



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Lập trình Android

Bài 26: Đa phương tiện - Multimedia

Phòng LT & Mạng

<http://csc.edu.vn/lap-trinh-va-csdl>





Nội dung

1. Chụp ảnh sử dụng Camera
2. Truy cập ảnh từ Gallery
3. Play audio
4. Thu âm audio
5. Play video
6. Ghi hình video



Sử dụng Camera

- Minh họa sau hướng dẫn cách sử dụng **Camera** để chụp ảnh, sử dụng **Intent**
- **Bước 1: Thiết lập FileProvider**
 - Khai báo sử dụng **FileProvider** trong AndroidManifest.xml

```
<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="edu.csc.camera.fileprovider"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/fileprovider" />
</provider>
```

- Với file **res/xml/fileprovider** như sau:

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



Sử dụng Camera (2)

- **Bước 2:** Sử dụng **Capture Intent**

```
public final static int CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE = 1001;

private File photoFile;

private void onLaunchCamera() {
    // tạo Intent để truy cập Camera
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

    // tạo file lưu hình ảnh và truy xuất trong tương lai
    String photoFileName = "IMG_" + System.currentTimeMillis();
    photoFile = getPhotoFileUri(photoFileName);

    // bao bọc file bởi ContentProvider
    Uri fileProvider = FileProvider.getUriForFile(this, "edu.csc.camera.fileprovider", photoFile);
    intent.putExtra(MediaStore.EXTRA_OUTPUT, fileProvider);
    if (intent.resolveActivity(getPackageManager()) != null) {
        // thực hiện Intent để chụp ảnh
        startActivityForResult(intent, CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE);
    }
}
```





Sử dụng Camera (2)

- **Bước 2** (tiếp): Tạo file để lưu hình ảnh chụp được

```
public File getPhotoFileUri(String fileName) {  
    // Dùng "getExternalFilesDir" để không cần request external read/write runtime permissions  
    File mediaStorageDir = new File(getExternalFilesDir(Environment.DIRECTORY_PICTURES), "TAG");  
    // Tạo thư mục chứa hình ảnh nếu chưa tồn tại  
    if (!mediaStorageDir.exists() && !mediaStorageDir.mkdirs()){  
        Log.d("TAG", "failed to create directory");  
    }  
    // Trả về đối tượng file tương ứng với file name  
    File file = new File(mediaStorageDir.getPath() + File.separator + fileName);  
    return file;  
}
```



Sử dụng Camera (3)

