

ANDROID APPLICATION FOR LATEST HEADLINES

Submitted by:

Venkateswaran B

Sivasabari R

Venkatesan S

Description

The app's main feature is displaying a list of news articles, each with a title, image, and brief description. Users can scroll through the list of articles and tap on an article to view more details. The app uses the Jetpack Compose UI toolkit to build the UI and it uses the coil library to load images. The app fetches data from a remote server using Retrofit library and demonstrates how to use the Jetpack Compose UI toolkit for Android development.

Required Dependencies

Add a library or plugin dependency. The best way to add and manage build dependencies is to use version catalogs, the method new projects use by default.

Add A Library Or Plugin

Dependency Order

Dependency Information For

To create an Android application for fetching the latest headlines, you'll typically need several key dependencies. These may include:

Networking Library: A library like Retrofit or OkHttp for making HTTP requests to fetch news data from APIs.

JSON Parsing: A library such as Gson or Jackson for parsing JSON responses from the news API.

- ▶ **Dependency Injection:** Dagger 2 or Hilt for managing dependencies and improving code structure.
- ▶ **Coroutines:** Kotlin Coroutines for handling asynchronous tasks, especially when making network calls.
- ▶ **RecyclerView:** Android's RecyclerView for displaying the list of news articles efficiently.
- ▶ **ViewModel and LiveData:** Architecture components for managing UI-related data in a lifecycle-conscious way.
- ▶ **Permissions:** Internet permission in the AndroidManifest.xml to allow network access.
- ▶ **Gradle Dependencies:** Specific dependencies to be added in the build.gradle file, such as groovy
- ▶ **UI Components:** Material Design components for a modern UI experience..
- ▶ These dependencies will help you build a robust Android application that can fetch and display the latest news headlines effectively..

Program Code:

```
<com.google.android.material.tabs.TabLayout
    Android:id="@+id/include"
    Android:layout_width="match_parent"
    Android:layout_height="?attr/actionBarSize"
    Android:layout_below="@id/toolbar"
    Android:layout_marginTop="0dp"
    Android:backgroundTint="@color/black"
    App:tabIndicatorColor="#03A9F4"
    App:tabIndicatorHeight="3.5dp"
    App:tabMode="scrollable"
    App:tabSelectedTextColor="#03A9F4"
    App:tabTextColor="@color/white">
```

```
<com.google.android.material.tabs.TabItem
    Android:id="@+id/health"
    Android:layout_width="wrap_content"
    Android:layout_height="wrap_content"
    Android:text="Medical" />
</com.google.android.material.tabs.TabLayout>
<androidx.viewpager.widget.ViewPager
    Android:id="@+id/fragmentcontainer"
    Android:layout_width="match_parent"
    Android:layout_height="match_parent"
    Android:layout_below="@id/include"
    Android:layout_marginTop="0dp">
```

```
ViewPager viewPager = findViewById(R.id.fragmentcontainer);
tabLayout = findViewById(R.id.include);
pagerAdapter = new PagerAdapter(getSupportFragmentManager(),6);
viewPager.setAdapter(pagerAdapter);
tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
    @Override
    Public void onTabSelected(TabLayout.Tab tab)
{
    }
});
viewPager.addOnPageChangeListener(new
TabLayout.TabLayoutOnPageChangeListener(tabLayout));
```

Output:

