

Android Getting Started with Material Design

424 Comments . By [Ravi Tamada](#)

Like 332

Tweet

G+1 279

You might have heard of android [Material Design](#) which was introduced in Android Lollipop version. In Material Design lot of new things were introduced like **Material Theme**, new **widgets**, **custom shadows**, **vector drawables** and custom **animations**. If you haven't working on Material Design yet, this article will give you a good start.

In this tutorial we are going to learn the basic steps of Material Design development i.e writing the custom theme and implementing the navigation drawer using the [RecyclerView](#).

Go through the below links to get more knowledge over Material Design.

> [Material Design Specifications](#)

> [Creating Apps with Material Design](#)

DOWNLOAD CODE

VIDEO DEMO

Android Getting Started with Material Design (Demo)



1. Downloading Android Studio

Before going further, download the [Android Studio](#) and do the necessary setup as I am going to use Android Studio for all my tutorial from now on. If you are trying the Android Studio for the first time, go the [overview](#) doc to get complete overview of android studio.

2. Material Design Color Customization

Material Design provides set of properties to customize the Material Design Color theme. But we use five primary attributes to customize overall theme.

colorPrimaryDark – This is darkest primary color of the app mainly applies to notification bar background.

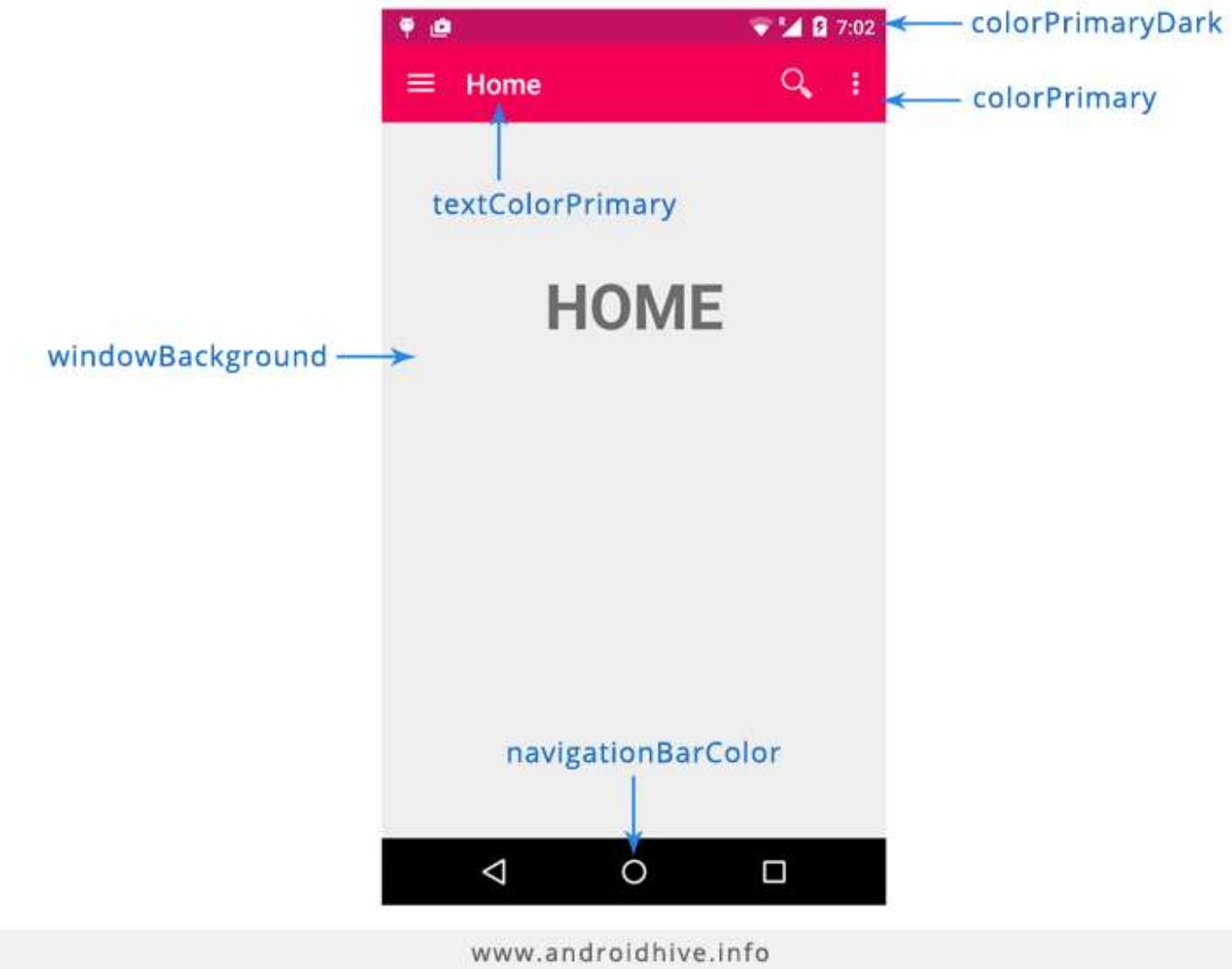
colorPrimary – This is the primary color of the app. This color will be applied as toolbar background.

