



ANDROID DEVELOPMENT

CONTENT PROVIDER

Tan Cher Wah (isstcw@nus.edu.sg)



Agenda

- Content Provider
- File Provider
 - File Sharing
 - Content URI
 - Permission



Content Provider

- A Content Provider has data
 - Contacts
 - Calendar
 - Files
- Shares data across applications
- Various Providers in Android
 - Contacts Provider
 - Calendar Provider
 - File Provider



- URI - Universal Resource Identifier
 - Represents a unique resource
 - Is a superset of URL where protocols are prefixed –
http://, ftp://, mailto://
- A **Content URI** is used to identify data in a provider
 - Has the form - **content://<path>**
 - Via the Debugger, the content URI for our video is:
content://com.cherwah.videoexample.provider/internal/mov_bbb.mp4



Play Video example

- Example usage for FileProvider
 - Play a video residing in our **internal storage**
 - Use an **Implicit** intent
 - Use **FileProvider** to share our file with a Video Player (the Player is not part of our app)



Steps to use the FileProvider

- Specify the FileProvider in AndroidManifest.xml
 - Provide all required **attributes**
 - Specify an **XML file** that contains the shared paths
- Specify actual **Shared Paths** in that XML file
- Invoke the FileProvider in your code



Add FileProvider to AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.cherwah.videoexample">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".VideoActivity"/>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <provider
            android:name="android.support.v4.content.FileProvider"
            android:authorities="com.cherwah.videoexample.provider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/file_paths">
            </meta-data>
        </provider>

    </application>
</manifest>
```



FileProvider attributes

AndroidManifest.xml

android:name	Set to “android.support.v4.content.FilePro vider”
android:authorities	Need to define one unique authority. Use your package name.
android:exported	False
android:grantUriPermissions	True
android:resource	Specify XML file to paths



Specify paths in XML folder

VideoExample [D:\AndroidStudioProjects\VideoExample] - ...\app\src\main\res\xml\file_paths.xml [app] - Android Studio

Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

VideoExample > app > src > main > res > xml > file_paths.xml

Android > MainActivity.java > VideoActivity.java > AndroidManifest.xml > file_paths

app

- manifests
- java
- generatedJava
- res
 - drawable
 - layout
 - mipmap
 - values
 - xml

file_paths.xml

Gradle Scripts

- build.gradle (Project: VideoExample)
- build.gradle (Module: app)
- gradle-wrapper.properties (Gradle)
- proguard-rules.pro (ProGuard Rule)
- gradle.properties (Project Properties)
- settings.gradle (Project Settings)
- local.properties (SDK Location)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <paths xmlns:android="http://schemas.android.com/apk/res/android">
3   <file-path name="internal" path="." />
4 </paths>
```



FileProvider attributes

Shared Paths

Attribute	Storage	Directory
<files-path>	internal app storage	getFilesDir() = data/data/<package_name>
<external-path>	public external storage	Environment.getExternalStorageDirectory() = /mnt/sdcard/data/data/<package_name>
<external-files-path>	external app storage	getExternalFilesDir() = /mnt/sdcard



Invoking FileProvider

- Request runtime permissions
- Set up FileProvider

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Intent intent = new Intent();
    intent.setAction(android.content.Intent.ACTION_VIEW);

    ActivityCompat.requestPermissions(this,
        new String[]{
            Manifest.permission.READ_EXTERNAL_STORAGE,
            Manifest.permission.WRITE_EXTERNAL_STORAGE}, 1);

    File file = new File(getFilesDir() + "/mov_bbb.mp4");
    Uri uri = FileProvider.getUriForFile(this,
        "com.cherwah.videoexample.provider", file);
    intent.setDataAndType(uri, "video/*");
    intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);

    if (intent.resolveActivity(getPackageManager()) != null)
        startActivity(intent);
}
```



Playing the Video

Device File Explorer

Emulator Pixel_2_API_28 Android 9, API 28

Name	Per...	Date	Size
data	drwxrwx	2019-05-23 06:13	4 KB
app	drwxrwx	2019-05-23 06:13	4 KB
data	drwxrwx	2019-05-23 06:13	4 KB
com.android.wallpaper.livepicker	drwxrwx	2019-05-23 06:13	4 KB
com.android.wallpaperbackup	drwxrwx	2019-05-23 06:13	4 KB
com.breel.geswallpapers	drwxrwx	2019-05-23 06:13	4 KB
com.cherwah.checkfiles	drwxrwx	2019-05-23 06:13	4 KB
com.cherwah.datapersistence	drwxrwx	2019-05-23 06:13	4 KB
com.cherwah.mediaexample	drwxrwx	2019-05-23 06:13	4 KB
com.cherwah.videoexample	drwxrwx	2019-05-23 06:13	4 KB
cache	drwxrwx	2019-06-10 03:35	4 KB
code_cache	drwxrwx	2019-06-10 03:35	4 KB
files	drwxrwx	2019-06-10 03:57	4 KB
videos	drwxrwx	2019-06-10 05:00	4 KB
mov_bbb.mp4	-rwxrwx	2019-06-10 05:03	770 KB
com.google.android.apps.docs	drwxrwx	2019-05-23 06:13	4 KB
com.google.android.apps.inputm	drwxrwx	2019-05-23 06:13	4 KB
com.google.android.apps.maps	drwxrwx	2019-05-23 06:13	4 KB
com.google.android.apps.messag	drwxrwx	2019-05-23 06:13	4 KB
com.google.android.apps.nexusla	drwxrwx	2019-05-23 06:13	4 KB





References

- FileProvider -
<https://developer.android.com/reference/android/support/v4/content/FileProvider>
- Sharing files using FileProvider -
<https://infinum.co/the-capsized-eight/share-files-using-fileprovider>
- Open a file in another app using FileProvider -
<https://medium.com/@ali.dev/open-a-file-in-another-app-with-android-fileprovider-for-android-7-42c9abb198c1>



Active Recall

FileProvider
Share Files
getUriForFile
resolveActivity
Runtime Permission
AndroidManifest.xml
Content URI
content://<path>
FLAG_GRANT_READ_URI_PERMISSION