TabLayout Step by Implemtation to Design Tabs with Swipe Views

Step 1: Create a New Project with an Empty Activity

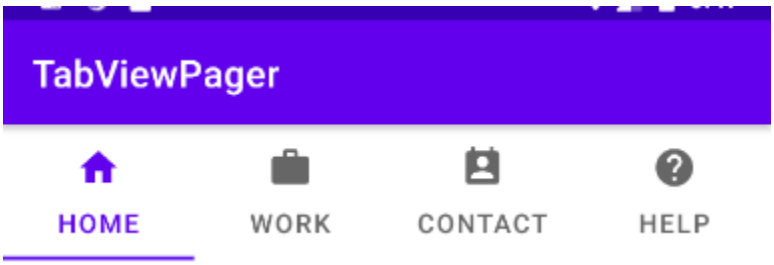
Add the following dependency on your build.gradle

```
// Get xml ids to Kotlin Code
apply plugin : "kotlin-android"
apply plugin : "kotlin-android-extensions"

// Material Design Library - also for ViewPager2
implementation 'com.google.android.material:material:1.5.0-alpha02'
```

Step (Optional)

The Default color of the Appbar will looks as mentioned below for the style

```
<style name="Theme.TabViewPager"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
```



If you want to customize the Appbar in your layout file, use the given code.

Go to your res\values\themes.xml to change the theme as mentioned below.

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
```

Go to your main-activity.xml. (The file has TabLayout and ViewPager2) add the below code to customize and choose whatever style you want

```
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
```

```
        app:popupTheme="@style/AppTheme.PopupOverlay" />
```

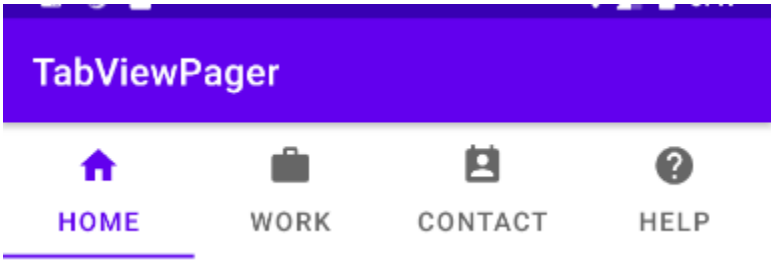
Step 3: Step Go to your activity_main.xml

Use the below code to add ViewPager2(to get the swipe view) and Tab Layout with their ids.

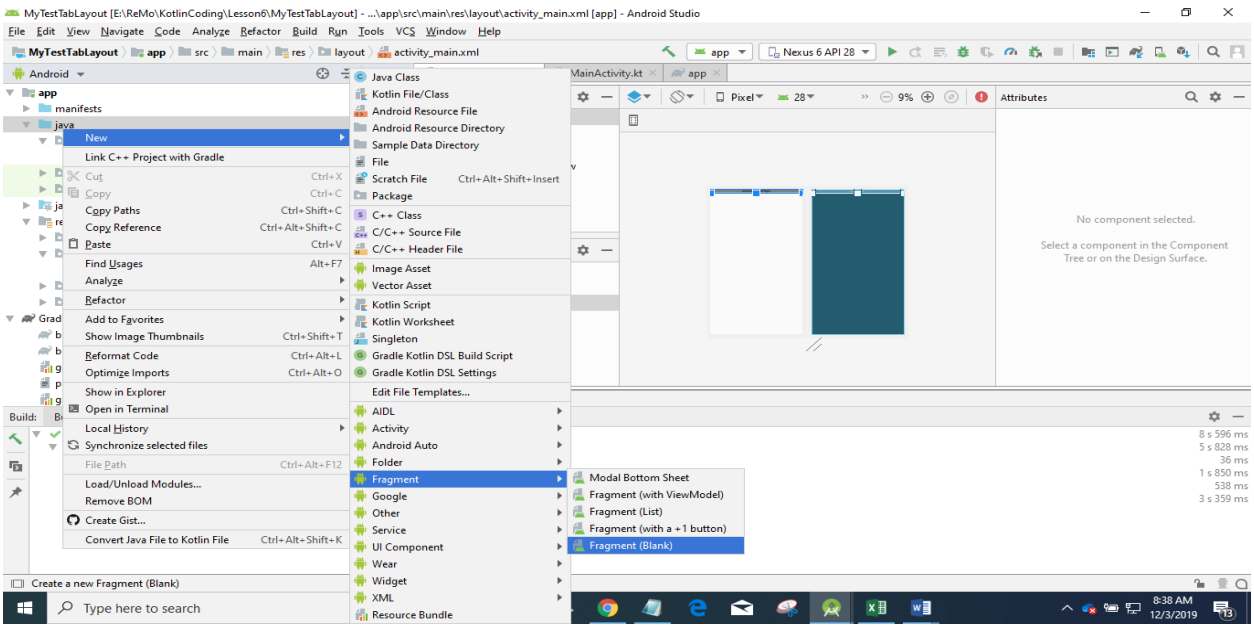
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical">
    <com.google.android.material.tabs.TabLayout
    android:id="@+id/tlayaout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    </com.google.android.material.tabs.TabLayout>

    <androidx.viewpager2.widget.ViewPager2
        android:id="@+id/viewpager"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </androidx.viewpager2.widget.ViewPager2>
</LinearLayout>
```

