法定位视频，此方法不改变mediaplayer的状态；亦可调用stop()放弃视频播放，使mediaplayer处于Stopped状态。一般我们会在此状态上调用start()方法开始播放视频。
* 状态4：Started(开始)状态   
  当处于Prepared状态、Paused状态和PlayebackCompeleted状态时，调用Started()方法即可进入该状态。在该状态中，mediaplayer开始播放视频，可以通过seekTo()方法和start()方法改变视频播放的进度，当Looping为真且播放完毕后，它会重新开始播放（即循环播放）；否则播放完毕后，会触发事件并调用OnCompletionaListener.OnCompletion()方法，进行特定操作，并进入PlaybackCompleted状态。在此状态中，亦可调用pause()方法或者stop()方法让视频暂停或停止，此时mediaplayer分别处于Stopped和Paused状态。
* 状态5：Stopped(停止)状态   
  当 mediaplayer处于Prepared、Started、Paused、PlaybackCompleted状态时，调用stop()方法即可进入本状态。应特别注意的是，在本状态中，若想重新开始播放，不能直接调用start()方法，必须调用prepare()方法或prepareAsync()方法重新让mediaplayer处于Prepared状态方可调用start()方法播放视频。
* 状态6：Paused(暂停)状态   
  当mediaplayer处于Started状态是，调用pause()方法即可进入本状态。在本状态里，可直接调用start()方法使，mediaplayer回到Started状态，亦可调用stop()方法停止视频播放，让播放器处于停止态。
* 状态7：PlaybackCompleted(播放完成)状态   
  当mediaplayer播放完成且Looping为假时即可进入本状态。在本状态可调用start()方法使mediaplayer回到Started状态（注意此时是从头开始播放）；亦可调用stop()方法使mediaplayer处于停止态，结束播放。
* 状态8：Error(错误)状态   
  当mediaplayer出现错误时处于此状态。

调用release()方法即可释放此mediaplayer对象。

使用MediaPlayer播放音频

@Override

public void onClick(View v) {

switch (v.getId()){

case R.id.btn\_play:

if(isRelease){

mPlayer = MediaPlayer.create(this,R.raw.fly);

isRelease = false;

}

mPlayer.start(); //开始播放

break;

case R.id.btn\_pause:

mPlayer.pause(); //停止播放

break;

case R.id.btn\_stop:

mPlayer.reset(); //重置MediaPlayer

mPlayer.release(); //释放MediaPlayer

isRelease = true;

break;

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21

注意事项：   
播放的是res/raw目录下的音频文件，创建MediaPlayer调用的是create方法，第一次启动播放前不需要再调用prepare()，如果是使用构造方法构造的话，则需要调用一次prepare()方法！   
另外贴下官方文档中，从其他两种途径播放音频的示例代码:   
**本地Uri：**

Uri myUri = ....; /\*\*initialize Uri here\*/

MediaPlayer mediaPlayer = new MediaPlayer();

mediaPlayer.setAudioStreamType(AudioManager.STREAM\_MUSIC);

mediaPlayer.setDataSource(getApplicationContext(), myUri);

mediaPlayer.prepare();

mediaPlayer.start();

* 1
* 2
* 3
* 4
* 5
* 6
* 1
* 2
* 3
* 4
* 5
* 6

**外部URL：**

String url = "http://........"; // your URL here

MediaPlayer mediaPlayer = new MediaPlayer();

mediaPlayer.setAudioStreamType(AudioManager.STREAM\_MUSIC);

mediaPlayer.setDataSource(url);

mediaPlayer.prepare(); // might take long! (for buffering, etc)

mediaPlayer.start();

* 1
* 2
* 3
* 4
* 5
* 6
* 1
* 2
* 3
* 4
* 5
* 6

Note：假如你通过一个URL以流的形式播放在线音频文件，该文件必须可以进行渐进式下载。

使用MediaPlayer + SurfaceView播放视频

MediaPlayer主要用于播放音频，没有提供图像输出界面，所以我们需要借助其他的组件来显示MediaPlayer播放的图像输出，我们可以使用SurfaceView来显示。

