



Downloads

Tips

HOME &gt; MATERIAL DESIGN &gt;

Android Material Design working with Tabs

# Android Material Design working with Tabs

by Ravi Tamada / September 13, 2015 / 294 Comments

Android Design Support Library made our day easier by providing backward compatibility to number of material design components all the way back to Android 2.1. In Design support Library the components like navigation drawer, floating action button, snackbar, tabs, floating labels and animation frameworks were introduced. In this article we are going to learn how to implement material tabs in your apps.

Before going further, I suggest have a look at this tabs docs that defines do's and don'ts while implementing tabs.

[DOWNLOAD CODE](#)
[VIDEO DEMO](#)
[Subscribe to Newsletter](#)

Join our 746,498 subscribers and get access to the latest android tutorials,

SEARCH HERE

Search the site

WE'RE SOCIAL


[Like Page](#)

1 friend likes this



200 ★★★★★  
FIVE STAR  
REVIEWS

FOR YOUR ANDROID APP

[BUY NOW FOR \\$ 299](#)

## Android Material Design Tabs Tutorial (Demo)



freebies, scripts and  
much more!

Email

Sign Up



We strictly care about  
your privacy!

### POPULAR ANDROID TUTORIALS

## 1. Making the App Material

We'll start this by creating a new project and applying the material theme. If you are not aware of android material design, my previous article [Android Getting Started with Material Design](#) gives you a good start.

**1.** In Android Studio, go to **File ⇒ New Project** and fill all the details required to create a new project. When it prompts to select a default activity, select **Blank Activity** and proceed.

**2.** Open **build.gradle** and add android design support library **com.android.support:design:23.0.1**

### build.gradle

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:23.0.1'
```

```
        compile 'com.android.support:design:23.0.1'
    }
```

3. Open **colors.xml** located under **res ⇒ values** and add the below color values.

```
colors.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#125688</color>
    <color name="colorPrimaryDark">#125688</color>
    <color name="textColorPrimary">#FFFFFF</color>
    <color name="windowBackground">#FFFFFF</color>
    <color name="navigationBarColor">#000000</color>
    <color name="colorAccent">#c8e8ff</color>
</resources>
```

4. Add the below dimensions to **dimens.xml** located under **res ⇒ 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="tab_max_width">264dp</dimen>
    <dimen name="tab_padding_bottom">16dp</dimen>
    <dimen name="tab_label">14sp</dimen>
    <dimen name="custom_tab_layout_height">72dp</dimen>
</resources>
```

5. Open **styles.xml** located under **res ⇒ values** and add below styles. The styles defined in this styles.xml are common to all the android versions.

Menu using Navigation Drawer - 1,026,976 views

6. Android Custom ListView with Image and Text - 947,077 views

7. Android Login and Registration with PHP, MySQL and SQLite - 897,925 views

8. Android GPS, Location Manager Tutorial - 684,565 views

9. Android Tab Layout with Swipeable Views - 653,518 views

10. Android Tab Layout Tutorial - 579,535 views

```
styles.xml
```

```
<resources>
```

```
    <style name="MyMaterialTheme" parent="MyMaterialThemeBase">
        <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>
```

6. Now under **res**, create a folder named **values-v21**. Inside **values-v21**, create another **styles.xml** with the below styles. These styles are specific to **Android 5.0**

```
styles.xml
```

```
<resources>
```

```
    <style name="MyMaterialTheme" parent="MyMaterialThemeBase">
        <item name="android:windowContentTransitions">true</item>
        <item name="android:windowAllowEnterTransitionFirstTimeOnly">false</item>
        <item name="android:windowAllowReturnTransitionFirstTimeOnly">false</item>
        <item name="android:windowSharedElementEnterTransition">@transition/moving</item>
        <item name="android:windowSharedElementExitTransition">@transition/moving</item>
    </style>
```

```
</resources>
```

7. Finally open **AndroidManifest.xml** and modify the theme to our customized theme by changing the **android:theme** attribute value.

```
        android:theme="@style/MyMaterialTheme"
```

### AndroidManifest.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/MyMaterialTheme" >
        <activity
            android:name=".activity.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Run the app and verify the material theme by observing the notification bar color. If you see the notification bar color changed, it means that the material design theme is applied successfully.



www.androidhive.info

Now we have our app material ready. So let's start adding the tabs. But before that we'll create few fragment activities for testing purpose. All these fragment activities contains very simple UI with only one TextView.