- **Bước 3:** Xử lý kết quả trả về từ Camera app

`@Override`

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if (requestCode == CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE) {  
        if (resultCode == RESULT_OK) {  
            // tại thời điểm này, đã có hình ảnh từ camera lưu trên file  
            Bitmap takenImage = BitmapFactory.decodeFile(photoFile.getAbsolutePath());  
            // RESIZE BITMAP, để tránh lỗi OutOfMemoryError khi hiển thị ảnh lên UI  
            Bitmap resizedBitmap = BitmapScaler.scaleToFitWidth(takenImage, 200);  
            // Hiển thị hình ảnh vừa chụp lên image view  
            ivPreview.setImageBitmap(takenImage);  
        } else { // Xử lý cho trường hợp không chụp ảnh (chọn Cancel ở Camera app)  
            Toast.makeText(this, "Picture wasn't taken!", Toast.LENGTH_SHORT).show();  
        }  
    }  
}
```

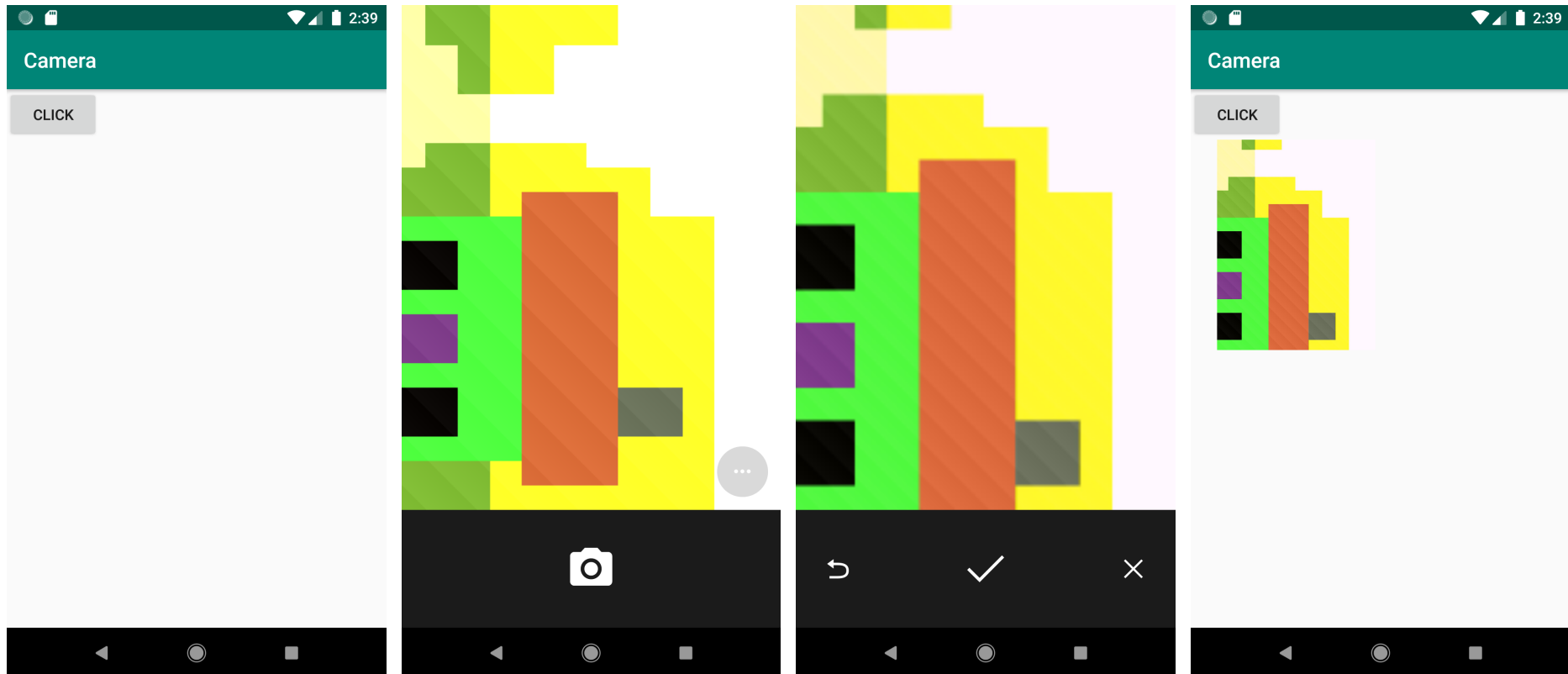


BitmapScaler: Tham khảo tại <https://gist.github.com/nesquena/3885707fd3773c09f1bb>



Sử dụng Camera (4)

- Kết quả thử nghiệm:





Truy cập Gallery

- **Bước 1:** Sử dụng Pick Intent

```
public final static int PICK_PHOTO_REQUEST_CODE = 1046;
```

```
private void onPickPhoto() {
```

```
    // Tạo Intent để chọn hình ảnh từ Gallery
```

```
    Intent intent = new Intent(Intent.ACTION_PICK,  
        MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
```

