

Huaxiao's Blog

Android多媒体文件管理浅析

📅 2018-04-30 | 👁 浏览 次

简介

Android多媒体文件管理简单来说，有三个部分

- 扫描：MediaScannerService
- 存储：MediaProvider
- 查询：MediaStore

MediaScannerService

值得一提的是，MediaScannerService这类文件，AndroidStudio中不能直接search到，需要去官网查询其源码，这里推荐一个在线查看Android源码的链接<http://androidxref.com/>

```
1  public class MediaScannerService extends Service implements Runnable
2
3  // 核心方法
4  private void scan(String[] directories, String volumeName) {
5      ...
6
7      try {
8          ...
9
10         try {
11             ...
12
13             MediaScanner scanner = createMediaScanner();
14             scanner.scanDirectories(directories, volumeName);
15         } catch (Exception e) {
16             ...
17         }
18
19         ...
20
21     } finally {
22         ...
```

```
23     }  
24 }
```

应用级Service，用于媒体文件扫描

scan是其核心方法，从源码中可以看出，最终是调用了MediaScanner的scanDirectories方法

MediaScannerReceiver

与MediaScannerService配套的BroadcastReceiver

```
1  @Override  
2  public void onReceive(Context context, Intent intent) {  
3      final String action = intent.getAction();  
4      final Uri uri = intent.getData();  
5      if (Intent.ACTION_BOOT_COMPLETED.equals(action)) {  
6          // Scan both internal and external storage  
7          scan(context, MediaProvider.INTERNAL_VOLUME);  
8          scan(context, MediaProvider.EXTERNAL_VOLUME);  
9      }  
10     } else {  
11         if (uri.getScheme().equals("file")) {  
12             // 省略...  
13             if (Intent.ACTION_MEDIA_MOUNTED.equals(action)) {  
14                 // scan whenever any volume is mounted  
15                 scan(context, MediaProvider.EXTERNAL_VOLUME);  
16             } else if (Intent.ACTION_MEDIA_SCANNER_SCAN_FILE.equals(action) &&  
17                 path != null && path.startsWith(externalStoragePath + "/"))  
18                 scanFile(context, path);  
19         }  
20     }  
21 }  
22 }
```

从onReceive方法中可以看出，MediaScannerReceiver执行scan的时机有三种

- 启动完毕，扫描内部存储和外部存储
- sdcard挂载完毕，扫描外部存储
- 扫描单个文件

MediaScanner

具体执行scan动作的类

MediaScannerProvider

ContentProvider用于保存查询结果，结果由数据库保存

以外部存储为例，数据库的路径为/data/data/com.android.providers.media/databases/external.db

MediaStore

MediaStore是面向应用的入口，作用是便于用户查询已经扫描过的数据

MediaStore中的资源有四类

- MediaStore.Files
- MediaStore.Audio
- MediaStore.Images
- MediaStore.Video

媒体库查询

核心方法

```
1 public final @Nullable Cursor query(@RequiresPermission.Read
2                                     @NonNull Uri uri,
3                                     @Nullable String[] projection,
4                                     @Nullable String selection,
5                                     @Nullable String[] selectionArgs,
6                                     @Nullable String sortOrder)
```

Uri uri

```
1 Uri uri = MediaStore.Video.Media.INTERNAL_CONTENT_URI;
2 Uri uri = MediaStore.Video.Media.EXTERNAL_CONTENT_URI;
```

Uri总共有两种，分别对应内部存储和外部存储

String[] projection

```
1 String[] mediaColumns = {
2     MediaStore.Video.Media.DATA,
```

```
3         MediaStore.Video.Media.SIZE,  
4         MediaStore.Video.Media.DATE_MODIFIED,  
5         MediaStore.Video.Media.DURATION};
```