8. Under your main package create a fragment named **OneFragment.java** and add the below code.

```
OneFragment.java
package info.androidhive.materialtabs.fragments;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import info.androidhive.materialtabs.R;

public class OneFragment extends Fragment{

    public OneFragment() {
        // Required empty public constructor
    }

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

    @Override
    public View onCreateView(LayoutInflater inflater,
                           Bundle savedInstanceState)
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_one,
    }

}
```

9. Open **fragment\_one.xml** located under **res ⇒ layout** and do the below changes.

```
fragment_one.xml
<RelativeLayout 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="info.androidhive.materialtabs.fragments.OneFragment" >
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/one"  
    android:textSize="40dp"  
    android:textStyle="bold"  
    android:layout_centerInParent="true"/>  
  
</RelativeLayout>
```

**10.** Likewise create few more fragment activities with same code we used for OneFragment.java. I have created **TwoFragment.java**, **ThreeFragment.java**, **FourFragemnt.java** ..... upto **TenFragment.java**

## 2. Fixed Tabs

Fixed tabs should be used when you have limited number of tabs. These tabs are fixed in position. In android design support library lot of new elements like CoordinatorLayout, AppBarLayout, TabLayout and lot more were introduced. I won't cover all of these as it's not the agenda of this article.

**11.** Open the layout file of main activity (**activity\_main.xml**) and add below layout code.

**app:tabMode** – Defines the mode of the tab layout. In our case the value should be “**fixed**”

```
activity_main.xml  
<android.support.design.widget.CoordinatorLayout xmlns:  
    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.
```

```

<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>

```

## 12. Open **MainActivity.java** and do the below changes.

**tabLayout.setupWithViewPager()** – Assigns the ViewPager to TabLayout.

**setupViewPager()** – Defines the number of tabs by setting appropriate fragment and tab name.

**ViewPagerAdapter** – Custom adapter class provides fragments required for the view pager.

### MainActivity.java

```

package info.androidhive.materialtabs.activity;

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;

import info.androidhive.materialtabs.R;
import info.androidhive.materialtabs.fragments.OneFragment;
import info.androidhive.materialtabs.fragments.ThreeFragment;
import info.androidhive.materialtabs.fragments.TwoFragment;

public class MainActivity extends AppCompatActivity {

    private Toolbar toolbar;
    private TabLayout tabLayout;
    private ViewPager viewPager;

    @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);

        viewPager = (ViewPager) findViewById(R.id.viewpager);
        setupViewPager(viewPager);

        tabLayout = (TabLayout) findViewById(R.id.tabs);
        tabLayout.setupWithViewPager(viewPager);
    }

    private void setupViewPager(ViewPager viewPager) {
        PagerAdapter adapter = new PagerAdapter(getSupportFragmentManager());
        adapter.addFragment(new OneFragment(), "ONE");
        adapter.addFragment(new TwoFragment(), "TWO");
        adapter.addFragment(new ThreeFragment(), "THREE");
        viewPager.setAdapter(adapter);
    }

    class PagerAdapter extends FragmentPagerAdapter {
        private final List<Fragment> mFragmentList =
            new ArrayList<Fragment>();
        private final List<String> mFragmentTitleList =
            new ArrayList<String>();

        public PagerAdapter(FragmentManager manager) {
            super(manager);
        }

        @Override
        public Fragment getItem(int position) {
            return mFragmentList.get(position);
        }

        @Override
        public int getCount() {
            return mFragmentList.size();
        }

        @Override
        public CharSequence getPageTitle(int position) {
            return mFragmentTitleList.get(position);
        }
    }
}
```

```
@Override  
public int getCount() {  
    return mFragmentList.size();  
}  
  
public void addFragment(Fragment fragment, St  
    mFragmentList.add(fragment);  
    mFragmentTitleList.add(title);  
}  
  
@Override  
public CharSequence getPageTitle(int position)  
    return mFragmentTitleList.get(position);  
}  
}  
}
```

Now run the app. You should able to see the tabs displayed with swipe functionality between the tabs.



## 2.1 Full Width Tabs

If you want the tabs to be occupied the fullwidth of the screen, you need to assign `app:tabGravity="fill"` to our TabLayout.

### Android Tab Layout Landscape View



www.androidhive.info

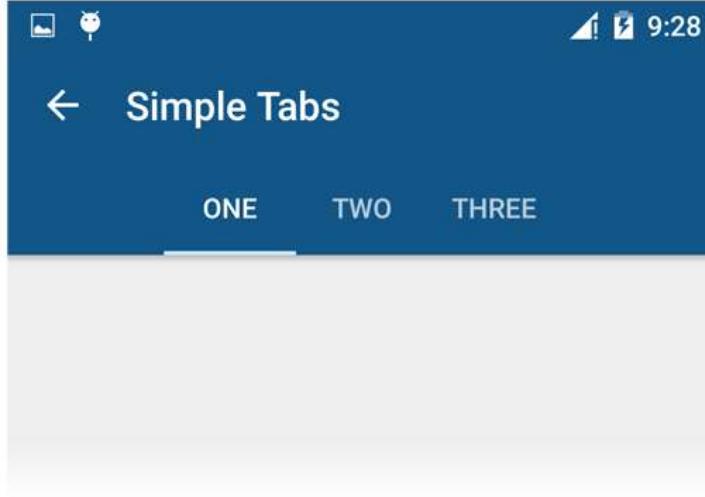
## 2.2 Center Aligned Tabs

If you want to keep your tabs horizontally centered, assign

`app:tabGravity="center"`

to TabLayout.