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<SurfaceView

android:id="@+id/surfaceView"

android:layout\_width="fill\_parent"

android:layout\_height="300dp" />

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_horizontal"

android:orientation="horizontal" >

<ImageButton

android:id="@+id/btnplay"

android:layout\_width="60dp"

android:layout\_height="60dp"

android:background="@mipmap/play"/>

<ImageButton

android:id="@+id/btnpause"

android:layout\_width="60dp"

android:layout\_height="60dp"

android:background="@mipmap/pause"/>

<ImageButton

android:id="@+id/btnstop"

android:layout\_width="60dp"

android:layout\_height="60dp"

android:background="@mipmap/stop"/>

</LinearLayout>

</LinearLayout>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37

public class MediaPlayerActivity extends Activity implements View.OnClickListener {

private ImageButton btnplay, btnstop, btnpause;

private SurfaceView surfaceView;

private MediaPlayer mediaPlayer;

private int position;

private String url1 = "http://flashmedia.eastday.com/newdate/news/2016-11/shznews1125-19.mp4";

private String url2 = "rtsp://184.72.239.149/vod/mp4:BigBuckBunny\_115k.mov";

private String url3 = "http://42.96.249.166/live/388.m3u8";

private String url4 = "http://61.129.89.191/ThroughTrain/download.html?id=4035&flag=-org-"; //音频url

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_mediaplayer);

btnplay = (ImageButton) this.findViewById(R.id.btnplay);

btnstop = (ImageButton) this.findViewById(R.id.btnstop);

btnpause = (ImageButton) this.findViewById(R.id.btnpause);

btnstop.setOnClickListener(this);

btnplay.setOnClickListener(this);

btnpause.setOnClickListener(this);

mediaPlayer = new MediaPlayer();

surfaceView = (SurfaceView) this.findViewById(R.id.surfaceView);

// 设置SurfaceView自己不管理的缓冲区

surfaceView.getHolder().setType(SurfaceHolder.SURFACE\_TYPE\_PUSH\_BUFFERS);

surfaceView.getHolder().addCallback(new SurfaceHolder.Callback() {

@Override

public void surfaceDestroyed(SurfaceHolder holder) {

}

@Override

public void surfaceCreated(SurfaceHolder holder) {

if (position > 0) {

try {

// 开始播放

play();

// 并直接从指定位置开始播放

mediaPlayer.seekTo(position);

position = 0;

} catch (Exception e) {

}

}

}

@Override

public void surfaceChanged(SurfaceHolder holder, int format, int width, int height) {

}

});

}

@Override

public void onClick(View v) {

switch (v.getId()) {

case R.id.btnplay:

play();

break;

case R.id.btnpause:

if (mediaPlayer.isPlaying()) {

mediaPlayer.pause();

} else {

mediaPlayer.start();

}

break;

case R.id.btnstop:

if (mediaPlayer.isPlaying()) {

mediaPlayer.stop();

}

break;

default:

break;

}

}

@Override

protected void onPause() {

// 先判断是否正在播放

if (mediaPlayer.isPlaying()) {

// 如果正在播放我们就先保存这个播放位置

position = mediaPlayer.getCurrentPosition();

mediaPlayer.stop();

}

super.onPause();

}

private void play() {

try {

mediaPlayer.reset();

mediaPlayer.setAudioStreamType(AudioManager.STREAM\_MUSIC);

// 设置需要播放的视频

Uri uri = Uri.parse(url1);

mediaPlayer.setDataSource(getApplicationContext(), uri);

// 把视频画面输出到SurfaceView

mediaPlayer.setDisplay(surfaceView.getHolder());

mediaPlayer.prepare();

// 播放

mediaPlayer.start();

Toast.makeText(this, "开始播放！", Toast.LENGTH\_LONG).show();

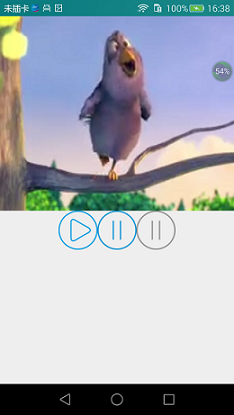
} catch (Exception e) {

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68
* 69
* 70
* 71
* 72
* 73
* 74
* 75
* 76
* 77
* 78
* 79
* 80
* 81
* 82
* 83
* 84
* 85
* 86
* 87
* 88
* 89
* 90
* 91
* 92
* 93
* 94
* 95
* 96
* 97
* 98
* 99
* 100
* 101
* 102
* 103
* 104
* 105
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68
* 69
* 70
* 71
* 72
* 73
* 74
* 75
* 76
* 77
* 78
* 79
* 80
* 81
* 82
* 83
* 84
* 85
* 86
* 87
* 88
* 89
* 90
* 91
* 92
* 93
* 94
* 95
* 96
* 97
* 98
* 99
* 100
* 101
* 102
* 103
* 104
* 105

