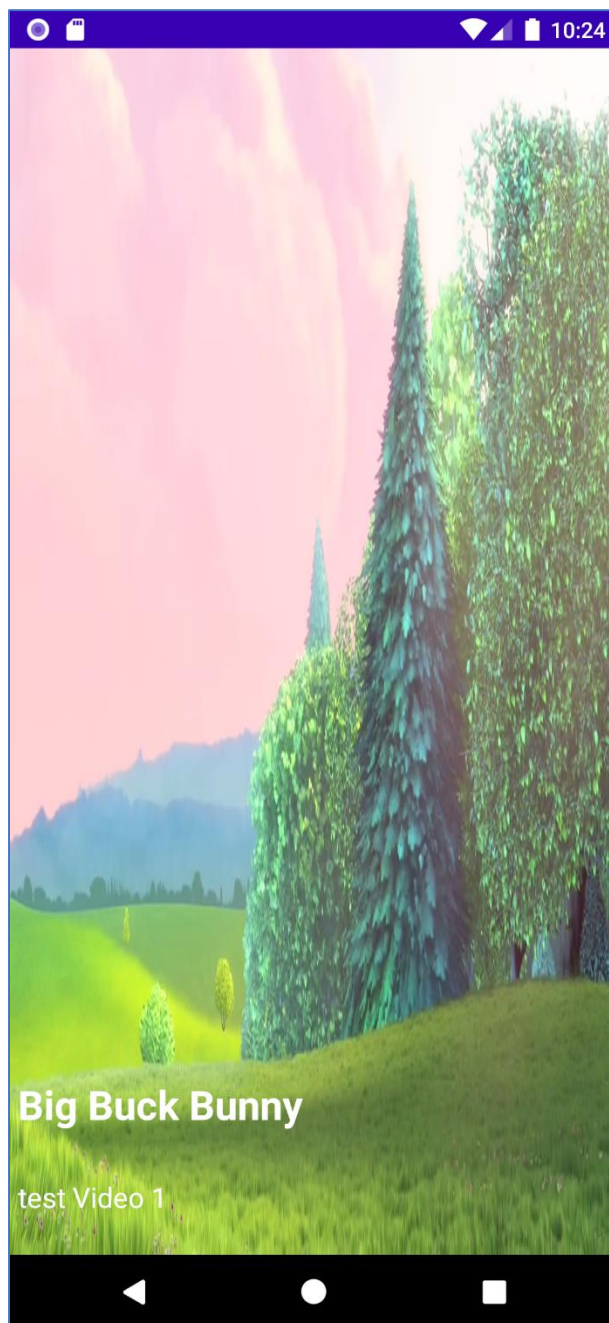


Lab04: VideoView + ViewPager2



Bước 0:

- Thêm quyền truy cập INTERNET cho ứng dụng ở Manifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET" />
</manifest>

