





ANDROID DEVELOPMENT

CONTENT PROVIDER

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







- 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/interna l/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"</pre>
  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:supportsRtl="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>
     ovider
       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>
```







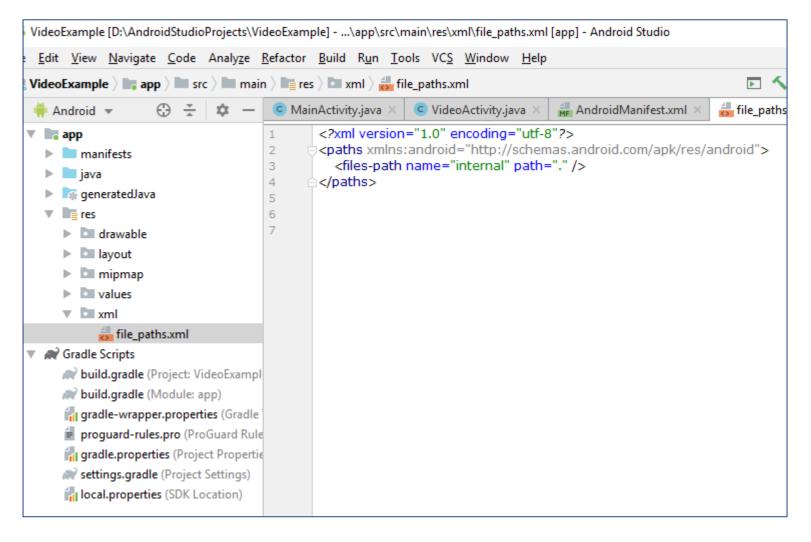
		AndroidManifest.xml
android:name	Set to "android.suppo vider"	ort.v4.content.FilePro
android:authorities	Need to define authority. Use	e one unique your package name.
android:exported	False	
android:grantUriPermissions	True	
android:resource	Specify XML fi	le to paths



Specify paths in XML folder













Shared Paths

Attribute	Storage	Directory
<files-path></files-path>	internal app storage	getFilesDir() =
		data/data/ <package_name></package_name>
<external-path></external-path>	public external storage	Environment.getExternalSto rageDirectory() =
		/mnt/sdcard/data/data/ <pac kage_name></pac
<external-files-path></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);
}
```

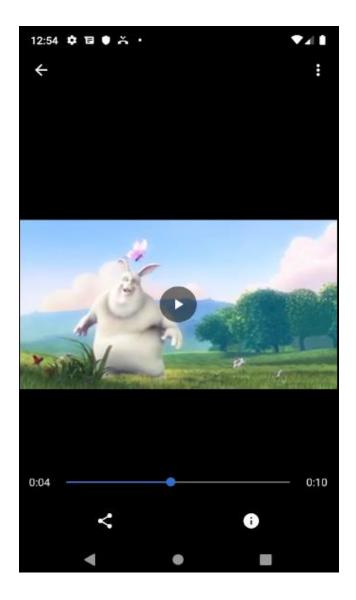


Playing the Video





Device File Explorer			ф –
Emulator Pixel_2_API_28 Android 9, API 28			*
Name	Per	Date	Size
▼ ■ data	drwxrw	2019-05-23 06:13	4 KB
▶ app		2019-05-23 06:13	4 KB
data		2019-05-23 06:13	4 KB
com.android.wallpaper.livepicker			4 KB
com.android.wallpaperbackup	drwxrw	2019-05-23 06:13	4 KB
com.breel.geswallpapers	drwxrw	2019-05-23 06:13	4 KB
com.cherwah.checkfiles	drwxrw	2019-05-23 06:13	4 KB
com.cherwah.datapersistence	drwxrw	2019-05-23 06:13	4 KB
com.cherwah.mediaexample	drwxrw	2019-05-23 06:13	4 KB
com.cherwah.videoexample	drwxrw	2019-05-23 06:13	4 KB
cache	drwxrw	2019-06-10 03:35	4 KB
code_cache	drwxrw	2019-06-10 03:35	4 KB
▶ I III files	drwxrw	2019-06-10 03:57	4 KB
▶ I videos	drwxrw	2019-06-10 05:00	4 KB
🐈 mov_bbb.mp4	-rwxrw:	2019-06-10 05:03	770 KB
com.google.android.apps.docs	drwxrw	2019-05-23 06:13	4 KB
com.google.android.apps.inputm	drwxrw	2019-05-23 06:13	4 KB
com.google.android.apps.maps	drwxrw	2019-05-23 06:13	4 KB
com.google.android.apps.messag	drwxrw	2019-05-23 06:13	4 KB
com.google.android.apps.nexusla			4 KB









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







