

How to use Android ViewPager?

Android Mobile Development Apps/Applications

Before getting into the example, we should know what is view pager in android. View pager found in Support Library in Android, using view pager we can switch the fragments.

This example demonstrates how to use android view pager.

Step 1 – Create a new project in Android Studio, go to File ⇒ New Project and fill all required details to create a new project.

Step 2 – Add the following code in build.gradle.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.example.andy.myapplication"
        minSdkVersion 19
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
}
```

Step 3 – Add the following code to res/layout/activity_main.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|enterAlways"
            app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
        <android.support.design.widget.TabLayout
            android:id="@+id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:tabMode="fixed"
            app:tabGravity="fill"/>
    </android.support.design.widget.AppBarLayout>
    <android.support.v4.view.ViewPager
        android:id="@+id/viewpager"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior" />
</android.support.design.widget.CoordinatorLayout>
```

Step 4 – Add the following code to src/MainActivity.java

```
package com.example.andy.myapplication;
import android.annotation.TargetApi;
import android.os.Bundle;
import android.support.design.widget.TabLayout;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    private ViewPager viewPager;
    private Toolbar toolbar;
    private TabLayout tabLayout;
    @TargetApi(Build.VERSION_CODES.O)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        tabLayout = findViewById(R.id.tabs);
        viewPager = findViewById(R.id.viewpager);
        ViewPagerAdapter adapter = new ViewPagerAdapter(getSupportFragmentManager());
        adapter.addFragment(new OneFragment(), "ONE");
        adapter.addFragment(new TwoFragment(), "TWO");
        viewPager.setAdapter(adapter);
        tabLayout.setupWithViewPager(viewPager);
    }

    class ViewPagerAdapter extends FragmentPagerAdapter {
        private final List<Fragment> mList = new ArrayList<>();
        private final List<String> mTitleList = new ArrayList<>();
        public ViewPagerAdapter(FragmentManager supportFragmentManager) {
            super(supportFragmentManager);
        }
        @Override
        public Fragment getItem(int i) {
            return mList.get(i);
        }
        @Override
        public int getCount() {
            return mList.size();
        }
        public void addFragment(Fragment fragment, String title) {
            mList.add(fragment);
            mTitleList.add(title);
        }
        @Override
        public CharSequence getPageTitle(int position) {
            return mTitleList.get(position);
        }
    }
}
```

Step 5 – Add the following code to src/OneFragment.java

```
package com.example.andy.myapplication;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

public class OneFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.fragment_one, container, false);
        TextView textView = v.findViewById(R.id.text);
        textView.setText("First Fragment");
        return v;
    }
}
```

Step 6 – Add the following code to src/TwoFragment.java

```
package com.example.andy.myapplication;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;


public class TwoFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.fragment_one, container, false);
        TextView textView = v.findViewById(R.id.text);
        textView.setText("Second Fragment");
        return v;
    }
}
```

Step 7 – Add the following code to res/layout/fragment_one.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="20sp"
    />
</LinearLayout>
```

Step 8 – Add the following code to styles.xml.

```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="windowNoTitle">true</item>
        <item name="windowActionBar">false</item>
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```

Let's try to run your application. I assume you have connected your actual Android Mobile device with your computer. To run the app from android studio, open one of your project's activity files and click Run  icon from the toolbar. Select your mobile device as an option and then check your mobile device which will display your default screen –