```
    if (intent.resolveActivity(getPackageManager()) != null) {
```

```
        // thực hiện Intent để chọn ảnh
```

```
        startActivityForResult(intent, PICK_PHOTO_REQUEST_CODE);
```

```
    }
```

```
}
```




Truy cập Gallery (2)

- **Bước 2:** Xử lý kết quả trả về **Gallery** app

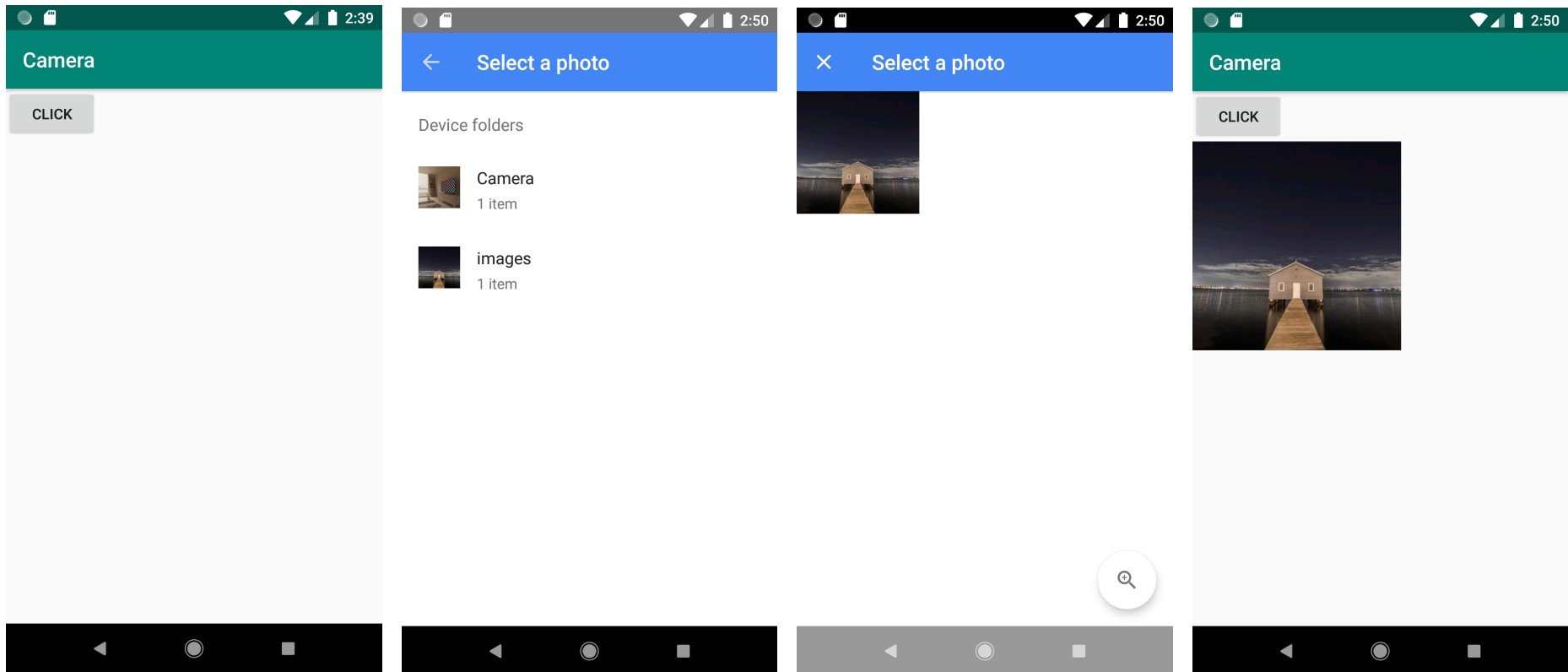
@Override

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if (requestCode == PICK_PHOTO_REQUEST_CODE) {  
        if (resultCode == RESULT_OK) {  
            if (data != null) {  
                Uri photoUri = data.getData();  
                // Lấy uri của hình ảnh từ data  
                Bitmap selectedImage = null;  
                try { // Tạo đối tượng bitmap tương ứng với uri  
                    selectedImage = MediaStore.Images.Media.getBitmap(this.getContentResolver(), photoUri);  
                } catch (IOException e) {  
                    e.printStackTrace();  
                }  
                // Hiển thị hình ảnh (bitmap) được chọn lên image view  
                if (selectedImage != null) {  
                    ivPreview.setImageBitmap(selectedImage);  
                }  
            }  
        }  
    }  
}
```



Truy cập Gallery (3)

- Kết quả thử nghiệm:





Audio & Video

- Android hỗ trợ những định dạng audio phổ biến như mp3, wav, acc, flac... (xem thêm tại <https://developer.android.com/guide/topics/media/media-formats>)
- Có thể play audio & video từ nhiều nguồn khác nhau:
 - Raw resources
 - File trong hệ thống
 - Từ network (streaming)
- Để play audio & video trong ứng dụng, có thể sử dụng class MediaPlayer
- Để thu âm & thu hình trong ứng dụng, có thể sử dụng class MediaRecorder



MediaPlayer - Play audio từ Raw resources file

- Tạo đối tượng **MediaPlayer** sử dụng phương thức **MediaPlayer.create()** và sử dụng phương thức **start()** để bắt đầu play audio

```
MediaPlayer mediaPlayer = MediaPlayer.create(this, R.raw.sample);  
mediaPlayer.start();
```