### Android Tab Layout - Gravity Center



www.androidhive.info

## 3. Scrollable Tabs

The scrollable tabs should be used when you have many number of

tabs where there is insufficient space on the screen to fit all of them. To make the tabs scrollable, set

`app:tabMode="scrollable"` to TabLayout.

13. Open **activity\_main.xml** and change the `app:tabMode` to **scrollable**.

```
<android.support.design.widget.TabLayout  
    android:id="@+id/tabs"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:tabMode="scrollable"/>
```

14. Edit **MainActivity.java** and add few fragments to ViewPager in **setupViewPager()** method. I have added total of 10 fragments to ViewPager. After the changes, your main activity should look like below.

```
MainActivity.java  
package info.androidhive.materialtabs.activity;  
  
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 Toolbar toolbar;  
    private TabLayout tabLayout;  
    private ViewPager viewPager;  
  
    @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)

viewPager = (ViewPager) findViewById(R.id.viewpager)
setupViewPager(viewPager);

tabLayout = (TabLayout) findViewById(R.id.tabs)
tabLayout.setupWithViewPager(viewPager);
}

private void setupViewPager(ViewPager viewPager)
    ViewPagerAdapter adapter = new ViewPagerAdapter();
    adapter.addFrag(new OneFragment(), "ONE");
    adapter.addFrag(new TwoFragment(), "TWO");
    adapter.addFrag(new ThreeFragment(), "THREE");
    adapter.addFrag(new FourFragment(), "FOUR");
    adapter.addFrag(new FiveFragment(), "FIVE");
    adapter.addFrag(new SixFragment(), "SIX");
    adapter.addFrag(new SevenFragment(), "SEVEN");
    adapter.addFrag(new EightFragment(), "EIGHT");
    adapter.addFrag(new NineFragment(), "NINE");
    adapter.addFrag(new TenFragment(), "TEN");
    viewPager.setAdapter(adapter);
}

class ViewPagerAdapter extends FragmentPagerAdapter
    private final List<Fragment> mFragmentList =
    private final List<String> mFragmentTitleList

    public ViewPagerAdapter(FragmentManager manager) {
        super(manager);
    }

    @Override
    public Fragment getItem(int position) {
        return mFragmentList.get(position);
    }

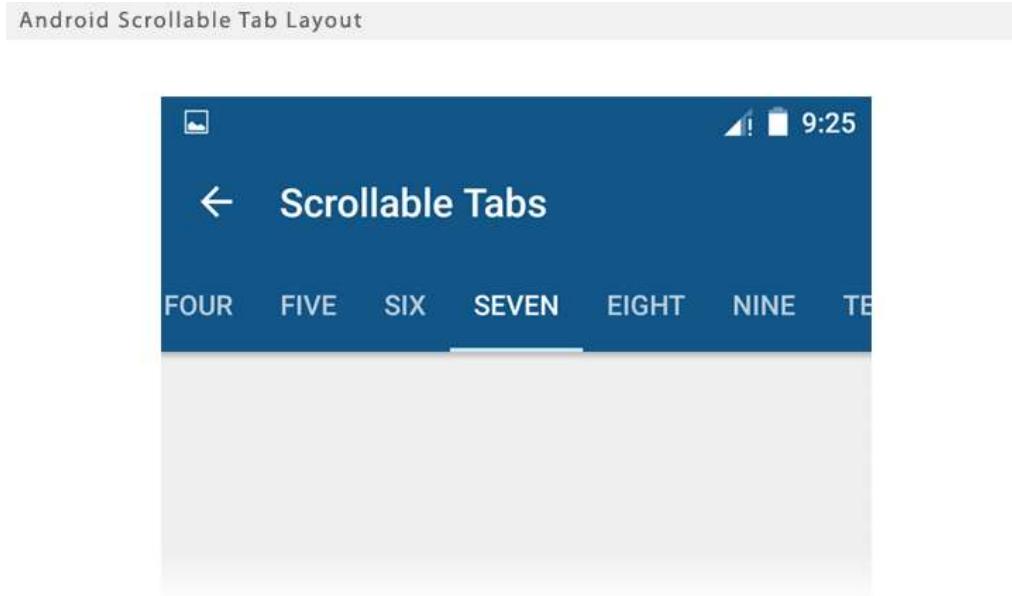
    @Override
    public int getCount() {
        return mFragmentList.size();
    }

    public void addFrag(Fragment fragment, String title) {
        mFragmentList.add(fragment);
        mFragmentTitleList.add(title);
    }

    @Override
    public CharSequence getPageTitle(int position) {
        return mFragmentTitleList.get(position);
    }
}
```



Now if you run the app, you can see the more number of tabs with scrollable functionality.



## 4. Tabs with Icon & Text

Sometimes you might wanted to add an icon to Tab. Earlier adding an icon to tab is tedious process. But with the design support library it is very easy. All you have to do is call **setIcon()** method by passing appropriate icon. The icon will be placed in front of tab label.

```
tabLayout.getTabAt(0).setIcon(tabIcons[0]);
tabLayout.getTabAt(1).setIcon(tabIcons[1]);
```

**15.** Open your **MainActivity.java** and modify the code as below. Here I have added a new method called **setupTabIcons()** in which I have set all the tab icons.

