**Case Study: News App with Fragments**

**Overview**

The app will display news articles fetched from a public news API. The user will be able to switch between different news categories (e.g., General, Technology, Sports) using a tabbed interface, where each tab will represent a category and will be managed by a Fragment.

**Features**

1. **Tabbed Layout**: Each tab will represent a different news category.
2. **News Listing**: Each tab will display a list of news articles related to its category.
3. **Article Details**: Clicking on a news article will open a detailed view.
4. **API Integration**: News articles will be fetched from a public news API.

**Technical Stack**

* **Programming Language**: Kotlin
* **UI Framework**: Fragments, ViewPager2, and TabLayout
* **Networking**: Retrofit for API calls
* **News API**: [News API](https://newsapi.org/)

**Steps to Develop the App**

**1. Set Up the Project**

* Create a new project in Android Studio.
* Add necessary dependencies for:
  + Retrofit – making api calls
  + Gson for JSON parsing – parsing the API results
  + ViewPager2 and TabLayout for the UI. – android UI components

In your build.gradle file, include:

gradle

Copy code

implementation 'com.google.android.material:material:1.6.0' – rich look and feel

implementation 'androidx.viewpager2:viewpager2:1.0.0' – adding ViewPager

implementation 'com.squareup.retrofit2:retrofit:2.9.0' – adding Refrofit making API Calls

implementation 'com.squareup.retrofit2:converter-gson:2.9.0' – parsing JSON results

implementation (“com.github.bumptech.glide:glide:4.16.0”) – image loading and caching

**2. Design the UI**

* Use **ViewPager2** for swiping between fragments and **TabLayout** to represent the news categories.
* **activity\_main.xml**:

xml

Copy code

<com.google.android.material.tabs.TabLayout

android:id="@+id/tabLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="?attr/colorPrimary" />

<androidx.viewpager2.widget.ViewPager2

android:id="@+id/viewPager"

android:layout\_width="match\_parent"

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

**3. Create News API Interface**

* Register for a free API key at [NewsAPI](https://newsapi.org/).
* Create the interface to interact with the News API.

kotlin

Copy code

interface NewsApiService {

@GET("v2/top-headlines")

suspend fun getTopHeadlines(

@Query("category") category: String,

@Query("apiKey") apiKey: String = "YOUR\_API\_KEY",

@Query("country") country: String = "us"

): NewsResponse

}

* Example response model:

kotlin

Copy code

data class NewsResponse(val articles: List<Article>)

data class Article(val title: String, val description: String, val urlToImage: String)

**4. Create Fragments for Each Category**

Each fragment will represent a news category (e.g., Technology, Sports).

* **NewsFragment.kt**:

kotlin

Copy code

class NewsFragment : Fragment() {

private lateinit var newsAdapter: NewsAdapter

private lateinit var category: String

override fun onCreateView(

inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?

): View? {

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

val recyclerView = view.findViewById<RecyclerView>(R.id.recyclerView)

// Initialize RecyclerView and Adapter

newsAdapter = NewsAdapter(listOf())

recyclerView.adapter = newsAdapter

recyclerView.layoutManager = LinearLayoutManager(context)

// Get category from arguments

category = arguments?.getString("category") ?: "general"

// Fetch news articles based on the category

fetchNews()

return view

}

private fun fetchNews() {

val retrofit = Retrofit.Builder()

.baseUrl("https://newsapi.org/")

.addConverterFactory(GsonConverterFactory.create())

.build()

val service = retrofit.create(NewsApiService::class.java)

CoroutineScope(Dispatchers.IO).launch {

val response = service.getTopHeadlines(category)

withContext(Dispatchers.Main) {

newsAdapter.updateArticles(response.articles)

