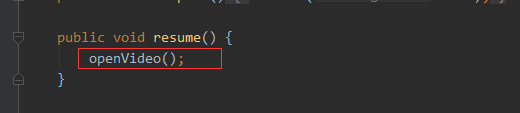
1. VideoView 创建与使用:

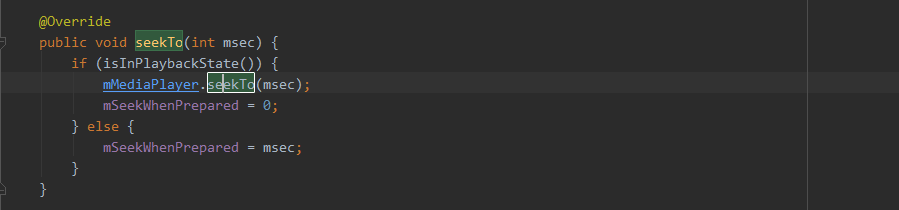
MediaController mediaController = new MediaController(this);  
mBinding.videoView.setMediaController(mediaController);  
mBinding.videoView.setOnPreparedListener(this);  
mBinding.videoView.setOnCompletionListener(this);  
String uri = "android.resource://" + getPackageName() + "/" + R.raw.*guide\_live*;  
mBinding.videoView.setVideoURI(Uri.*parse*(uri));  
mBinding.videoView.start();

1. 由于是本地视频，所以之前我是放到了assets资源文件夹里面了，但是播放不了，Uri uri = Uri.parse(“file:///android\_asset/beep.mp4”); 这样子是不行的。随后查到资料需要放在raw资源文件夹里面，Uri uri = Uri.parse(“android:resources://”+getPackageName()+R.raw.xxx);这样是ok的，
2. 当外部主动触发视频的暂停与恢复时，也遇到了坑；暂停之后，再次使用resume恢复播放视频时，是从头播放的：



其实就是从新创建新的播放器，从头播放。针对这一个坑，目前发现有两种解决方案：

1》、在暂停的时机通过获取当前进度，videoView.getCurrentPosition();而videoView提供了一个seekTo



其实还是给了MediaPlayer播放控制器，这样子在恢复的时候就可以通过videoView.seekTo指定上一次播放位置：

Public void pause(){

Int position = videoView.getCurrentPosition();

videoView.pause();

}

Public void resume(){

videoView.seekTo(currentPostion);

videoView.start(); // videoView.resume()好像不行

}

2》、第二种是通过获取MediaPlayer实例来操作视频的暂停与播放,实现MediaPlayer.OnPreparedListener接口，可以看到实现方法

@Override

public void onPrepared(MediaPlayer mp) {

}

可以获得播放器的实例，这样子就不用记录暂停时播放的位置。

1. 如果需要在播放之前展示该视频的缩略图，那么自然也有方法：

@TargetApi(Build.VERSION\_CODES.ICE\_CREAM\_SANDWICH)

private void createVideoThumbnail() {

Observable<Bitmap> observable = Observable.create(new Observable.OnSubscribe<Bitmap>() { @Override public void call(Subscriber<? super Bitmap> subscriber) { Bitmap bitmap = null; MediaMetadataRetriever retriever = new MediaMetadataRetriever(); int kind = MediaStore.Video.Thumbnails.MINI\_KIND; if (Build.VERSION.SDK\_INT >= 14) { retriever.setDataSource(mVideoUrl, new HashMap<String, String>()); } else { retriever.setDataSource(mVideoUrl); } bitmap = retriever.getFrameAtTime(); subscriber.onNext(bitmap); retriever.release(); } }); observable.observeOn(AndroidSchedulers.mainThread()) .subscribeOn(Schedulers.io()) .subscribe(new Action1<Bitmap>() { @Override public void call(Bitmap bitmap) { //设置封面 mYourVideoPlayerContainer.setBackgroundDrawable(new BitmapDrawable(bitmap)); } }); }