Android ContentProvider 使用

牧秦丶 关注
0.533 2017.11.03 10:02:25 字数 599 阅读 11,014

1、简介

ContentProvider 为存储和获取数据提供统一的接口。可以在不同的应用程序之间共享数据。Android已经为常见的一些数据提供了默认的 ContentProvider。

数据通过唯一的 **URI** 标识来源。ContentProvider 将数据看作表，查询 / 操作数据的时候，通过类似数据库操作的 `insert / delete / query / update` 方法来实现增删查改操作。

作为应用间数据交换 / 共享接口，当然需要有一个"桥梁"来连接数据提供方和使用方。数据提供方提供数据，使用方使用 `content://authorities/path` 类似的 URI 来访问数据。

2、数据提供方

2.1、Provider 声明

数据提供方需要在 `Androidmanifest.xml` 中声明 ContentProvider 组件。一个 ContentProvider 组件声明如下：

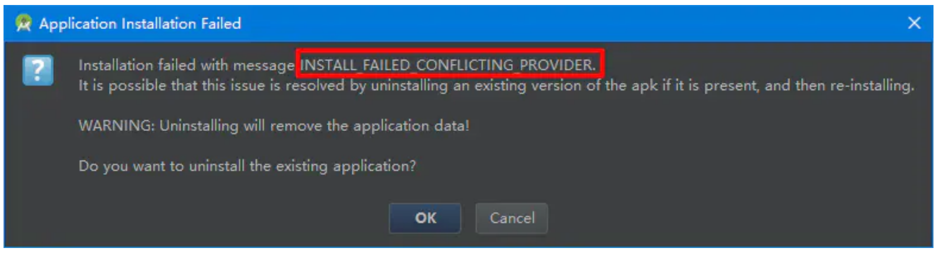
```
<permission
    android:name="com.straw.providerhost.course.read"
    android:protectionLevel="normal"/>
<permission
    android:name="com.straw.providerhost.course.write"
    android:protectionLevel="normal"/>

<application>
    <provider
        android:authorities="com.straw.course"
        android:name="com.straw.providerhost.CourseContentProvider"
        android:exported="true"
        android:readPermission="com.straw.providerhost.course.read"
        android:writePermission="com.straw.providerhost.course.write"/>
</application>
```

属性	含义
authorities	指定 authorieties（类似包名的格式，如 `com.xx.xx`），访问者通过 `content://authorities/path` 的形式访问
name	这个 Provider 对应的 Java 类名
exported	为 `true` 则表示导出。不设置或设置为 `false` 时，其他应用如果使用这个 Provider，则会出现 `java.lang.SecurityException: Permission Denial` 错误
readPermission	读数据者所需声明权限，数据访问者不声明权限则会出现 `java.lang.SecurityException: Permission Denial` 错误
writePermission	写数据者所需声明权限，数据访问者不声明权限则会出现 `java.lang.SecurityException: Permission Denial` 错误

2.2、其他情况

那两个 App 同时声明相同 `authorities` 的 ContentProvider 可不可以呢？答案是 **不行**。如果两个 App 声明了相同 `authorities` 的 Provider，第二个 App 在安装时会出现

`INSTALL_FAILED_CONFLICTING_PROVIDER` 错误。错误如下：


Application Installation Failed

Installation failed with message INSTALL_FAILED_CONFLICTING_PROVIDER.
It is possible that this issue is resolved by uninstalling an existing version of the apk if it is present, and then re-installing.

WARNING: Uninstalling will remove the application data!

Do you want to uninstall the existing application?

Provider 冲突

2.3、ContentProvider 类

一个具体的 ContentProvider 类需要继承自 `android.content.ContentProvider`， 并且实现 `onCreate / getType / insert / delete / update / query` 这几个方法。一般我们在 onCreate 方法中打开数据库，在对应的操作方法中根据 URI 的不同，操作不同的数据。

先定义一个帮助类，来声明一些常量。例如：

```java
public class CourseProviders {

    public static final String AUTHORITIES = "com.straw.course";
    public static final String COURSE_PATH = "course";


    public static final Uri BASE_URI = Uri.parse("content://" + AUTHORITIES);
    public static final Uri COURSE_URI = Uri.withAppendedPath(BASE_URI, COURSE_PATH);


    public static class CourseColumn {

        public static final String ID = "id";
        public static final String NAME = "NAME";
        public static final String TEACHER_NAME = "teacher_name";
        public static final String WATCH_COUNT = "watch_count";
        public static final String VIDEO_URL = "video_url";
    }
}
```