代码很简单，布局有个SurfaceView，然后调用getHolder获得一个SurfaceHolder对象，在这里完成SurfaceView相关的设置，设置了类型以及一个Callback接口，重写了SurfaceView创建时，发生变化时，以及销毁时的三个方法！然后按钮控制播放、暂停以及停止而已。   


注：这里视频宽高总是等于定义的SurfaceView布局宽高，所以视频可能会被拉伸变形。那么如何解决呢？

...

mMediaPlayer.setOnVideoSizeChangedListener(new OnVideoSizeChangedListener(){

@Override

public void onVideoSizeChanged(MediaPlayer mp, int width, int height) {

if (width == 0 || height == 0) {

Log.e(TAG, "invalid video width(" + width + ") or height(" + height + ")");

return;

}

mIsVideoSizeKnown = true;

mVideoWidth = width;

mVideoHeight = height;

if (mIsVideoReadyToBePlayed && mIsVideoSizeKnown) {

startVideoPlayback();

}

}

});

mMediaPlayer.setOnPreparedListener(new OnPreparedListener(){

@Override

public void onPrepared(MediaPlayer mp) {

mIsVideoReadyToBePlayed = true;

if (mIsVideoReadyToBePlayed && mIsVideoSizeKnown) {

startVideoPlayback();

}

}

});

private void startVideoPlayback() {

holder.setFixedSize(mVideoWidth, mVideoHeight);

mMediaPlayer.start();

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32

VideoView

除了使用MediaPlayer + SurfaceView播放视频的方式，我们还可以使用VideoView来直接播放视频。SurfaceView播放视频时，如果不进行设置，视频宽高总是等于定义的SurfaceView布局宽高，所以视频可能会被拉伸变形。而使用VideoView时，视频宽度等于VideoView布局宽，但是高是自适应的，自动调整宽高比到视频原始比例，所以不会有拉伸。

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/activity\_main"

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="com.hx.mediaplay.MainActivity"

android:orientation="vertical">

<Button

android:id="@+id/go"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="mediaPlayer+sufaceview" />

<EditText

android:id="@+id/url"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="url" />

<Button

android:id="@+id/play"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Play" />

<VideoView

android:id="@+id/video"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_centerInParent="true" />

</LinearLayout>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37

public class MainActivity extends AppCompatActivity {

private Button go;

private EditText url;

private Button button;

private VideoView videoview;

private MediaController mMediaController;

private String url1 = "http://flashmedia.eastday.com/newdate/news/2016-11/shznews1125-19.mp4";

private String url2 = "rtsp://184.72.239.149/vod/mp4:BigBuckBunny\_115k.mov";

private String url3 = "http://42.96.249.166/live/388.m3u8";

private String url4 = "http://61.129.89.191/ThroughTrain/download.html?id=4035&flag=-org-"; //音频url

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

go = (Button) findViewById(R.id.go);

url = (EditText) findViewById(R.id.url);

button = (Button) findViewById(R.id.play);

videoview = (VideoView) findViewById(R.id.video);

mMediaController = new MediaController(this);

videoview.setMediaController(mMediaController);

url.setText(url1);

button.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v) {

loadView(url.getText().toString());

}

});

go.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v) {

Intent intent = new Intent(MainActivity.this, MediaPlayerActivity.class);

startActivity(intent);

}

});

}

public void loadView(String path) {

Uri uri = Uri.parse(path);

videoview.setVideoURI(uri);

videoview.start();

videoview.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mp) {

// mp.setLooping(true);

mp.start();// 播放

Toast.makeText(MainActivity.this, "开始播放！", Toast.LENGTH\_LONG).show();

}

});

videoview.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {

@Override

public void onCompletion(MediaPlayer mp) {

Toast.makeText(MainActivity.this, "播放完毕", Toast.LENGTH\_SHORT).show();

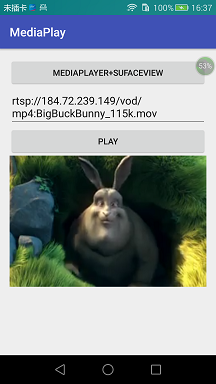
}

});

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63



[**Demo下载地址**](http://download.csdn.net/detail/huaxun66/9693730)