用于指定查询后，返回给用户的媒体信息

String selection & String[] selectionArgs

这两个是可选项，用户可以通过这两个参数，定制化查询条件

```
String selection = MediaStore.Video.Media.MIME_TYPE + "=? or "  
    + MediaStore.Video.Media.MIME_TYPE + "=? or "  
    + MediaStore.Video.Media.MIME_TYPE + "=? or "  
    + MediaStore.Video.Media.MIME_TYPE + "=? or "  
    + MediaStore.Video.Media.MIME_TYPE + "=?";  
  
String[] selectionArgs = new String[]{"video/mp4", "video/avi", "video/quicktime", "video/w
```

需要注意的是，selection中的条件和selectionArgs数组中的类型，需要是一一对应的，否则会发生异常

String sortOrder

查询的排序方式

.nomedia

文件夹中新建一个.nomedia的空文件，会屏蔽掉系统默认的媒体库扫描。遇到带有该文件的文件夹，只能通过文件遍历的方式进行扫描

MIME_TYPE

需要注意的是MIMETYPE不一定和文件的扩展名一致，比如mov格式的文件，MIMETYPE为video/quicktime等等

新增视频资源并通知手机相册

基本方法

方法一：发送广播，通知媒体库刷新

```
1 Intent intent = new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);
2 intent.setData(Uri.fromFile(new File(filePath)));
3 context.sendBroadcast(intent);
```

方法二：插入数据库

也可以使用直接插入媒体库数据库的方法来实现（但这两种方法不能同时使用，会导致媒体库中出现重复的数据）

```
1 ContentValues values = new ContentValues(4);
2 values.put(MediaStore.Video.Media.TITLE, "");
3 values.put(MediaStore.Video.Media.MIME_TYPE, minetype);
4 values.put(MediaStore.Video.Media.DATA, path);
5 values.put(MediaStore.Video.Media.DURATION, duration_int);
6 context.getContentResolver().insert(MediaStore.Video.Media.EXTERNAL_CONTENT_URI,
```

插入数据库的这种方法，这边博文中有详细介绍[blog](#)

方法三：MediaScannerConnection.scanFile

```
1 public static void insert(Context context, String[] paths, String[] types) {
2     MediaScannerConnection.scanFile(
3         context,
4         paths,
5         types,
6         new MediaScannerConnection.OnScanCompletedListener() {
7             @Override
8             public void onScanCompleted(String path, Uri uri) {
9                 LogUtils.i(TAG, "insert onScanCompleted path " + path + " ur
10             }
11         });
12 }
```

机型适配

部分机型上述两种方法都不能实现实时刷新（只能重启手机才能看到效果），考虑可能是第三方ROM的改动导致的，将目录设置为 /sdcard/DCIM/Camera 可以解决大部分机型的问题，但vivo和魅族手机仍旧不能刷新，vivo的地址应该为 /sdcard/相机，魅族的地址应该为 /sdcard/DCIM/Video

下面给出一个获取输出地址的方法

```
1 public static String getVideoOutputDir() {
2     String sdcardPath = Environment.getExternalStorageDirectory().getPath() + Fi
3     String brand = SystemUtil.getDeviceBrand();
4
5     if (TextUtils.isEmpty(brand)) {
6         return sdcardPath + "DCIM" + File.separator + "Camera";
7     }
8
9     brand = brand.toLowerCase();
10    if (brand.contains("vivo")) {
11        return sdcardPath + "相机";
12    } else if (brand.contains("meizu")) {
13        return sdcardPath + "DCIM" + File.separator + "Video";
14    }
15
16    return sdcardPath + "DCIM" + File.separator + "Camera";
17 }
18
19 public static String getDeviceBrand() {
20     return android.os.Build.BRAND;
21 }
```

值得一提的是，如果想删除相册中的某个资源文件，OPPO的某些机型（ColorOS 3.1系统），会弹窗提示：删除图库中的照片和视频权限申请，需要用户手动去确认删除。这个没有特别好的办法

参考资料

- <https://my.oschina.net/youranhongcha/blog/787223>
- <https://droidyue.com/blog/2014/07/12/scan-media-files-in-android-chinese-edition/>

#Android #多媒体

◀ 从问题出发，解析Activity、Window、View三者关系