#### MainActivity.java

```
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 Toolbar toolbar;
    private TabLayout tabLayout;
    private ViewPager viewPager;
    private int[] tabIcons = {
        R.drawable.ic_tab_favourite,
        R.drawable.ic_tab_call,
        R.drawable.ic_tab_contacts
    };

    @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);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();

        if (id == R.id.action_settings) {
            Intent intent = new Intent(this, SettingsActivity.class);
            startActivity(intent);
        }

        return super.onOptionsItemSelected(item);
    }

    private void setupTabIcons() {
        tabLayout.getTabAt(0).setIcon(tabIcons[0]);
        tabLayout.getTabAt(1).setIcon(tabIcons[1]);
        tabLayout.getTabAt(2).setIcon(tabIcons[2]);
    }
}
```

```
viewPager = (ViewPager) findViewById(R.id.vie
setupViewPager(viewPager);

tabLayout = (TabLayout) findViewById(R.id.tabl
tabLayout.setupWithViewPager(viewPager);
setupTabIcons();
}

private void setupTabIcons() {
    tabLayout.getTabAt(0).setIcon(tabIcons[0]);
    tabLayout.getTabAt(1).setIcon(tabIcons[1]);
    tabLayout.getTabAt(2).setIcon(tabIcons[2]);
}

private void setupViewPager(ViewPager viewPager)
    ViewPagerAdapter adapter = new ViewPagerAdapt
        adapter.addFrag(new OneFragment(), "ONE");
        adapter.addFrag(new TwoFragment(), "TWO");
        adapter.addFrag(new ThreeFragment(), "THREE")
    viewPager.setAdapter(adapter);
}

class ViewPagerAdapter extends FragmentPagerAdapter
    private final List<Fragment> mFragmentList =
    private final List<String> mFragmentTitleList

    public ViewPagerAdapter(FragmentManager manager)
        super(manager);
    }

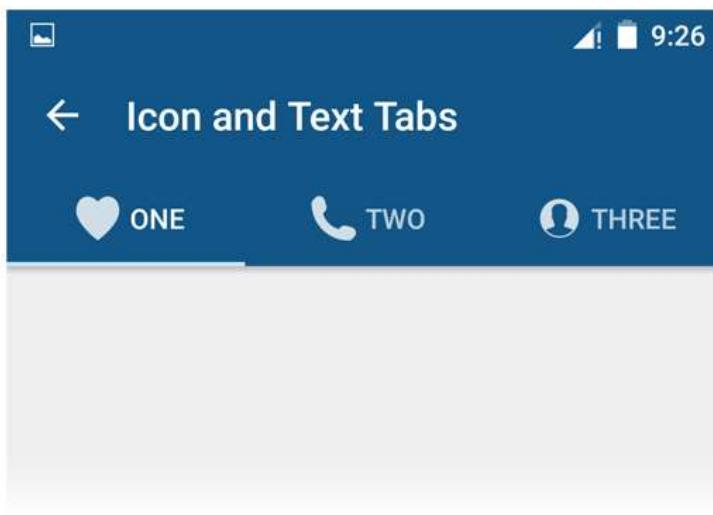
    @Override
    public Fragment getItem(int position) {
        return mFragmentList.get(position);
    }

    @Override
    public int getCount() {
        return mFragmentList.size();
    }

    public void addFrag(Fragment fragment, String title) {
        mFragmentList.add(fragment);
        mFragmentTitleList.add(title);
    }

    @Override
    public CharSequence getPageTitle(int position) {
        return mFragmentTitleList.get(position);
    }
}
```

## Android Tab Layout with Icon and Text



www.androidhive.info

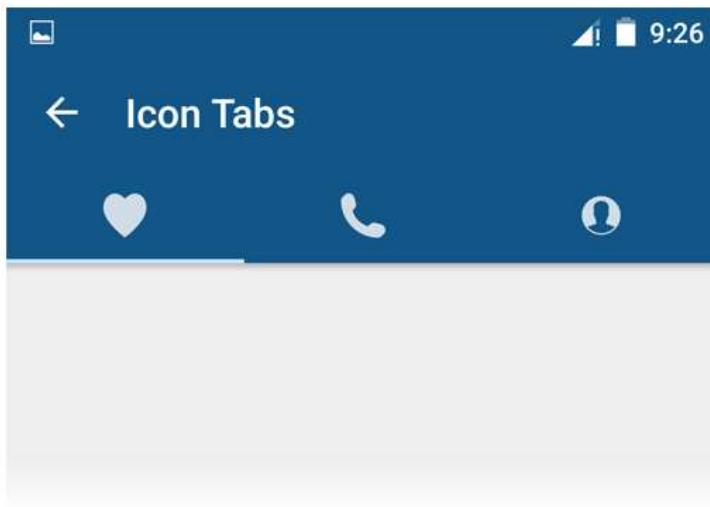
## 5. Tabs with only Icons

Setting only icon to tab is same as setting text and icon except the method `getPageTitle()` in ViewPagerAdapter class returns `null` instead of tab label.