Step 4: Add Vector Asset in the Drawable folder.

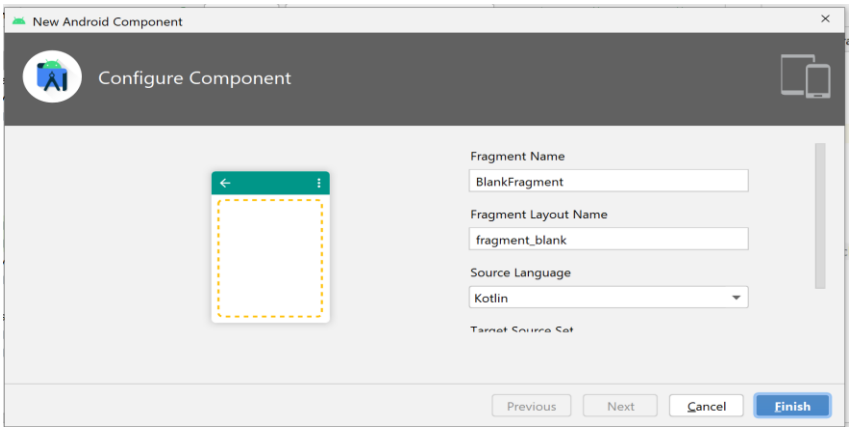
Problem Requirement: Design your code with 4 Fragments for the given Tabs. The drawable images are added from Vector Asset. To add pictures from the API library, **Right click drawable\New\Vector Asset and add four images as mentioned below on the Tabs.**



Step 5: Create a new Blank Fragment as per the screenshots.



In the next screen, Give the name of your Fragment as HomeFragment. **Click Finish**, Your Gradle Project take some time to Sync.



Fragment loaded with several lines of code. Only keep the below code and delete the remining codes.

```
class HomeFragment : Fragment() {
      override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_home, container, false)
    }
}
```

Step 6: Similar way creates another three fragments and named as

- WorkFragment
- ContactFragmnet
- HelpFragment

After completing the above steps, you will get four Fragements with its Layout.

The code HomeFragment and its Layout. Do the necessary changes in all Layout and its Fragment file. Similarly create for other three Fragments.

```
class HomeFragment : Fragment(R.layout.fragment_home)
```

```
<FrameLayout 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"
    tools:context=".HomeFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="My Home Page"
        android:textSize="18sp"
        android:gravity="center" />
</FrameLayout>
```

Step 7: Create your Adapter class

```
class MyViewAdapter(fm:FragmentManager,lc:Lifecycle) :
FragmentStateAdapter(fm,lc) {
    // How many Fragments - Returns the total number of items in the data
//set held by the adapter.
    override fun getItemCount(): Int = 4
    // Provide a new Fragment associated with the specified position.
    override fun createFragment(position: Int): Fragment {
        return   when(position){
            0->  HomeFragment()
            1->  WorkFragment()
            2-> ContactFragment()
            3->  HelpFragment()
            else-> Fragment()
        }
    }
}
```

Step 8: Do the implementation in MainActivity.xml

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        // Create an object for the Adapter Class
        val adapter = MyViewAdapter(supportFragmentManager,lifecycle)
        // Set the Adapter to your Viewpager UI
        viewpager.adapter = adapter
        /* Setting up Tab Layout with the ViewPageg2 is handled by the
TabLayoutMediator
        * by passing your tablayout id and viewpager id*/
        TabLayoutMediator(tlayaout,viewpager){tab,position->
            when(position){
                0->{
                    tab.text="Home"
                    tab.setIcon(R.drawable.home)
                }
                1->{
                    tab.text="Work"
                    tab.setIcon(R.drawable.work)
                }
                2->{
                    tab.text="Contact"
                    tab.setIcon(R.drawable.contact)
                }
                3->{
                    tab.text = "Help"
                    tab.setIcon(R.drawable.help)
                }
            }
        }.attach()
    }
}
```