Basillee的个人博客

- ⌂ 首页
- ❓ 标签
- ▣ 归档

Basillee

人生就像爬坡，得一步步来。

30	**1**	**8**
日志	分类	标签

Android MediaPlay的一个简单工具类及几点注意

发表于 2018-08-31 | 更新于: 2019-08-04 | 分类于 技术 | 11 次

字数统计: 818 | 阅读时长 ≈ 4

步骤一：代码如下

```java
import android.content.Context;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Handler;
import android.os.Message;
import android.util.Log;
import android.widget.ProgressBar;
import android.widget.SeekBar;

import java.io.IOException;

/**
 * Created by imac on 2018/8/28.
 */
public class MediaUtil implements MediaPlayer.OnPreparedListener, MediaPlayer.

    private static final String TAG = "LOG_MediaUtil";
    private static final int INTERNAL_TIME = 1000;

    private Context mContext;


    private MediaPlayer mediaPlayer;

    private SeekBar progressBar;


    public void setProgressBar(SeekBar progressBar) {
        this.progressBar = progressBar;
        this.progressBar.setOnSeekBarChangeListener(this);
    }

    private final int MSG_UPDATA_PROGRESS = 0x10001;

    private Handler mHandler = new Handler(){
        @Override
        public void handleMessage(Message msg) {
            switch (msg.what){
                case MSG_UPDATA_PROGRESS:
                    Log.i(TAG,"MSG_UPDATA_PROGRESS");
                    // 展示给进度条和当前时间
```

```java
                    if (mediaPlayer == null){
                        return;
                    }
                    int progress = mediaPlayer.getCurrentPosition();
                    if (progressBar != null){
                        Log.i(TAG,"MSG_UPDATA_PROGRESS progress = " + progress
                        progressBar.setProgress(progress);
                    }
                    updateProgress();
                    break;
                default:
                    break;
            }
            super.handleMessage(msg);
        }
    };

    public MediaUtil(Context mContext) {
        this.mContext = mContext;
        mediaPlayer = new MediaPlayer();
    }


    public void playMusic(final String musilcUrl) {
//        String url = "http://m10.music.126.net/20180829015325/2d0c208babcda7
        Log.i(TAG, "playMusic: enter");
        Uri musicUri = Uri.parse(musilcUrl);
//        Uri musicUri = Uri.parse(url);
        Log.i(TAG, "playMusic: musicUri = " + musicUri);
        if (mediaPlayer.isPlaying()) {
            Log.i(TAG, "playMusic: isPlaying");
            stopMusic();
        }

        try {

            mediaPlayer.setDataSource(mContext, musicUri);
            mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
            mediaPlayer.prepareAsync();
            mediaPlayer.setLooping(true);
            mediaPlayer.setOnPreparedListener(this);
            mediaPlayer.start();
        } catch (IOException e) {
            e.printStackTrace();
        }

    }

    public void stopMusic() {
        Log.i(TAG, "stopMusic: enter");
        if (mediaPlayer == null) {
            Log.i(TAG, "stopMusic: is null mediaPlayer");
            return;
        }
        mediaPlayer.reset();
        mediaPlayer.stop();
    }
```

```java
    @Override
    public void onPrepared(MediaPlayer mp) {
        Log.i(TAG, "onPrepared: enter getDuration is =" + mp.getDuration());
        mp.start();
        updateProgress();
        progressBar.setMax(mp.getDuration());
    }


    public void releaseMediaPlayer() {
        Log.i(TAG, "releaseMediaPlayer: enter");
        if (mediaPlayer != null) {
            mHandler.removeMessages(MSG_UPDATA_PROGRESS);
            mediaPlayer.stop();
            mediaPlayer.release();
            mediaPlayer = null;
        }
    }

    @Override
    public void onCompletion(MediaPlayer mp) {

    }



    // 每间隔1s通知更新进度
    private void updateProgress() {
        // 使用Handler每间隔1s发送一次空消息，通知进度条更新
        Message msg = Message.obtain();// 获取一个现成的消息
        msg.what = MSG_UPDATA_PROGRESS;
        mHandler.sendMessageDelayed(msg, INTERNAL_TIME);
    }

    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromU
        if(mediaPlayer != null && fromUser){
            mediaPlayer.seekTo(progress);
        }
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

    }
}
```

步骤二：注意点

1.勿忘mediaPlayer.reset();

当 媒 体 已 经 播 放 了 的 时 候 ， 如 果 再 直 接 调 用 mediaPlayer.stop() 暂 停 音 乐 ， 而 不 调 用

mediaPlayer.reset();且接使用setDataSource()重新播放，则会出现IllegalStateException 异常。

源码如下：

```
/**
 * Sets the data source as a content Uri.
 *
 * @param context the Context to use when resolving the Uri
 * @param uri the Content URI of the data you want to play
 * @throws IllegalStateException if it is called in an invalid state
 */
public void setDataSource(@NonNull Context context, @NonNull Uri uri)
        throws IOException, IllegalArgumentException, SecurityException, Illeg
    setDataSource(context, uri, null, null);
}
```

2.建议使用 mediaPlayer.prepareAsync();

prepare()会阻塞UI，容易造成卡顿，建议使用异步的prepareAsync();

源码：

```
/**
 * Prepares the player for playback, synchronously.
 *
 * After setting the datasource and the display surface, you need to either
 * call prepare() or prepareAsync(). For files, it is OK to call prepare(),
 * which blocks until MediaPlayer is ready for playback.
 *
 * @throws IllegalStateException if it is called in an invalid state
 */
public void prepare() throws IOException, IllegalStateException {
    _prepare();
    scanInternalSubtitleTracks();

    // DrmInfo, if any, has been resolved by now.
    synchronized (mDrmLock) {
        mDrmInfoResolved = true;
    }
}

/**
 * Prepares the player for playback, asynchronously.
 *
 * After setting the datasource and the display surface, you need to either
 * call prepare() or prepareAsync(). For streams, you should call prepareAsync
 * which returns immediately, rather than blocking until enough data has been
 * buffered.
 *
 * @throws IllegalStateException if it is called in an invalid state
 */
public native void prepareAsync() throws IllegalStateException;
```

使用prepareAsync()时，实现OnPreparedListener接口，当资源准备好之后，调用mp.start();播放媒体

3.进度条如代码所示，不细说了。主要实现SeekBar.OnSeekBarChangeListener 接口，进行拖动进度条时进行监听。

步骤三：进一步思考

由于Android 原生的MediaPlay没有缓存，重复播放相同音乐时会造成重复加载网络资源。这部分可以考虑缓存到本地，减少流量的消耗以及加快播放速度。

本文作者： Basillee

本文链接： http://qrcodewithlogo.com/Android MediaPlay的一个简单工具类及几点注意.html

版权声明： 本博客所有文章除特别声明外，均采用 CC BY-NC-SA 3.0 许可协议。转载请注明出处！

Android