<application
```

- Chuẩn bị file .mp4 và copy vào res/**raw**: Thư mục dùng để chứa đựng các file mà bạn được phép định nghĩa nội dung tùy thích, không bị gò bó phải là XML hay Java code gì cả.)

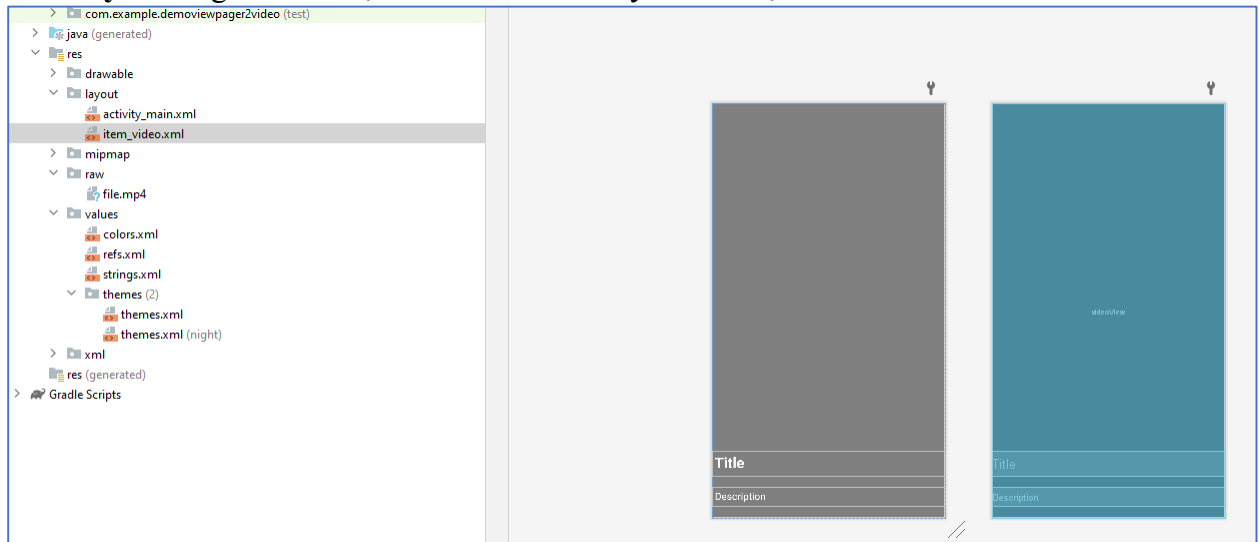
Bước 1: Thiết kế giao diện sử dụng

- ✓ **Giao diện MainActivity**: Sử dụng **ViewPager2** (Thuộc tính để trượt xuống-lên sẽ **android:orientation="vertical"**)

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/black"
tools:context=".MainActivity">
<androidx.viewpager2.widget.ViewPager2
    android:id="@+id/viewPager2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- ✓ Xây dựng Layout cho **item_video** (Gồm **VideoView**, **TextView** cho Title , **TextView** cho Description)

ở **res/layout** Right click chọn **New/Resource Layout** file đặt tên **"item_video.xml"**



```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent">

<VideoView
    android:id="@+id/videoView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

</VideoView>

<TextView
    android:id="@+id/textVideoTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    android:padding="5dp"
    android:textColor="@color/white"
    android:text="Title"
    android:textSize="26sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/textVideoDescription"
    app:layout_constraintEnd_toEndOf="parent" />

<TextView
    android:id="@+id/textVideoDescription"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    android:padding="5dp"
    android:shadowColor="@color/black"
    android:text="Description"
    android:textColor="@color/white"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Bước 2: Tạo Java class “VideoItem” chứa thông tin và hàm cung cấp **List<VideoItem>** có sẵn (ở video local thì mình phải đổi sang đường dẫn android.resource://.)

```
public class VideoItem {
    public String Url;
    public String Title;
    public String Description;

    public VideoItem(String url, String title, String description) {
        Url = url;
        Title = title;
        Description = description;
    }
    public static List<VideoItem> GetList(Context context)
    {
        List<VideoItem> list = new ArrayList<VideoItem>();

        //from Raw folder
        VideoItem itemVideo1= new VideoItem("file.mp4", "From RAW folder",
"test Video 0");
        itemVideo1.Url = "android.resource://" + context.getPackageName() +
"/" + R.raw.file1;
        list.add(itemVideo1);

        //From the Internet URL
        list.add(new VideoItem(
"http://commondatastorage.googleapis.com/gtv-videos-
bucket/sample/BigBuckBunny.mp4", "Big Buck Bunny", "test Video 1"));
        list.add(new VideoItem("http://commondatastorage.googleapis.com/gtv-
```

```
videos-bucket/sample/ElephantsDream.mp4", "Elephants Dream", "test Video 2"));  
    list.add(new VideoItem("http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ForBiggerBlazes.mp4", "For Bigger Blazes", "test Video 3"));  
    list.add(new VideoItem("http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ForBiggerEscapes.mp4", "For Bigger Escapes", "test Video 4"));  
    return list;  
}  
}
```

Bước 3: (Tương tự cách làm để đổ dữ liệu cho RecyclerView)

