Android How to Build Intro Slider for your App

by Ravi Tamada / 257 Comments

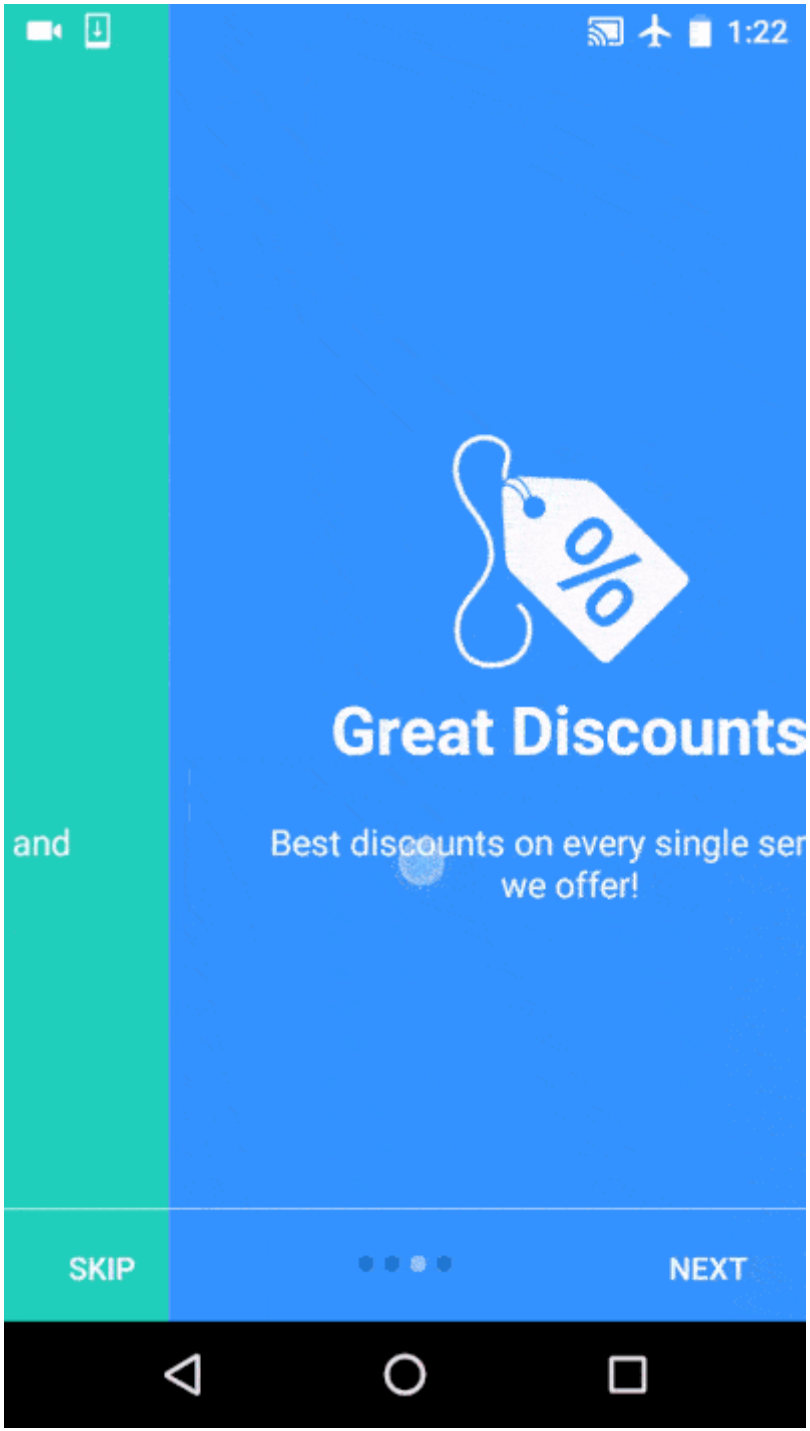
Adding Welcome / Intro screens in your app is a great way of showcasing the major features of the app. Previously I explained about adding a static Splash Screen to your app. In this article we are going to learn how to add an intro slider to your app where user can swipe through few slides before getting into app.

To demonstrate, I am creating a simple app that contains few intro slides with next and skip navigation. The user can navigate through each slide using swipe gesture or using the next button. I'll also explain how to show the intro only when the app is launched for the first time.

DOWNLOAD CODE DOWNLOAD .APK

VIDEO DEMO

Here is the final app we are going to build.