然后实现具体的 ContentProvider：

```java
public class CourseContentProvider extends ContentProvider {

    private CourseSqliteHelper mSqliteHelper;
    private SQLiteDatabase mDatabase;
    private UriMatcher mUriMatcher = new UriMatcher(UriMatcher.NO_MATCH);


    private static final int PROVIDE_COURSE = 1;


    @Override
    public boolean onCreate() {
        mSqliteHelper = new CourseSqliteHelper(getContext());
        mDatabase = mSqliteHelper.getWritableDatabase();

        mUriMatcher.addURI(CourseProviders.AUTHORITIES,
                CourseProviders.COURSE_PATH, PROVIDE_COURSE);
        return true;
    }

    @Nullable
    @Override
    public Cursor query(@NonNull Uri uri, String[] projection,
            String selection, String[] selectionArgs, String sortOrder) {

        switch (mUriMatcher.match(uri)) {
            case PROVIDE_COURSE:
                return mDatabase.query(CourseSqliteHelper.COURSE_TABLE_NAME,
                        projection, selection, selectionArgs, null, null, sortOrder);

            default:
                break;
        }

        return null;
    }

    @Nullable
    @Override
    public String getType(@NonNull Uri uri) {
        return null;
    }

    @Nullable
    @Override
    public Uri insert(@NonNull Uri uri, ContentValues values) {
        Uri result = null;
        switch (mUriMatcher.match(uri)) {
            case PROVIDE_COURSE:
                long rowId = mDatabase.insert(
                        CourseSqliteHelper.COURSE_TABLE_NAME, null, values);
                result = ContentUris.withAppendedId(uri, rowId);
                break;

            default:
                break;
        }

        return result;
    }

    @Override
    public int delete(@NonNull Uri uri, String selection, String[] selectionArgs) {
        switch (mUriMatcher.match(uri)) {
            case PROVIDE_COURSE:
                return mDatabase.delete(CourseSqliteHelper.COURSE_TABLE_NAME,
                        selection, selectionArgs);

            default:
                break;
        }

        return 0;
    }

    @Override
    public int update(@NonNull Uri uri, ContentValues values, String selection, String
        switch (mUriMatcher.match(uri)) {
            case PROVIDE_COURSE:
                return mDatabase.update(CourseSqliteHelper.COURSE_TABLE_NAME,
                        values, selection, selectionArgs);

            default:
                break;
        }

        return 0;
    }
}
```

通过 `android.content.UriMatcher` 提供的 `match` 方法，很方便的将 `content://authorities/path` 这样的访问 URI 匹配到正确的数据访问路径。

3、数据访问方

我们可以在本应用内通过 ContentProvider 访问，也可以在其他应用中访问。在 ContentProvider 所在应用外访问时，会拉起 ContentProvider 所在的 App（会拉起 Application，但不会打开任何 Activity）。

3.1、权限声明

首先，在使用方的 `AndroidManifest.xml` 中需要声明所使用 ContentProvider 的权限，如果只读就声明读权限，如果读写都需要就声明读写权限。如：

```
<uses-permission android:name="com.straw.providerhost.course.read"/>
<uses-permission android:name="com.straw.providerhost.course.write"/>
```

3.2、具体使用

```
private void queryAll() {
    Cursor cursor = context.getContentResolver().query(CourseProviders.COURSE_URI, nul
    if (cursor == null) {
        return;
    }

    mCourseInfoList.clear();
    while (cursor.moveToNext()) {
        CourseInfo info = new CourseInfo();
        info.mId = cursor.getString(cursor.getColumnIndex(CourseProviders.CourseColumn
        info.mName = cursor.getString(cursor.getColumnIndex(CourseProviders.CourseColu
        info.mTeacherName = cursor.getString(cursor.getColumnIndex(CourseProviders.Cou
        info.mWatchCount = cursor.getInt(cursor.getColumnIndex(CourseProviders.CourseC
        info.mVideoUrl = cursor.getString(cursor.getColumnIndex(CourseProviders.Course

        mCourseInfoList.add(info);
    }

    cursor.close();

    mAdapter.notifyDataSetChanged();
}
```

通过 `context.getContentResolver` 方法获取 `ContentResolver`，然后使用其提供的 `insert` / `delete` / `update` / `query` 方法进行 `增删改查` 操作即可。

6人点赞 >   Android