- Tạo Adapter để render từng item_Video và ViewHolder tương ứng

Tạo lớp VideoAdapter **extends**

RecyclerView.Adapter<VideoAdapter.VideoViewHolder>

Như vậy sẽ có:

VideoViewHolder là 1 **ViewHolder**: nhằm để tùy chỉnh một View theo ý, giúp tránh việc phải thực hiện findViewById() và làm cho ứng dụng mượt hơn.

```
public class VideoAdapter extends RecyclerView.Adapter<VideoAdapter.VideoViewHolder> {  
    List<VideoItem> listVideos;  
  
    public VideoAdapter(List<VideoItem> list) {  
        this.listVideos = list;  
    }  
  
    @NonNull  
    @Override  
    public VideoViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
        return new VideoViewHolder(LayoutInflater.from(parent.getContext()).inflate(R.layout.item_video, parent, attachToRoot: false));  
    }  
  
    @Override  
    public void onBindViewHolder(@NonNull VideoViewHolder holder, int position) {  
        VideoItem item = listVideos.get(position);  
        holder.txtVideoTitle.setText(item.Title);  
        holder.txtVideoDescription.setText(item.Description);  
        holder.videoView.setVideoPath(item.Url);  
        holder.videoView.start();  
    }  
  
    @Override  
    public int getItemCount() {  
        return listVideos.size();  
    }  
}  
  
static class VideoViewHolder extends RecyclerView.ViewHolder {  
    VideoView videoView;  
    TextView txtVideoTitle, txtVideoDescription;  
  
    public VideoViewHolder(@NonNull View itemView) {  
        super(itemView);  
        videoView = itemView.findViewById(R.id.videoView);  
        txtVideoTitle = itemView.findViewById(R.id.txtVideoTitle);  
        txtVideoDescription = itemView.findViewById(R.id.txtVideoDescription);  
    }  
}
```

Bước 4: Sử dụng với ViewPager2

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        ViewPager2 viewPager = findViewById(R.id.viewPager2);  
        viewPager.setAdapter(new VideoAdapter(VideoItem.getList(getBaseContext())));  
    }  
}
```

Chạy thử (Do kết nối mạng có thể chậm khi load Videos)

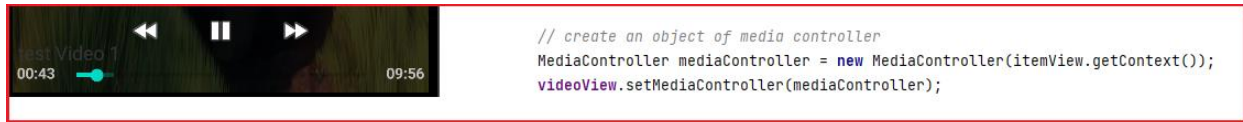
Người ta thường thêm **ProgressBar** để hiển thị Loading... cho người dùng 😊 (Bước 7)

Bước 5: Để Play/Stop khi touch vào VideoView. Thêm sự kiện OnClickListener cho VideoView

```
static class VideoViewHolder extends RecyclerView.ViewHolder {  
    VideoView videoView;  
    TextView txtVideoTitle, txtVideoDescription;  
  
    public VideoViewHolder(@NonNull View itemView) {  
        super(itemView);  
        videoView = itemView.findViewById(R.id.videoView);  
        txtVideoTitle = itemView.findViewById(R.id.txtVideoTitle);  
        txtVideoDescription = itemView.findViewById(R.id.txtVideoDescription);  
  
        videoView.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                if(videoView.isPlaying())  
                    videoView.pause();  
                else  
                    videoView.start();  
            }  
        });  
    }  
}
```

Bước 6: Hiện thị Seeking trên VideoView

- ✓ Thêm Mediacontroller ở ViewHolder



Bước 7:

- Kéo ProgressBar (**vertical**) vào item_video.xml và thực hiện ánh xạ view tương ứng ở VideoViewHolder
- Khi Video chuẩn bị xong (bắt đầu play được) thì Tắt (invisible) progressBar



-----Chúc các bạn thành công !-----