POPULAR ANDROID TUTORIALS

1. Android SQLite Database Tutorial - 1,634,211 views
2. How to connect Android with PHP, MySQL - 1,595,297 views
3. Android JSON Parsing Tutorial - 1,490,993 views
4. Android Push Notifications using Firebase Cloud Messaging FCM & PHP - 1,416,392 views
5. Android Sliding Menu using Navigation Drawer - 1,343,272 views
6. Android Login and Registration with PHP, MySQL and SQLite - 1,188,129 views
7. Android Custom ListView with Image and Text - 1,079,151 views
8. Android GPS, Location Manager Tutorial - 889,218 views
9. Android Tab Layout with Swipeable Views - 794,398 v

Let's start by creating a new project.

10. Android Expandable List View Tutorial - 718,928 views

1. Creating New Project

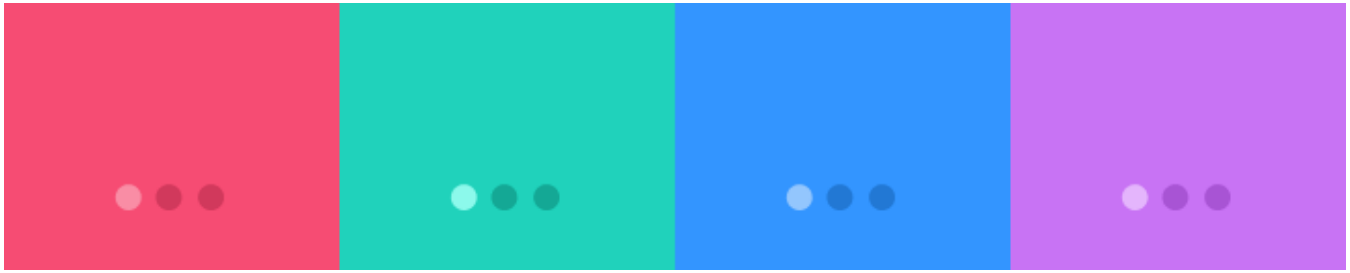
1. Create a new project in Android Studio from **File ⇒ New Project**. When it prompts you to select the default activity, select **Empty Activity** and proceed.

2. Download this res.zip and add them to your projects **res** folder. This zip file contains few drawable images required for this app.

2. Choosing the Colors

It's completely upto you how you design the intro screens considering the type of app you are building. For this example, I am placing a big image in centre and few texts below it. At the bottom, number of dots are placed indicating the number of slides it has.

Below is the color palette I have selected to design the screens. For every screen we need three colors i.e background color and two dot colors when it is active / inactive.



3. Open **colors.xml** located under **res ⇒ values** and add the below color values. You can see after adding the colors, I have kept them into arrays **array_dot_active** and **array_dot_inactive**, so that we can load them easily in our activity.

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>

    <!-- Screens background color-->
    <color name="bg_screen1">#f64c73</color>
    <color name="bg_screen2">#20d2bb</color>
    <color name="bg_screen3">#3395ff</color>
    <color name="bg_screen4">#c873f4</color>

    <!-- dots inactive colors -->
    <color name="dot_dark_screen1">#d1395c</color>
    <color name="dot_dark_screen2">#14a895</color>
    <color name="dot_dark_screen3">#2278d4</color>
    <color name="dot_dark_screen4">#a854d4</color>

    <!-- dots active colors -->
    <color name="dot_light_screen1">#f98da5</color>
    <color name="dot_light_screen2">#8cf9eb</color>
    <color name="dot_light_screen3">#93c6fd</color>
    <color name="dot_light_screen4">#e4b5fc</color>

    <array name="array_dot_active">
        <item>@color/dot_light_screen1</item>
        <item>@color/dot_light_screen2</item>
        <item>@color/dot_light_screen3</item>
        <item>@color/dot_light_screen4</item>
    </array>

    <array name="array_dot_inactive">
        <item>@color/dot_dark_screen1</item>
        <item>@color/dot_dark_screen2</item>
        <item>@color/dot_dark_screen3</item>
        <item>@color/dot_dark_screen4</item>
    </array>
</resources>
```

4. Open **strings.xml** located under **res ⇒ values** and add the below string values. Here I am mentioning a title and description for each slide.

strings.xml

```
<resources>
    <string name="app_name">Intro Slider</string>
    <string name="title_activity_welcome">Home Screen</string>
    <string name="next">NEXT</string>
    <string name="skip">SKIP</string>
    <string name="start">GOT IT</string>

    <string name="slide_1_title">Hello Food!</string>
    <string name="slide_1_desc">The easiest way to order food from your fav

