

Manejo de Video y Almacenamiento Externo en Android con Java

Guardar el video en la Memoria Externa, a diferencia de la Interna, requiere **Permisos Peligrosos** y tiene implicaciones en la privacidad y la persistencia de los datos.

1. Permisos y Configuración Esencial

Para esta operación, necesitas los permisos para la cámara y, crucialmente, el permiso de escritura para guardar el archivo en el almacenamiento compartido.

A. AndroidManifest.xml

XML

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
    android:maxSdkVersion="28" />

<application ...>
    <provider
        android:name="androidx.core.content.FileProvider"
        android:authorities="${applicationId}.provider"
        android:exported="false"
        android:grantUriPermissions="true">
        <meta-data
            android:name="android.support.FILE_PROVIDER_PATHS"
            android:resource="@xml/file_paths_external" />
    </provider>
</application>
```

B. Permisos en Tiempo de Ejecución (Runtime)

Debes solicitar el permiso **CAMERA** y **WRITE_EXTERNAL_STORAGE** (si la targetSdkVersion es 28 o inferior) al usuario antes de

llamar a la cámara.

2. Configurar FileProvider para Memoria Externa

El archivo XML debe mapear un directorio en la Memoria Externa donde se guardará el video, preferiblemente el directorio estándar de videos.

res/xml/file_paths_external.xml (Asegúrate de que el nombre del recurso coincida con el *Manifest*):

XML

```
<?xml version="1.0" encoding="utf-8"?>
<paths>
  <external-files-path
    name="external_videos"
    path="Movies" />
</paths>
```

3. Preparar el Fichero de Destino en la Memoria Externa

Creamos el archivo en un directorio específico de tu app dentro de la Memoria Externa, usando **Environment.DIRECTORY_MOVIES**.

Java

```
// MainActivity.java
private String currentVideoPath;

private File crearArchivoDeVideoExterno() throws IOException {
  // 1. Crea un nombre de archivo único.
```

```

String timeStamp = new SimpleDateFormat("yyyyMMdd_HH:mm:ss", Locale.getDefault()).format(new Date());
String videoFileName = "VID_" + timeStamp + ".mp4";

// 2. Obtiene el directorio de destino: getExternalFilesDir(Environment.DIRECTORY_MOVIES)
// El video se guardará en /Android/data/com.tudominio.tuapp/files/Movies/
File storageDir = getExternalFilesDir(Environment.DIRECTORY_MOVIES);

// 3. Crea el archivo temporal con extensión .mp4.
File video = File.createTempFile(
    videoFileName, /* Prefijo */
    ".mp4", /* Sufijo */
    storageDir /* Directorio */
);

currentVideoPath = video.getAbsolutePath();
return video;
}

```

4. Llamar a la Grabadora de Video usando Intent

Utilizamos la acción **MediaStore.ACTION_VIDEO_CAPTURE** y el FileProvider para generar la URI de destino.

Java

```

private static final int REQUEST_VIDEO_CAPTURE = 4;

private void dispatchTakeVideoIntentExterno() {
    // 1. Verificar Permisos de CÁMARA y ESCRITURA aquí.

    Intent takeVideoIntent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);

```

```
if (takeVideoIntent.resolveActivity(getPackageManager()) != null) {
    File videoFile = null;
    try {
        // Llama a la función que crea el archivo en Memoria Externa.
        videoFile = crearArchivoDeVideoExterno();
    } catch (IOException ex) {
        Log.e("VideoCapture", "Error creando archivo de video: " + ex.getMessage());
    }

    if (videoFile != null) {
        // 2. Obtiene la URI del archivo a través de FileProvider.
        Uri videoURI = FileProvider.getUriForFile(this,
            getApplicationContext().getPackageName() + ".provider",
            videoFile);

        // 3. Pasa la URI al Intent para que la cámara guarde allí el video.
        takeVideoIntent.putExtra(MediaStore.EXTRA_OUTPUT, videoURI);

        // Opcional: Limitar el tamaño o la calidad del video.
        // takeVideoIntent.putExtra(MediaStore.EXTRA_VIDEO_QUALITY, 1); // 1 = Alta calidad

        // 4. Lanza el Intent.
        startActivityForResult(takeVideoIntent, REQUEST_VIDEO_CAPTURE);
    }
}
```

5. Manejar el Resultado y Notificar a la Galería (Opcional)

El video ya está guardado en la Memoria Externa. Si lo guardaste en un directorio público estándar (como el que se obtendría con `Environment.getExternalStoragePublicDirectory(...)` -aunque está obsoleto-), debes notificar al sistema para que aparezca en la galería. Si usaste **`getExternalFilesDir()`**, esto no es estrictamente necesario, pero es útil si necesitas escanearlo.

Java

`@Override`

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == REQUEST_VIDEO_CAPTURE && resultCode == RESULT_OK) {
        // El video está guardado de forma persistente en: currentVideoPath.

        Toast.makeText(this, "Video guardado en Memoria Externa.", Toast.LENGTH_LONG).show();

        // Si quieres que la galería lo muestre inmediatamente:
        galleryAddVideo(currentVideoPath);
    }
}

/**
 * Notifica al sistema para que indexe el archivo de video.
 * Esto es necesario si guardas en un directorio público.
 */
private void galleryAddVideo(String videoPath) {
    Intent mediaScanIntent = new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);
    File f = new File(videoPath);
    Uri contentUri = Uri.fromFile(f);
    mediaScanIntent.setData(contentUri);
    this.sendBroadcast(mediaScanIntent);
}
```

```
}
```

6. Consideraciones Avanzadas (Android 10+)

Para aplicaciones con targetSdkVersion **29 (Android 10)** y superior que desean guardar el video en un **directorio público** (Movies/), ya no se usa WRITE_EXTERNAL_STORAGE. En su lugar, debes usar la **MediaStore API** para insertar el video.

- **Proceso con MediaStore:**

1. Crea una entrada de ContentValues con metadatos (título, fecha, MIME type).
2. Obtén una Uri de la colección MediaStore.Video.Media.EXTERNAL_CONTENT_URI.
3. Pasa esta Uri al Intent de la cámara usando MediaStore.EXTRA_OUTPUT.

Este método es más complejo en Java puro pero es la vía moderna y segura para interactuar con almacenamiento compartido.

Fuentes

1. <https://github.com/NormanGadenya/Let-s-Talk-App>