**16.** Open **MainActivity.java** and modify the `getPageTitle()` method as below and run the project.

```
@Override  
public CharSequence getPageTitle(int position) {  
  
    // return null to display only the icon  
    return null;  
}
```

## Android Tab Layout with only Icons



www.androidhive.info

## 6. Custom Tab View with Icon & Text

Setting a custom view to the tab is very useful when you are not able to achieve desired output by following the methods provided by tab layout. While setting a custom view to tab, make sure that you follow the [specs](#) suggested by android for tabs.

When we set the tab an icon and text, you can see the icon is horizontally aligned with tab text. But if you want to place the icon above the tab label, you have to use a custom view to achieve it.

17. Under **res ⇒ values**, create an xml file named **fonts.xml** and add below string value. This xml file defines the font family for the tab label.

```
fonts.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="font_fontFamily_medium">sans-serif</string>
</resources>
```

**18. Under `res ⇒ values-v21`, create another xml named `fonts.xml`.**

```
fonts.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="font_fontFamily_medium">sans-serif-
</resources>
```

**19. Open `activity_main.xml` and set the custom height to TabLayout. Setting this height is important as placing icon above the tab label takes more space than normal.**

```
<android.support.design.widget.TabLayout
    android:id="@+id/tabs"
    android:layout_width="match_parent"
    android:layout_height="@dimen/custom_tab_]
    app:tabMode="fixed"
    app:tabGravity="fill"/>
```

**20. Create an xml layout named `custom_tab.xml` under `res ⇒ layout`. In this layout we have defined the custom view for the tab.**

```
custom_tab.xml
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/a
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/tab"
    android:textColor="@color/colorAccent"
    android:textSize="@dimen/tab_label"
    android:fontFamily="@string/font_fontFamily_medium"
```

**21. Open `MainActivity.java` and modify the code as below. Here if you observe `setupTabIcons()` method, I have rendered `custom_tab.xml` layout in each tab using below lines of code.**

```
TextView tabOne = (TextView) LayoutInflater.from(this)
tabOne.setText("ONE");
tabOne.setCompoundDrawablesWithIntrinsicBounds(0, R.c
tabLayout.getTabAt(0).setCustomView(tabOne);
```