    <string name="slide_2_title">Movie Tickets</string>
    <string name="slide_2_desc">Book movie tickets for your family and frie

    <string name="slide_3_title">Great Discounts</string>
    <string name="slide_3_desc">Best discounts on every single service we o

    <string name="slide_4_title">World Travel</string>
    <string name="slide_4_desc">Book tickets of any transportation and trav

    <string name="play_again_desc">To see the welcome slider again, goto Se
    <string name="play_again">Play Again</string>
</resources>
```

5. Open **dimens.xml** located under **res ⇒ values** and add the below values.

dimens.xml

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="fab_margin">16dp</dimen>
```

```
    <dimen name="dots_height">30dp</dimen>
    <dimen name="dots_margin_bottom">20dp</dimen>
    <dimen name="img_width_height">120dp</dimen>
    <dimen name="slide_title">30dp</dimen>
    <dimen name="slide_desc">16dp</dimen>
    <dimen name="desc_padding">40dp</dimen>
</resources>
```

6. Also make sure that you have these styles in your **styles.xml** as the app is crashing because of recent changes in build tools.

styles.xml

```
<resources>

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

    <style name="AppTheme.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dar

    <style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Ligh

</resources>
```

The welcome / intro slider should be shown only once when the app is launched for the very first time. If the user launches the app on second time, he should directly go to main screen. To achieve this, we use SharedPreferences to store a boolean value indicating first time launch.

7. Create a class named **PrefManager.java** and do the below changes. **isFirstTimeLaunch()** returns **true** if the app is launched for the first time.

PrefManager.java

```
package info.androidhive.introslider;

import android.content.Context;
import android.content.SharedPreferences;

/**
 * Created by Lincoln on 05/05/16.
 */
public class PrefManager {
    SharedPreferences pref;
    SharedPreferences.Editor editor;
    Context _context;

    // shared pref mode
    int PRIVATE_MODE = 0;

    // Shared preferences file name
    private static final String PREF_NAME = "androidhive-welcome";

    private static final String IS_FIRST_TIME_LAUNCH = "IsFirstTimeLaunch";

    public PrefManager(Context context) {
        this._context = context;
        pref = _context.getSharedPreferences(PREF_NAME, PRIVATE_MODE);
        editor = pref.edit();
    }

    public void setFirstTimeLaunch(boolean isFirstTime) {
        editor.putBoolean(IS_FIRST_TIME_LAUNCH, isFirstTime);
        editor.commit();
    }

    public boolean isFirstTimeLaunch() {
        return pref.getBoolean(IS_FIRST_TIME_LAUNCH, true);
    }
```

```
}
```

3. Creating the Welcome Slides

Now it's time to create the layouts required for the slider. In total I am keeping 4 slides for the intro screen. So we need four separate layouts for four slide. The layout of each slide remains the same except the images, text and colors. Alternatively you can create separate Fragment for each slide to have more control over the UI elements displayed in the slide.

8. So quickly create four xml layouts named **welcome_side1.xml**, **welcome_side2.xml**, **welcome_side3.xml** and **welcome_side4.xml** under **res** ⇒ **layouts**.

welcome_slide1.xml

welcome_slide1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/bg_screen1">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:gravity="center_horizontal"
        android:orientation="vertical">

        <ImageView
            android:layout_width="@dimen/img_width_height"
            android:layout_height="@dimen/img_width_height"
            android:src="@drawable/ic_food" />
```

```
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/slide_1_title"
            android:textColor="@android:color/white"
            android:textSize="@dimen/slide_title"
            android:textStyle="bold" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:paddingLeft="@dimen/desc_padding"
            android:paddingRight="@dimen/desc_padding"
            android:text="@string/slide_1_desc"
            android:textAlignment="center"
            android:textColor="@android:color/white"
            android:textSize="@dimen/slide_desc" />

    </LinearLayout>
</RelativeLayout>
```

welcome_slide2.xml

```
welcome_slide2.xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/bg_screen2">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:gravity="center_horizontal"
        android:orientation="vertical">
```

```
        <ImageView
            android:layout_width="@dimen/img_width_height"
            android:layout_height="@dimen/img_width_height"
            android:src="@drawable/ic_movie" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/slide_2_title"
            android:textColor="@android:color/white"
            android:textSize="@dimen/slide_title"
            android:textStyle="bold" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:paddingLeft="@dimen/desc_padding"
            android:paddingRight="@dimen/desc_padding"
            android:text="@string/slide_2_desc"
            android:textAlignment="center"
            android:textColor="@android:color/white"
            android:textSize="@dimen/slide_desc" />