- Để dừng, sử dụng phương thức **stop()**. Nếu muốn replay, cần phải gọi phương thức **reset()** và **prepare()** sau đó gọi lại phương thức **start()**

```
if (mediaPlayer.isPlaying())  
    mediaPlayer.stop();
```

- Để tạm dừng, sử dụng phương thức **pause()**. Nếu muốn tiếp tục play, gọi phương thức **start()**
- Các phương thức khác của MediaPlayer:
 - **seekTo(...)** - di chuyển tới vị trí millisecond của audio
 - **getCurrentDuration()** – lấy vị trí hiện tại của audio theo milliseconds
 - **getDuration()** – lấy tổng độ dài thời gian của audio



MediaPlayer - Play audio từ file trong hệ thống

- Tạo đối tượng **MediaPlayer** bằng phương thức khởi tạo
- Gọi phương thức **setDataSource** tương ứng với chuỗi tên file audio
- Gọi phương thức **prepare()** sau đó gọi phương thức **start()**

```
try {  
    String filePath = "/sdcard/my_favorite_song.mp3";  
    MediaPlayer mp = new MediaPlayer();  
    mp.setDataSource(filePath);  
    mp.prepare();  
    mp.start();  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

- Phương thức **stop()** và **pause()** hoạt động giống như với Raw file
- Lưu ý, cần khai báo và request runtime permission **READ_EXTERNAL_STORAGE**

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```



MediaPlayer – Streaming audio từ Network

- Khai báo sử dụng kết nối **Internet**:

```
<uses-permission android:name="android.permission.INTERNET" />
```

- **Streaming** audio file với những định dạng được Android hỗ trợ:

```
String url = "https://file-examples.com/wp-content/uploads/2017/11/file_example_MP3_700KB.mp3";
final MediaPlayer mediaPlayer = new MediaPlayer();
mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC); // thiết lập loại audio là streaming
mediaPlayer.setOnErrorListener(new MediaPlayer.OnErrorListener() { // nếu audio không thể prepare
    @Override
    public boolean onError(MediaPlayer mp, int what, int extra) {
        return false; // MediaPlayer chuyển sang trạng thái Lỗi → Cần phải reset
    }
});
mediaPlayer.setOnPreparedListener(new MediaPlayer.OnPreparedListener() { // nếu prepared được audio
    @Override
    public void onPrepared(MediaPlayer mp) {
        mediaPlayer.start(); // bắt đầu play audio khi prepared xong
    }
});
try {
    mediaPlayer.setDataSource(url); // thiết lập nguồn dữ liệu tới URL của audio file
} catch (IOException e) {
    e.printStackTrace();
}
mediaPlayer.prepareAsync(); // Trigger prepare streaming audio file
```



Thu âm audio

- Để thu âm audio, cần khai báo các permissions và xin phép người dùng lúc runtime:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

- Thực hiện thu âm:

```
PackageManager manager = this.getPackageManager(); // kiểm tra thiết bị có microphone hay không
if (manager.hasSystemFeature(PackageManager.FEATURE_MICROPHONE)) {
    // thiết lập đường dẫn file audio
    String fileName = Environment.getExternalStorageDirectory().getAbsolutePath() + "/audiorecordtest.3gp";
    MediaRecorder mediaRecorder = new MediaRecorder(); // tạo đối tượng recorder
    // thiết lập audio format and encoder
    mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
    mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
    mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
    mediaRecorder.setOutputFile(fileName); // thiết lập đường dẫn file output
    try { // bắt đầu thu âm
        mediaRecorder.prepare();
        mediaRecorder.start();
    } catch (IOException e) {
        Log.e("TAG", "prepare() failed");
    }
} else { // thiết bị không có microphone
    Toast.makeText(this, "This device doesn't have a mic!", Toast.LENGTH_LONG).show();
}
```





Thu âm audio (2)

- Dừng thu âm:

```
mediaRecorder.stop();
```

```
mediaRecorder.reset();
```

```
mediaRecorder.release();
```




Play video

- Android hỗ trợ play các file có định dạng như MP4...
(<https://developer.android.com/guide/topics/media/media-formats>) sử dụng **VideoView**.
Thêm VideoView vào layout:

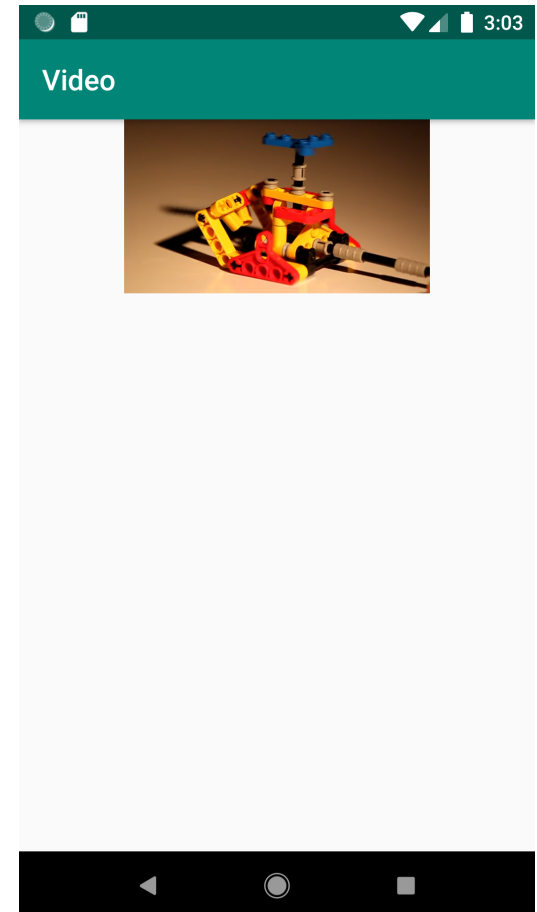
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <VideoView
        android:id="@+id/videoView"
        android:layout_width="640px"
        android:layout_height="480px" />

</LinearLayout>
```

- Play video file từ Raw resource:

```
videoView.setVideoURI(Uri.parse("android.resource://"
    + getPackageName() + "/" + R.raw.small));
videoView.setMediaController(new MediaController(this));
videoView.requestFocus();
videoView.start();
```





Streaming video

- Có thể **streaming** các file video mà Android hỗ trợ dùng **MediaView**:

```
videoView.setVideoPath("http://techslides.com/demos/sample-videos/small.mp4");
MediaController mediaController = new MediaController(this);
mediaController.setAnchorView(videoView);
videoView.setMediaController(mediaController);
videoView.requestFocus();
videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
    public void onPrepared(MediaPlayer mp) {
        videoView.start();
    }
});
```

- Lưu ý: khai báo sử dụng **permission** cho phép truy cập **Internet**

```
<uses-permission android:name="android.permission.INTERNET" />
```



Thu hình video

- Có thể sử dụng Intent để thu hình video sử dụng Camera. Đầu tiên cần khai báo và request runtime permission `WRITE_EXTERNAL_STORAGE`

```
private static final int VIDEO_CAPTURE_REQUEST = 1011;
private Uri videoUri; // đường dẫn file kết quả

if (getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA_FRONT)) {
    Intent intent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
    File mediaFile = new File(
        Environment.getExternalStorageDirectory().getAbsolutePath() + "/my_video.mp4");
    videoUri = Uri.fromFile(mediaFile);
    intent.putExtra(MediaStore.EXTRA_OUTPUT, videoUri);
    startActivityForResult(intent, VIDEO_CAPTURE_REQUEST);
} else {
    Toast.makeText(this, "No camera on device", Toast.LENGTH_LONG).show();
}
```



Thu hình video (2)

- Cần override phương thức **onActivityResult** để xử lý kết quả trả về. Lưu ý: để play được video từ **Uri**, cần thiết lập cấu hình **FileProvider** tương tự khi chụp ảnh.

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if (requestCode == VIDEO_CAPTURE_REQUEST) {  
        if (resultCode == RESULT_OK) {  
            Log.i("TAG", "Video has been saved to: " + data.getData());  
            videoView.setVideoURI(videoUri);  
            videoView.setMediaController(new MediaController(this));  
            videoView.requestFocus();  
            videoView.start();  
        } else if (resultCode == RESULT_CANCELED) {  
            Log.i("TAG", "Video recording cancelled.");  
        } else {  
            Log.i("TAG", "Failed to record video");  
        }  
    }  
}
```