### MainActivity.java

```
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 android.view.LayoutInflater;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.List;

import info.androidhive.materialtabs.R;
import info.androidhive.materialtabs.fragments.OneFra
import info.androidhive.materialtabs.fragments.ThreeF
import info.androidhive.materialtabs.fragments.TwoFra

public class MainActivity extends AppCompatActivity {

    private Toolbar toolbar;
    private TabLayout tabLayout;
    private ViewPager viewPager;
    private int[] tabIcons = {
        R.drawable.ic_tab_favourite,
        R.drawable.ic_tab_call,
        R.drawable.ic_tab_contacts
    };

    @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);

        viewPager = (ViewPager) findViewById(R.id.vie
        setupViewPager(viewPager);

    }

    private void setupViewPager(ViewPager viewPager) {
        ViewPagerAdapter adapter = new ViewPagerAdapter(getSupportFragmentManager());
        adapter.addFragment(new OneFragment(), "One");
        adapter.addFragment(new TwoFragment(), "Two");
        adapter.addFragment(new ThreeFragment(), "Three");
        viewPager.setAdapter(adapter);
    }
}
```

```
tabLayout = (TabLayout) findViewById(R.id.tabLayout);
tabLayout.setupWithViewPager(viewPager);
setupTabIcons();
}

private void setupTabIcons() {

    TextView tabOne = (TextView) LayoutInflater.from(this).inflate(R.layout.custom_tab, null);
    tabOne.setText("ONE");
    tabOne.setCompoundDrawablesWithIntrinsicBounds(0, 0, 0, 0);
    tabLayout.getTabAt(0).setCustomView(tabOne);

    TextView tabTwo = (TextView) LayoutInflater.from(this).inflate(R.layout.custom_tab, null);
    tabTwo.setText("TWO");
    tabTwo.setCompoundDrawablesWithIntrinsicBounds(0, 0, 0, 0);
    tabLayout.getTabAt(1).setCustomView(tabTwo);

    TextView tabThree = (TextView) LayoutInflater.from(this).inflate(R.layout.custom_tab, null);
    tabThree.setText("THREE");
    tabThree.setCompoundDrawablesWithIntrinsicBounds(0, 0, 0, 0);
    tabLayout.getTabAt(2).setCustomView(tabThree)
}

private void setupViewPager(ViewPager viewPager) {
    ViewPagerAdapter adapter = new ViewPagerAdapter(viewPager);
    adapter.addFrag(new OneFragment(), "ONE");
    adapter.addFrag(new TwoFragment(), "TWO");
    adapter.addFrag(new ThreeFragment(), "THREE");
    viewPager.setAdapter(adapter);
}

class ViewPagerAdapter extends FragmentPagerAdapter {
    private final List<Fragment> mFragmentList = new ArrayList<Fragment>();
    private final List<String> mFragmentTitleList = new ArrayList<String>();

    public ViewPagerAdapter(FragmentManager manager) {
        super(manager);
    }

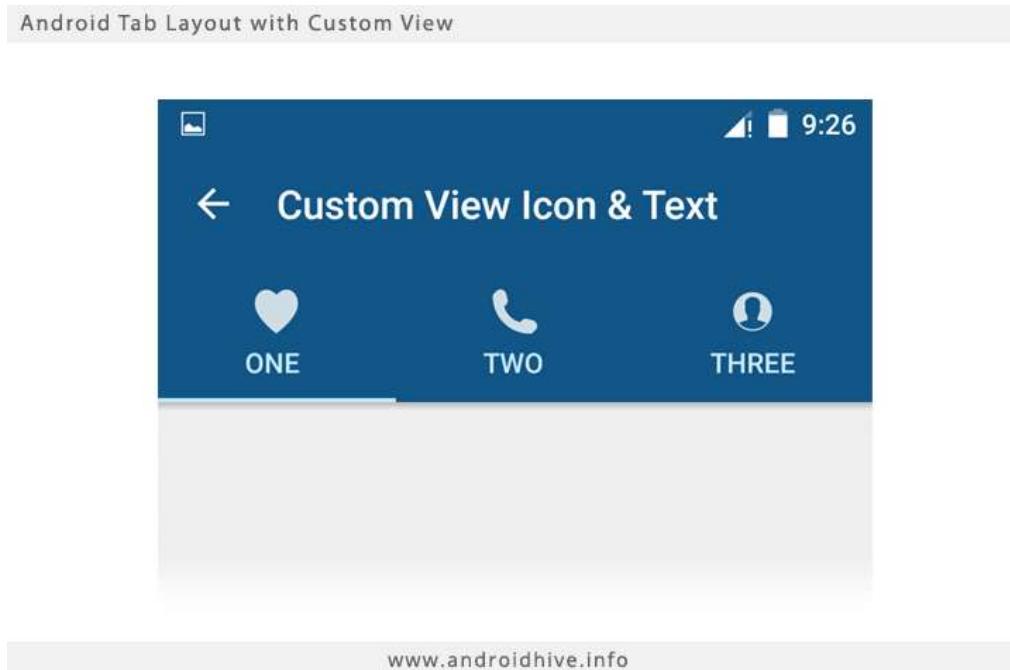
    @Override
    public Fragment getItem(int position) {
        return mFragmentList.get(position);
    }

    @Override
    public int getCount() {
        return mFragmentList.size();
    }

    public void addFrag(Fragment fragment, String title) {
        mFragmentList.add(fragment);
        mFragmentTitleList.add(title);
    }
}
```

```
@Override  
public CharSequence getPageTitle(int position)  
    return mFragmentTitleList.get(position);  
}  
}  
}
```

Now if you run the app, you can see the icon placed above the tab label.



I hope this article provided useful information about the tab layout using design support library. If you have any queries please do comment below.



Subscribe to our  
Newsletter!

Join our 746,498 subscribers and

get instant access to the latest  
android tutorials, app reviews and  
much more!

Email

SUBSCRIBE

---

## ABOUT THE AUTHOR

---



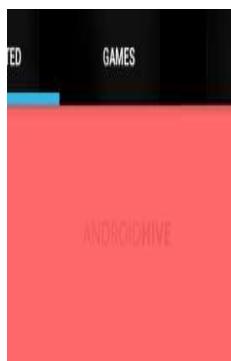
### Ravi Tamada

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

---



**Android Tab  
Layout with  
Swipeable  
Views**

by Ravi Tamada



**Android  
Working with  
Recycler View**  
by Ravi Tamada



**Android Using  
External Fonts**  
by Ravi Tamada



**Android  
Material  
Design  
Snackbar  
Example**

[Comments](#)[Community](#)

▼

[Recommend 33](#)[Share](#)[Sort by Newest ▾](#)[Join the discussion...](#)**sasi** • 4 days ago

Hi Ravi,

Thanks for the gud tutorial, the code is working fine but i want to connect to bluetooth automatically when i click/scroll to another tab. can u pls help.

Can u pls let me know how to add onclick listner to the tabs.. or is it possible to add onclicklistener.... or is there another way to perform the same...

[^](#) | [▼](#) • [Reply](#) • [Share >](#)**Druckerschwaerze** • 6 days ago

thank you!!

[^](#) | [▼](#) • [Reply](#) • [Share >](#) **imma\_newbie** • 6 days ago

i would like to know how can i modify the back button on the toolbar?

[^](#) | [▼](#) • [Reply](#) • [Share >](#)**George** ➔ **imma\_newbie** • 6 days ago

you have to delete this line in the OnCreate method in main activity ...

```
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
```

[^](#) | [▼](#) • [Reply](#) • [Share >](#)**Tasmia Rahman** • 9 days ago

Can you please do a tutorial on this :

<https://raw.githubusercontent.com/>....

and explain the codes along with it?