    </LinearLayout>


</RelativeLayout>
```

welcome_slide3.xml

welcome_slide3.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/bg_screen3">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:gravity="center_horizontal"
        android:orientation="vertical">

        <ImageView
            android:layout_width="@dimen/img_width_height"
            android:layout_height="@dimen/img_width_height"
            android:src="@drawable/ic_discount" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/slide_3_title"
            android:textColor="@android:color/white"
            android:textSize="@dimen/slide_title"
```

```
            android:textStyle="bold" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:paddingLeft="@dimen/desc_padding"
            android:paddingRight="@dimen/desc_padding"
            android:text="@string/slide_3_desc"
            android:textAlignment="center"
            android:textColor="@android:color/white"
            android:textSize="@dimen/slide_desc" />

    </LinearLayout>

</RelativeLayout>
```

welcome_slide4.xml

welcome_slide4.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/bg_screen4">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:gravity="center_horizontal"
        android:orientation="vertical">

        <ImageView
            android:layout_width="@dimen/img_width_height"
            android:layout_height="@dimen/img_width_height"
            android:src="@drawable/ic_travel" />
```

```
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/slide_4_title"
            android:textColor="@android:color/white"
            android:textSize="@dimen/slide_title"
            android:textStyle="bold" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:paddingLeft="@dimen/desc_padding"
            android:paddingRight="@dimen/desc_padding"
            android:text="@string/slide_4_desc"
            android:textAlignment="center"
            android:textColor="@android:color/white"
            android:textSize="@dimen/slide_desc" />

    </LinearLayout>

</RelativeLayout>
```

9. Once the layouts are ready, create a new activity named **WelcomeActivity.java** for the welcome slider. **Right Click** on package **New** ⇒ **Activity** ⇒ **Empty Activity**.

10. Open **activity_welcome.xml** and modify the code as below. Here I am adding **ViewPager** for slider, **LinearLayout** for **bottom dots** and two buttons for **Skip** and **Next** navigation.

activity_welcome.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:showIn="@layout/activity_welcome">


    <android.support.v4.view.ViewPager
        android:id="@+id/view_pager"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <LinearLayout
        android:id="@+id/layoutDots"
        android:layout_width="match_parent"
        android:layout_height="@dimen/dots_height"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="@dimen/dots_margin_bottom"
        android:gravity="center"
        android:orientation="horizontal"></LinearLayout>

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:alpha=".5"
        android:layout_above="@id/layoutDots"
        android:background="@android:color/white" />

    <Button
        android:id="@+id/btn_next"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:background="@null"
        android:text="@string/next"
        android:textColor="@android:color/white" />

    <Button
        android:id="@+id/btn_skip"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:background="@null"
        android:text="@string/skip"
```

```
        android:textColor="@android:color/white" />

</RelativeLayout>
```

11. Open **WelcomeActivity.java** and modify the code as below. Here I have taken care of few things

> Check for the fist time app launch using **prefManager.isFirstTimeLaunch()** method. If it returns true, **MainActivity** will be launched skipping the intro activity.

> Made the notification bar transparent, so that the view background color can be seen through.

> Created a **PagerAdapter** for the **ViewPager** and inflated all the layouts we created earlier.

