

使用 MediaPlayer 播放音乐

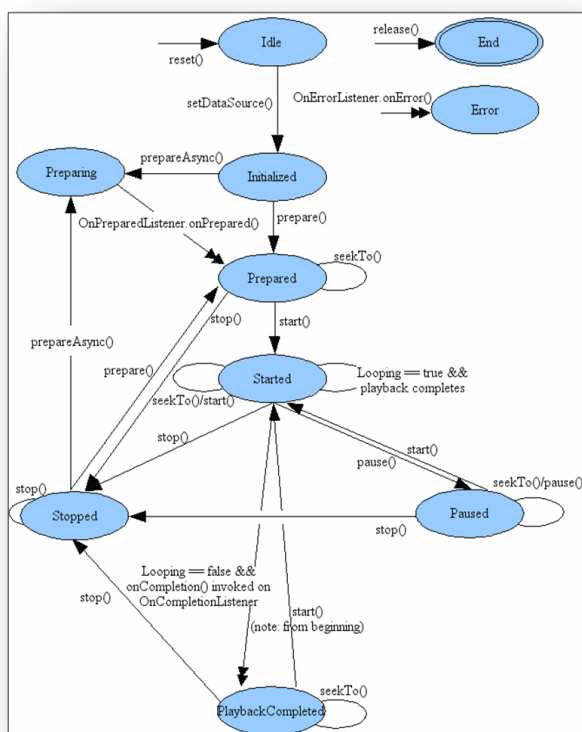
知识解析

MediaPlayer 可用于控制播放声音/视频，下图描述播放状态的转换。

一旦发生错误，MediaPlayer 对象会进入到 Error 状态。

为了重用处于 Error 状态的 MediaPlayer 对象，可以调用 `reset()` 方法来把这个对象恢复成 Idle 状态。

在开始播放之前，MediaPlayer 对象必须要进入 Prepared 状态。



有两种方法（同步和异步）可以使 MediaPlayer 对象进入 Prepared 状态：要么调用 `prepare()` 方法（同步），此方法返回就表示该 MediaPlayer 对象已经进入了 Prepared 状

态;要么调用 `prepareAsync()` 方法(异步),此方法会使此 `MediaPlayer` 对象进入 `Preparing` 状态并返回,而内部的播放引擎会继续未完成的准备工作。

`Preparing` 是一个中间状态,在此状态下调用任何方法的结果都是未知的。

要开始播放,必须调用 `start()` 方法。当此方法成功返回时,`MediaPlayer` 的对象处于 `Started` 状态。`isPlaying()` 方法可以被调用来测试某个 `MediaPlayer` 对象是否在 `Started` 状态。

播放可以被暂停,停止,以及调整当前播放位置。当调用 `pause()` 方法并返回时,会使 `MediaPlayer` 对象进入 `Paused` 状态。

调用 `start()` 方法会让一个处于 `Paused` 状态的 `MediaPlayer` 对象从之前暂停的地方恢复播放。当调用 `start()` 方法返回的时候,`MediaPlayer` 对象的状态会又变成 `Started` 状态。

调用 `stop()` 方法会停止播放,并且还会让一个处于 `Started`、`Paused`、`Prepared` 或 `PlaybackCompleted` 状态的 `MediaPlayer` 进入 `Stopped` 状态。

调用 `seekTo()` 方法可以调整播放的位置。

目前的播放位置,实际可以调用 `getCurrentPosition()` 方法得到。

相关的监听器

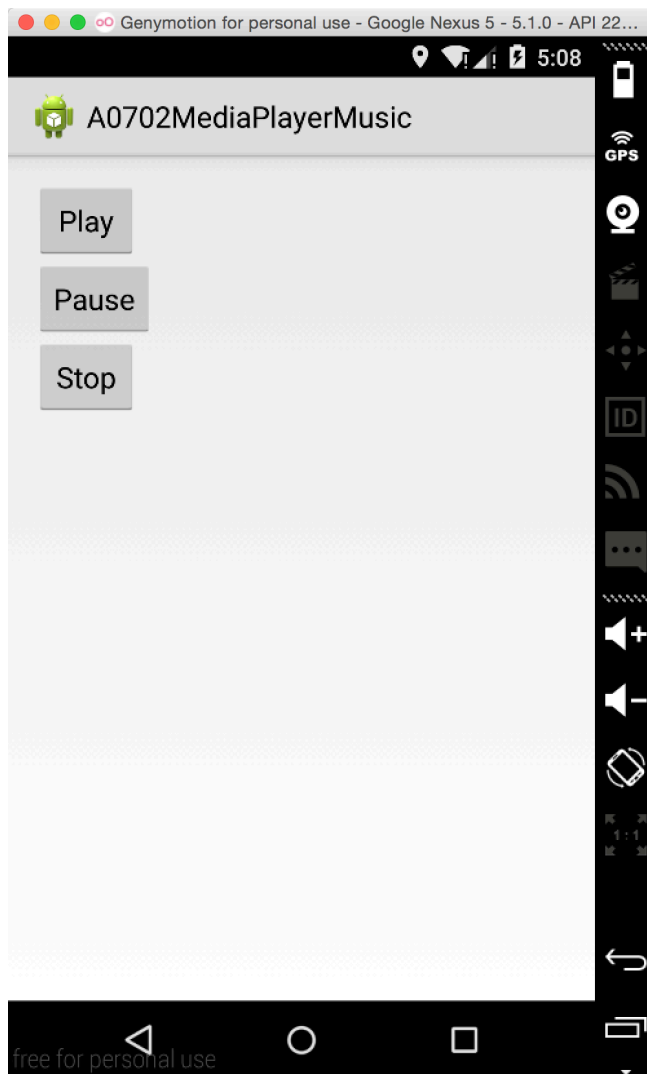
当播放发生错误时,可以使用 `OnErrorListener` 监听

当准备工作完全完成时,可以使用 `OnPreparedListener` 监听

当播放完成时,可以使用 `OnCompletionListener` 监听

当实际的定位播放操作完成之后,可以使用 `OnSeekCompleteListener` 方法

功能演示



实战操作

第一步，实例化 MediaPlayer 对象，并设置播放文件的路径和状态。

```
player = new MediaPlayer(); // idle
try {
    player.setDataSource(
```

```
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC) +  
"/jyz.mp3");  
  
    } catch (IllegalArgumentException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    } catch (SecurityException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    } catch (IllegalStateException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```

职业素质

MediaPlayer 可用于控制播放声音/视频, 是 Android 中最常用的播放音频和视频的方法。MediaPlayer 要播放的文件主要包括 3 个来源:

1. 用户在应用中事先自带的 resource 资源

例如: `MediaPlayer.create(this, R.raw.test);`

2. 存储在 SD 卡或其他文件路径下的媒体文件

例如: `mp.setDataSource("/sdcard/test.mp3");`

3. 网络上的媒体文件, 例如:

`mp.setDataSource("http://www.citynorth.cn/music/confucius.mp3");`

