

Android调起系统视频并回调得到视频路径

2022-03-11 14:07:16 By 小峰

闲言少叙，直接上干货。

调起系统视频集：

```
Intent intent = new Intent();
intent.setType("video/*");
intent.setAction(Intent.ACTION_GET_CONTENT);
intent.addCategory(Intent.CATEGORY_OPENABLE);
startActivityForResult(intent, 1111);
```

onActivityResult中的回调

```
if (requestCode == 1111) {
    if (data != null) {
        int duration = data.getIntExtra("dur", 0);
        String realPathFromUri = UriUtils.getPath(mContext, data.getData());
        L.e("111", "" + realPathFromUri);
        File file = new File(PathUtil.getInstance().getImagePath(), "thvideo" + System.currentTimeMillis());
        try {
            FileOutputStream fos = new FileOutputStream(file);
            Bitmap ThumbBitmap = ThumbnailUtils.createVideoThumbnail(realPathFromUri, 3);
            ThumbBitmap.compress(Bitmap.CompressFormat.JPEG, 100, fos);
            fos.close();
            sendVideoMessage(realPathFromUri, file.getAbsolutePath(), duration);
        } catch (Exception e) {
            e.printStackTrace();
        }
    } else {
        Toast.makeText(mContext, "文件损坏，请重新选择", Toast.LENGTH_SHORT).show();
    }
}
```

其中用到的工具类：

```
@SuppressWarnings("NewApi")
public class UriUtils {
```

```
/**
```

```
 * Get a file path from a Uri. This will get the the path for Storage Access
```

- * Framework Documents, as well as the _data field for the MediaStore and
- * other file-based ContentProviders.
- *
- * @param context The context.
- * @param uri The Uri to query.
- */

```
public static String getPath(final Context context, final Uri uri) {

    final boolean isKitKat = Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT;

    // DocumentProvider
    if (isKitKat && DocumentsContract.isDocumentUri(context, uri)) {
        // ExternalStorageProvider
        if (isExternalStorageDocument(uri)) {
            final String docId = DocumentsContract.getDocumentId(uri);
            final String[] split = docId.split(":");
            final String type = split[0];

            if ("primary".equalsIgnoreCase(type)) {
                return Environment.getExternalStorageDirectory() + "/" + split[1];
            }

            // TODO handle non-primary volumes
        }
        // DownloadsProvider
        else if (isDownloadsDocument(uri)) {

            final String id = DocumentsContract.getDocumentId(uri);
            final Uri contentUri = ContentUris.withAppendedId(
                Uri.parse("content://downloads/public_downloads"), Long.valueOf(id));

            return getDataColumn(context, contentUri, null, null);
        }
        // MediaProvider
        else if (isMediaDocument(uri)) {
            final String docId = DocumentsContract.getDocumentId(uri);
            final String[] split = docId.split(":");
            final String type = split[0];

            Uri contentUri = null;
            if ("image".equalsIgnoreCase(type)) {
                contentUri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
            } else if ("video".equalsIgnoreCase(type)) {
                contentUri = MediaStore.Video.Media.EXTERNAL_CONTENT_URI;
            } else if ("audio".equalsIgnoreCase(type)) {
                contentUri = MediaStore.Audio.Media.EXTERNAL_CONTENT_URI;
            }

            final String selection = "_id=?";
            final String[] selectionArgs = new String[] {
```

```
        split[1]
    };

    return getDataColumn(context, contentUri, selection, selectionArgs);
}
}
// MediaStore (and general)
else if ("content".equalsIgnoreCase(uri.getScheme())) {
    return getDataColumn(context, uri, null, null);
}
// File
else if ("file".equalsIgnoreCase(uri.getScheme())) {
    return uri.getPath();
}

return null;
}

/**
 * Get the value of the data column for this Uri. This is useful for
 * MediaStore Uris, and other file-based ContentProviders.
 *
 * @param context The context.
 * @param uri The Uri to query.
 * @param selection (Optional) Filter used in the query.
 * @param selectionArgs (Optional) Selection arguments used in the query.
 * @return The value of the _data column, which is typically a file path.
 */
public static String getDataColumn(Context context, Uri uri, String selection,
                                   String[] selectionArgs) {

    Cursor cursor = null;
    final String column = "_data";
    final String[] projection = {
        column
    };

    try {
        cursor = context.getContentResolver().query(uri, projection, selection, selectionArgs,
            null);
        if (cursor != null && cursor.moveToFirst()) {
            final int column_index = cursor.getColumnIndexOrThrow(column);
            return cursor.getString(column_index);
        }
    } finally {
        if (cursor != null)
            cursor.close();
    }
    return null;
}
```

```
/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is ExternalStorageProvider.
 */
public static boolean isExternalStorageDocument(Uri uri) {
    return "com.android.externalstorage.documents".equals(uri.getAuthority());
}

/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is DownloadsProvider.
 */
public static boolean isDownloadsDocument(Uri uri) {
    return "com.android.providers.downloads.documents".equals(uri.getAuthority());
}

/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is MediaProvider.
 */
public static boolean isMediaDocument(Uri uri) {
    return "com.android.providers.media.documents".equals(uri.getAuthority());
}
}
```

我的博客即将搬运同步至腾讯云+社区，邀请大家一同入驻：https://cloud.tencent.com/developer/support-plan?invite_code=130bka724y4ve

版权声明：

本站遵循 [署名-非商业性使用-相同方式共享 2.5](#) 共享协议。

转载请注明转自[疯人峰语](#)并标明URL。

本文链接：<https://www.iiii.name/?p=2490>