> Added **click event** listener to **Skip** and **Next** buttons. If next button is clicked, next slide will be shown. If Skip button is clicked, main activity will be launched directly.

```
WelcomeActivity.java
package info.androidhive.introslider;

import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.os.Build;
import android.os.Bundle;
import android.support.v4.view.PagerAdapter;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.text.Html;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
```

```
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;

public class WelcomeActivity extends AppCompatActivity {

    private ViewPager viewPager;
    private MyViewPagerAdapter myViewPagerAdapter;
    private LinearLayout dotsLayout;
    private TextView[] dots;
    private int[] layouts;
    private Button btnSkip, btnNext;
    private PrefManager prefManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Checking for first time launch - before calling setContentView()
        prefManager = new PrefManager(this);
        if (!prefManager.isFirstTimeLaunch()) {
            launchHomeScreen();
            finish();
        }

        // Making notification bar transparent
        if (Build.VERSION.SDK_INT >= 21) {
            getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI
        }

        setContentView(R.layout.activity_welcome);

        viewPager = (ViewPager) findViewById(R.id.view_pager);
        dotsLayout = (LinearLayout) findViewById(R.id.layoutDots);
        btnSkip = (Button) findViewById(R.id.btn_skip);
        btnNext = (Button) findViewById(R.id.btn_next);


        // layouts of all welcome sliders
        // add few more layouts if you want
        layouts = new int[]{
```

```
                R.layout.welcome_slide1,
                R.layout.welcome_slide2,
                R.layout.welcome_slide3,
                R.layout.welcome_slide4};

        // adding bottom dots
        addBottomDots(0);

        // making notification bar transparent
        changeStatusBarColor();

        myViewPagerAdapter = new MyViewPagerAdapter();
        viewPager.setAdapter(myViewPagerAdapter);
        viewPager.addOnPageChangeListener(viewPagerPageChangeListener);

        btnSkip.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                launchHomeScreen();
            }
        });

        btnNext.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // checking for last page
                // if last page home screen will be launched
                int current = getItem(+1);
                if (current < layouts.length) {
                    // move to next screen
                    viewPager.setCurrentItem(current);
                } else {
                    launchHomeScreen();
                }
            }
        });
    }

    private void addBottomDots(int currentPage) {
        dots = new TextView[layouts.length];

        int[] colorsActive = getResources().getIntArray(R.array.array_dot_a
        int[] colorsInactive = getResources().getIntArray(R.array.array_dot
```

```
        dotsLayout.removeAllViews();
        for (int i = 0; i < dots.length; i++) {
            dots[i] = new TextView(this);
            dots[i].setText(Html.fromHtml("&#8226;"));
            dots[i].setTextSize(35);
            dots[i].setTextColor(colorsInactive[currentPage]);
            dotsLayout.addView(dots[i]);
        }

        if (dots.length > 0)
            dots[currentPage].setTextColor(colorsActive[currentPage]);
    }

    private int getItem(int i) {
        return viewPager.getCurrentItem() + i;
    }

    private void launchHomeScreen() {
        prefManager.setFirstTimeLaunch(false);
        startActivity(new Intent(WelcomeActivity.this, MainActivity.class))
        finish();
    }

    //  viewpager change listener
    ViewPager.OnPageChangeListener viewPagerPageChangeListener = new ViewPa

        @Override
        public void onPageSelected(int position) {
            addBottomDots(position);

            // changing the next button text 'NEXT' / 'GOT IT'
            if (position == layouts.length - 1) {
                // last page. make button text to GOT IT
                btnNext.setText(getString(R.string.start));
                btnSkip.setVisibility(View.GONE);
            } else {
                // still pages are left
                btnNext.setText(getString(R.string.next));
                btnSkip.setVisibility(View.VISIBLE);
            }
        }
```

```
        @Override
        public void onPageScrolled(int arg0, float arg1, int arg2) {

        }

        @Override
        public void onPageScrollStateChanged(int arg0) {

        }
    };

    /**
     * Making notification bar transparent
     */
    private void changeStatusBarColor() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            Window window = getWindow();
            window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BA
            window.setStatusBarColor(Color.TRANSPARENT);
        }
    }

    /**
     * View pager adapter
     */
    public class MyViewPagerAdapter extends PagerAdapter {
        private LayoutInflater layoutInflater;

        public MyViewPagerAdapter() {
        }

        @Override
        public Object instantiateItem(ViewGroup container, int position) {
            layoutInflater = (LayoutInflater) getSystemService(Context.LAYO

            View view = layoutInflater.inflate(layouts[position], container
            container.addView(view);

            return view;
        }

        @Override
        public int getCount() {
```

```
            return layouts.length;
        }

        @Override
        public boolean isViewFromObject(View view, Object obj) {
            return view == obj;
        }


        @Override
        public void destroyItem(ViewGroup container, int position, Object o
            View view = (View) object;
            container.removeView(view);
        }
    }
}
```

12. Finally open **AndroidManifest.xml** and make **WelcomeActivity** as launcher activity. So that it will be shown as the first screen in the app.

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="info.androidhive.introslider">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name="info.androidhive.introslider.WelcomeActivit
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" /
            </intent-filter>
```

```
        </activity>
        <activity
            android:name="info.androidhive.introslider.MainActivity"
            android:label="@string/title_activity_welcome"
            android:theme="@style/AppTheme.NoActionBar"></activity>
    </application>

</manifest>
```

Run the app and play with awesomely looking intro sliders 🙂

Ravi Tamada

Ravi is hardcore Android programmer and Android programming has been his passion since he compiled his first hello-world program. Solving real problems of Android developers through tutorials has always been interesting part for him.

RELATED POSTS