}

}

}

companion object {

fun newInstance(category: String): NewsFragment {

val fragment = NewsFragment()

val args = Bundle()

args.putString("category", category)

fragment.arguments = args

return fragment

}

}

}

**5. Implement ViewPager and TabLayout in MainActivity**

* **MainActivity.kt**:

kotlin

Copy code

class MainActivity : AppCompatActivity() {

private lateinit var viewPager: ViewPager2

private lateinit var tabLayout: TabLayout

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

viewPager = findViewById(R.id.viewPager)

tabLayout = findViewById(R.id.tabLayout)

val categories = listOf("general", "technology", "sports")

// Set up ViewPager with Fragments

val pagerAdapter = NewsPagerAdapter(this, categories)

viewPager.adapter = pagerAdapter

// Connect TabLayout with ViewPager

TabLayoutMediator(tabLayout, viewPager) { tab, position ->

tab.text = categories[position].capitalize()

}.attach()

}

}

* **NewsPagerAdapter.kt**:

kotlin

Copy code

class NewsPagerAdapter(

activity: AppCompatActivity, private val categories: List<String>

) : FragmentStateAdapter(activity) {

override fun getItemCount(): Int = categories.size

override fun createFragment(position: Int): Fragment {

return NewsFragment.newInstance(categories[position])

}

}

**6. RecyclerView Adapter for News Articles**

* **NewsAdapter.kt**:

kotlin

Copy code

class NewsAdapter(private var articles: List<Article>) :

RecyclerView.Adapter<NewsAdapter.NewsViewHolder>() {

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

val title: TextView = itemView.findViewById(R.id.title)

val description: TextView = itemView.findViewById(R.id.description)

val imageView: ImageView = itemView.findViewById(R.id.image)

}

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

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

return NewsViewHolder(view)

}

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

val article = articles[position]

holder.title.text = article.title

holder.description.text = article.description

Glide.with(holder.itemView.context).load(article.urlToImage).into(holder.imageView)

}

override fun getItemCount(): Int = articles.size

fun updateArticles(newArticles: List<Article>) {

articles = newArticles

notifyDataSetChanged()

}

}

**7. Article Details Screen (Optional)**

* Add an OnClickListener to each item in the RecyclerView to open a new Activity that shows detailed information about the article, including a web view to load the full news page.

**8. Test the App**

Test the app to make sure:

* News articles load correctly.
* Switching between tabs updates the news categories.
* Clicking on a news article opens detailed information.

**Conclusion**

This News App demonstrates the use of **Fragments**, **ViewPager2**, and **TabLayout** in Android development. By integrating the News API, you provide users with real-time news articles based on different categories, making the app a great example of combining UI design with API data fetching.

Application Flow:-

1. AndroidManifest.xml -> MainActivity(Launcher)
2. Inside MainActivity.kt -> set the layout of activity\_main.xml
   * + - 1. TabLayout
         2. ViewPager

Pager SetUp

* Fetching TabLayout, ViewPager components
* Setting the ViewPager to Fragment
  + MainView -> ViewPager -> NewsFragment
  + //how many pages
  + val categories = *listOf*("general", "technology", "sports")  
    *// Set up ViewPager with Fragments*val pagerAdapter = NewsPagerAdapter(this, categories)  
    viewPager.*adapter* = pagerAdapter

Tab SetUp

*// Connect TabLayout with ViewPager*TabLayoutMediator(tabLayout, viewPager) **{** tab, position **->** tab.*text* = categories[position].*capitalize*()  
**}**.attach()

1. NewsPagerAdapter.kt
   1. Creating fragment based catagories size
2. NewsFragment.kt
   1. Fragment\_news.xml
      1. Recyclerview -> NewsAdapter -> item\_article.xml -> ViewHolder
   2. FetchNews
      1. RetroFit build the API, fetch response

BackEnd

Ui -> MainActivity, NewsFragment

Support -> NewsAdapter, NewsPagerAdpater

API Call -> APIService interface, Article data class, NewResponse Data class

FrontEnd

Activity\_main.xml, fragment\_news.xml, item\_article.xml