**Movie App with Retrofit and Navigation**

Here’s a step-by-step real-time case study for creating a **Movie App** in Android using **Retrofit** for network calls and basic **navigation**. We'll use **Kotlin**, **Fragment**, and **Navigation Component** for managing navigation.

**Features of the Movie App**

1. Display a list of movies fetched from a public API.
2. Show movie details on a separate screen when a movie is selected.

**Prerequisites**

* Android Studio setup.
* A free movie API (e.g., [The Movie Database (TMDB)](https://www.themoviedb.org/)).
* Basic knowledge of Kotlin, Fragments, and Navigation.

**Steps to Build the Movie App**

**1. Setup Dependencies**

In your build.gradle file:

groovy

CopyEdit

dependencies {

implementation "com.squareup.retrofit2:retrofit:2.9.0"

implementation "com.squareup.retrofit2:converter-gson:2.9.0"

implementation "androidx.navigation:navigation-fragment-ktx:2.7.0"

implementation "androidx.navigation:navigation-ui-ktx:2.7.0"

implementation "com.google.android.material:material:1.9.0"

implementation "androidx.recyclerview:recyclerview:1.3.1"

implementation "androidx.constraintlayout:constraintlayout:2.1.4"

}

**2. Retrofit API Setup**

Create an interface for API calls.

**MovieApiService.kt**:

kotlin

CopyEdit

import retrofit2.Call

import retrofit2.Retrofit

import retrofit2.converter.gson.GsonConverterFactory

import retrofit2.http.GET

import retrofit2.http.Query

const val BASE\_URL = "https://api.themoviedb.org/3/"

interface MovieApiService {

@GET("movie/popular")

fun getPopularMovies(@Query("api\_key") apiKey: String): Call<MovieResponse>

companion object {

fun create(): MovieApiService {

val retrofit = Retrofit.Builder()

.baseUrl(BASE\_URL)

.addConverterFactory(GsonConverterFactory.create())

.build()

return retrofit.create(MovieApiService::class.java)

}

}

}

**3. Data Models**

Create data models to parse JSON.

**MovieResponse.kt**:

kotlin

CopyEdit

data class MovieResponse(

val results: List<Movie>

)

data class Movie(

val id: Int,

val title: String,

val overview: String,

val poster\_path: String

)

**4. Main Navigation Layout**

Set up a navigation graph for navigation between fragments.

**res/navigation/nav\_graph.xml**:

xml

CopyEdit

<?xml version="1.0" encoding="utf-8"?>

<navigation xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

app:startDestination="@id/movieListFragment">

<fragment

android:id="@+id/movieListFragment"

android:name="com.example.movieapp.MovieListFragment"

android:label="Movies" >

<action

android:id="@+id/action\_movieListFragment\_to\_movieDetailFragment"

app:destination="@id/movieDetailFragment" />

</fragment>

<fragment

android:id="@+id/movieDetailFragment"

android:name="com.example.movieapp.MovieDetailFragment"

android:label="Movie Details" />

</navigation>

**5. Fragments**

**MovieListFragment.kt**:

kotlin

CopyEdit

class MovieListFragment : Fragment() {

private lateinit var recyclerView: RecyclerView

private val movieApi = MovieApiService.create()

override fun onCreateView(

inflater: LayoutInflater, container: ViewGroup?,

savedInstanceState: Bundle?

): View? {

val view = inflater.inflate(R.layout.fragment\_movie\_list, container, false)

recyclerView = view.findViewById(R.id.recyclerView)

recyclerView.layoutManager = LinearLayoutManager(requireContext())

fetchMovies()

return view

}

private fun fetchMovies() {

movieApi.getPopularMovies("your\_api\_key").enqueue(object : Callback<MovieResponse> {

override fun onResponse(call: Call<MovieResponse>, response: Response<MovieResponse>) {

if (response.isSuccessful) {

val movies = response.body()?.results ?: emptyList()

recyclerView.adapter = MovieAdapter(movies) { selectedMovie ->

val action = MovieListFragmentDirections

.actionMovieListFragmentToMovieDetailFragment(selectedMovie)

findNavController().navigate(action)

}

}

}

override fun onFailure(call: Call<MovieResponse>, t: Throwable) {

Toast.makeText(requireContext(), "Failed to fetch movies", Toast.LENGTH\_SHORT).show()

}

})

}

}

**MovieDetailFragment.kt**:

kotlin

CopyEdit

class MovieDetailFragment : Fragment() {

override fun onCreateView(

inflater: LayoutInflater, container: ViewGroup?,

savedInstanceState: Bundle?

): View? {

val view = inflater.inflate(R.layout.fragment\_movie\_detail, container, false)

val movie = arguments?.getParcelable<Movie>("selectedMovie")

view.findViewById<TextView>(R.id.titleTextView).text = movie?.title

view.findViewById<TextView>(R.id.overviewTextView).text = movie?.overview

val imageView = view.findViewById<ImageView>(R.id.posterImageView)

Glide.with(this)

.load("https://image.tmdb.org/t/p/w500${movie?.poster\_path}")

.into(imageView)

return view

}

}

**6. Adapter**

**MovieAdapter.kt**:

kotlin

CopyEdit

class MovieAdapter(

private val movies: List<Movie>,

private val onClick: (Movie) -> Unit

) : RecyclerView.Adapter<MovieAdapter.MovieViewHolder>() {

inner class MovieViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {

private val title = itemView.findViewById<TextView>(R.id.movieTitleTextView)

fun bind(movie: Movie) {

title.text = movie.title

itemView.setOnClickListener { onClick(movie) }

}

}

override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MovieViewHolder {

val view = LayoutInflater.from(parent.context).inflate(R.layout.item\_movie, parent, false)

return MovieViewHolder(view)

}

override fun onBindViewHolder(holder: MovieViewHolder, position: Int) {

holder.bind(movies[position])

}

override fun getItemCount(): Int = movies.size

}

**7. Layouts**

* **fragment\_movie\_list.xml**:

xml

CopyEdit

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recyclerView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" />

* **fragment\_movie\_detail.xml**:

xml

CopyEdit

<LinearLayout

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<ImageView

android:id="@+id/posterImageView"

android:layout\_width="match\_parent"

android:layout\_height="300dp"

android:scaleType="centerCrop" />

<TextView

android:id="@+id/titleTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="20sp"

android:padding="8dp" />

<TextView

android:id="@+id/overviewTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:padding="8dp" />

</LinearLayout>

* **item\_movie.xml**:

xml

CopyEdit

<LinearLayout

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="8dp">

<TextView

android:id="@+id/movieTitleTextView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="16sp" />

</LinearLayout>

**8. Navigation Host**

**MainActivity.kt**:

kotlin

CopyEdit

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

}

}

**activity\_main.xml**:

xml

CopyEdit

<androidx.fragment.app.FragmentContainerView

android:id="@+id/nav\_host\_fragment"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:navGraph="@navigation/nav\_graph"

app:defaultNavHost="true" />

**Run the App**

1. Replace your\_api\_key with your TMDB API key.
2. The app will display a list of popular movies, and clicking on a movie will navigate to the detail screen.