[^](#) | [▼](#) • [Reply](#) • [Share >](#) **kaushal** • 10 days ago

Hi....can we set different background colors for each tab...is it

possible

^ | v • Reply • Share >

**User X** • 11 days ago

Hey Ravi, I'm making a module based on this example, but I realized that my viewpager doesn't change the fragment when I change the selected Tab. Why in this example the change happens without setting manually which fragment must be shown?

^ | v • Reply • Share >

**goxx00** • 12 days ago

Hey :) i have a problem while step 12.

This is my error-code:

```
java.lang.NoSuchFieldError: No static field AppCompatTheme  
of type [I in class Landroid/support/v7/appcompat/R$styleable;  
or its superclasses (declaration of  
'android.support.v7.appcompat.R$styleable' appears in  
/data/data/com.example.gcp8fe.myapplication/files/instant-  
run/dex/slice-slice_2-classes.dex)
```

i hope you can help me. :)

^ | v • Reply • Share >

**Rick** ➔ goxx00 • 12 days ago

have you rebuild the gradle first before run the app?

these are my settings:

```
compileSdkVersion 23  
buildToolsVersion "23.0.1"
```

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    compile 'com.android.support:appcompat-v7:23.0.1'  
    compile 'com.android.support:design:23.0.1'  
}
```

My suggestion is to build the gradle first after fixing your build.gradle file.

^ | v • Reply • Share >

**Luthfi M Nabil** • 12 days ago

Works Nicely..

Thanks :)

^ | v • Reply • Share >

**Jovie Brett Bardoles** • 14 days ago

Hey man, do you know how to change the font of the tab texts?

^ | v • Reply • Share >

**User X** • 17 days ago

FATAL EXCEPTION: main

Process: com.example.curiousapps.ahtabs, PID: 16262

java.lang.NoClassDefFoundError: Failed resolution of:

Landroid/support/v7/internal/widget/TintManager;

On this line => tabLayout.setupWithViewPager(viewPager);

Why is this error appearing? I get stucked in step 12.

^ | v • Reply • Share >

**Rick** • 17 days ago

**@Ravi Tamada** Can you make a tutorial of a simple TextView displaying anything but compatible with every version of Android starting from 2.2, 2.3.3 and so on?? I don't think it will take much of your time, so it would be nice if you do that ASAP.

^ | v • Reply • Share >



**jeevan** ➔ Rick • 13 days ago

bro in xml layout declare<textview>  
then make sure that while ur creating ur app u choose  
api8 or android froyo

^ | v • Reply • Share >

**Rick** ➔ jeevan • 13 days ago

bro, tried it but nothing worked. Infact getting many errors now. Can you go through this link <https://www.dropbox.com/sh/nhc...> and fix the errors to make it work? Or at least, point me the errors I'm making?

^ | v • Reply • Share >

**Rick** ➔ jeevan • 13 days ago

Dude, tried it but nothing working. Infact, getting too many errors.  
I'm giving you the link of my source code. Can you please take a look and fix it for me? Or at least, point me the errors I'm making?

www.purplemath.com

Link: <https://www.dropbox.com/sh/nhc...>

^ | v • Reply • Share >

**Atul** • 18 days ago

Why to select "Blank Activity" and add tabs afterwards when it gives option "Tabbed Activity" itself ?

^ | v • Reply • Share >



Akash shrestha · 19 days ago

how to increase the size of icon in tab?? please help...

^ | v • Reply • Share >

**Inzimam Tariq** · 20 days ago

I started with the same instructions and its working at the 1st level, implementing Material Design to the toolbar But when I added fragments and their code app crashes and does not show anything. What to do now ?

^ | v • Reply • Share >

Ravi Tamada Mod → Inzimam Tariq · 19 days ago

Check the LogCat for errors.

1 ^ | v · Reply · Share >

**amaterasu07** → Ravi Tamada · 14 days ago

Hello. How do you handle the back button press on the left of the tabs?

^ | v • Reply • Share >

Rick → Ravi Tamada · 17 days ago

**@Ravi Tamada** Can you make a tutorial of a simple TextView displaying anything but compatible with every version of Android starting from 2.2, 2.3.3 and so on?? I don't think it will take much of your time, so it would be nice if you do that ASAP. Thanks.

^ | v • Reply • Share >

**Suraj Bobade** · 20 days ago

Hey, I have created one app, and now want to integrate two of activities in single tabbed activity.

## How to do that?

^ | v • Reply • Share >

**Nishu sharma** • 21 days ago

just one question, why you used CharSequence instead of String ??

^ | v • Reply • Share >

**Nishu sharma** • 21 days ago

hello ravi sir,

i am getting compile time error at line-13 in MainActivity.java

error - cannot resolve method

'setupViewPager(android.support.v4.view.ViewPager)'

any idea whats wrong??

sorry, its resolved, how can i delete this comment :P

and many many thanks to you Dear Ravi :)

^ | v • Reply • Share >

**Rakhi Dhavale** • 22 days ago

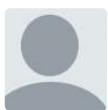
hi, thanks for making a post on this topic, I don't the text to remain capital, I have defined a custom style for that purpose, but still its not helping. What can be done?

^ | v • Reply • Share >

**Ravi Tamada** Mod ➔ Rakhi Dhavale • 21 days ago

Can you give me the code you defined in the styles?

^ | v • Reply • Share >



**Simar** • 23 days ago

How to change the Text Colors in the Tabs ?

^ | v • Reply • Share >

**Naufal Prakoso** • 23 days ago

I have 2 fragments and in a fragments have a different java logic. Where i am i put the java logic? When i put to the fragment, 'cannot resolve method findViewById', "Cannont resolve method getApplicationContext"

^ | v • Reply • Share >

**Joe Higashi** • 25 days ago

Click not Working =\

^ | v • Reply • Share >

**Surinder Rawat** • 25 days ago

I am using tab Layout. By default there is always a single tab opened. But the problem is that the single tab is not taking entire tab layout width i.e. the screen width. It is fixing on the center. I have tried adding tabGravity as 'fill' and tabMaxWidth as '0dp'.

1 ^ | v • Reply • Share >

**Akshay Kumar** • a month ago

Hello Sir, I am using same for Icon Text Tab but in my application icon and text are not coming side by side.

^ | v • Reply • Share >

**August** ↗ Akshay Kumar • 7 days ago

same here. did you solved it yet?

^ | v • Edit • Reply • Share >

**Alfian Effendi** • a month ago

no errors but always force closed. help please

^ | v • Reply • Share >

**Pavitra Kansara** • a month ago

Great article. Although, how would you add a DrawerLayout and NavigationView to the layout file in order to add a Navigation Drawer instead of ActionBar home button?

^ | v • Reply • Share >

**Wesley Chaves** • a month ago

what to do if you want the app already open in tabs (Custom View Icon & Text) type the layout of Whatsapp?

1 ^ | v • Reply • Share >

**Nattawut Kongtan** • a month ago

Hoe can I fix Missing styles. Is the correct theme chosen for this layout?

Thank you

^ | v • Reply • Share >

**Beghoura Farouk** • a month ago

tabLayout.setupWithViewPager(viewPager);

The app crash when i add this line any idea ???

^ | v • Reply • Share >

**emmarc** ↗ Beghoura Farouk • 19 days ago

I don't know if this can help you.  
I have the same problem not long ago, but I already fixed it with checking the build.gradle file. Apparently at that time, there were an inconsistency between com.android.support:appcompat and com.android.support:design version in my build.gradle file. So I changed them both to the same version. And voila!! The code can run just fine.

THESE LINES:

```
compile 'com.android.support:appcompat-v7:23.0.1'  
compile 'com.android.support:design:23.0.1'
```

^ | v • Reply • Share >

**Awera Suk** • a month ago

This tutorial work fine for static pages. How about if i want to change the view. If i have 4 tabs and 3 of them are displayed and one will be called on the action taken from the 3 tabs. When u implemented drawer u used frame\_container to replace the fragments with fragment transaction. How to replace the view pager to replace it with new fragment.

I have tried this and it is not working. That is why I am asking

```
FragmentTransaction trans =  
getActivity().getSupportFragmentManager().beginTransaction();  
trans.replace(R.id.viewpager, new XYZFragment());  
trans.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_CLOSE);  
trans.addToBackStack(null);  
trans.commit();
```

Thanks

^ | v • Reply • Share >

 **DonV** • a month ago

Hey..in two tabs i want to start to an activity.how to start an activity. If i do, i can only inflate my activity layout but not activity.please help me

^ | v • Reply • Share >

**Ahmet Keser** • 2 months ago

Hi Ravi,

Instead of keeping tablayout instance on Activity class, can I

apply same approach on Fragment,

I mean there will be a one fragment and on it there will be 3 different Fragment

is it possible?

^ | v • Reply • Share >

**Anurag Singh** • 2 months ago

what if we want to remove that "Back Option" icon (which is on top left corner) from app?

^ | v • Reply • Share >

**Pankaj Lilan** ➔ Anurag Singh • 2 months ago

**@Anurag Singh** you can write

getSupportActionBar().setDisplayHomeAsUpEnabled(true)

^ | v • Reply • Share >

**Gaurav meghanathi** • 2 months ago

i want to tab in page of bottom with text and image how i can set...?

^ | v • Reply • Share >

**Io Roy** ➔ Gaurav meghanathi • 18 days ago

add

android:layout\_gravity="bottom"

---

## QUICK CONTACT

---

- [Advertise with us](#) AndroidHive is beginner's
- [Privacy Policy](#) paradise for android
- [Terms of Service](#) tutorials, tips & tricks, games,
- [Sitemap](#) app reviews, hacks and super cool advanced topics.

---

## ABOUT ANDROIDHIVE

---