textColorPrimary - This is the primary color of text. This applies to toolbar title.

windowBackground - This is the default background color of the app.

navigationBarColor - This color defines the background color of footer navigation bar.

Android Material Design



You can go through this material design [color patterns](#) and choose the one that suits your app.

3. Creating Material Design Theme

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 **res** ⇒ **values** ⇒ **strings.xml** and add below string values.

STRINGS.XML

```
<resources>
    <string name="app_name">Material Design</string>
    <string name="action_settings">Settings</string>
    <string name="action_search">Search</string>
    <string name="drawer_open">Open</string>
    <string name="drawer_close">Close</string>

    <string name="nav_item_home">Home</string>
    <string name="nav_item_friends">Friends</string>
    <string name="nav_item_notifications">Messages</string>

    <!-- navigation drawer item labels -->
    <string-array name="nav_drawer_labels">
        <item>@string/nav_item_home</item>
        <item>@string/nav_item_friends</item>
        <item>@string/nav_item_notifications</item>
    </string-array>

    <string name="title_messages">Messages</string>
    <string name="title_friends">Friends</string>
    <string name="title_home">Home</string>
</resources>
```

3. Open **res ⇒ values ⇒ colors.xml** and add the below color values. If you don't find colors.xml, create a new resource file with the name.

COLORS.XML

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#F50057</color>
    <color name="colorPrimaryDark">#C51162</color>
    <color name="textColorPrimary">#FFFFFF</color>
    <color name="windowBackground">#FFFFFF</color>
    <color name="navigationBarColor">#000000</color>
    <color name="colorAccent">#FF80AB</color>
</resources>
```

4. Open **res ⇒ values ⇒ dimens.xml** and add below dimensions.

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="nav_drawer_width">260dp</dimen>
</resources>
```

5. Open **styles.xml** under **res ⇒ values** and add below styles. The styles defined in this **styles.xml** are common to all the android versions. Here I am naming my theme as **MyMaterialTheme**.

```
STYLES.XML
<resources>

    <style name="MyMaterialTheme" parent="MyMaterialTheme.Base">

        <style name="MyMaterialTheme.Base" parent="Theme.AppCompat.Light.DarkActionBar">
            <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 Lollipop** only.

```
STYLES.XML
<resources>

    <style name="MyMaterialTheme" parent="MyMaterialTheme.Base">
        <item name="android:windowContentTransitions">true</item>
        <item name="android:windowAllowEnterTransitionOverlap">true</item>
        <item name="android:windowAllowReturnTransitionOverlap">true</item>
        <item name="android:windowSharedElementEnterTransition">@android:transition/fade</item>
        <item name="android:windowSharedElementExitTransition">@android:transition/fade</item>
    </style>

</resources>
```

7. Now we have the basic Material Design styles ready. In order to apply the theme, open **AndroidManifest.xml** and modify the **android:theme** attribute of **<application>** tag.

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

So after applying the theme, your **AndroidManifest.xml** should look like below.

ANDROIDMANIFEST.XML

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

    <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>
```

Now if you run the app, you can see the notification bar color changed to the color that we have mentioned in our styles.

Android Material Design



3.1 Adding the Toolbar (Action Bar)

Adding the toolbar is very easy. All you have to do is, create a separate layout for the toolbar and include it in other layout wherever you want the toolbar to be displayed.

8. Create an xml file named **toolbar.xml** under **res** ⇒ **layout** and add **android.support.v7.widget.Toolbar** element. This create the toolbar with specific height and theming.

TOOLBAR.XML

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.Toolbar xmlns:android="http://schemas.android.com/apk/
    xmlns:local="http://schemas.android.com/apk/res-auto"
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:minHeight="?attr/actionBarSize"
    android:background="?attr/colorPrimary"
    local:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
    local:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
```

9. Open the layout file of your main activity (**activity_main.xml**) and add the **toolbar** using **<include>** tag.

ACTIVITY_MAIN.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=".MainActivity">

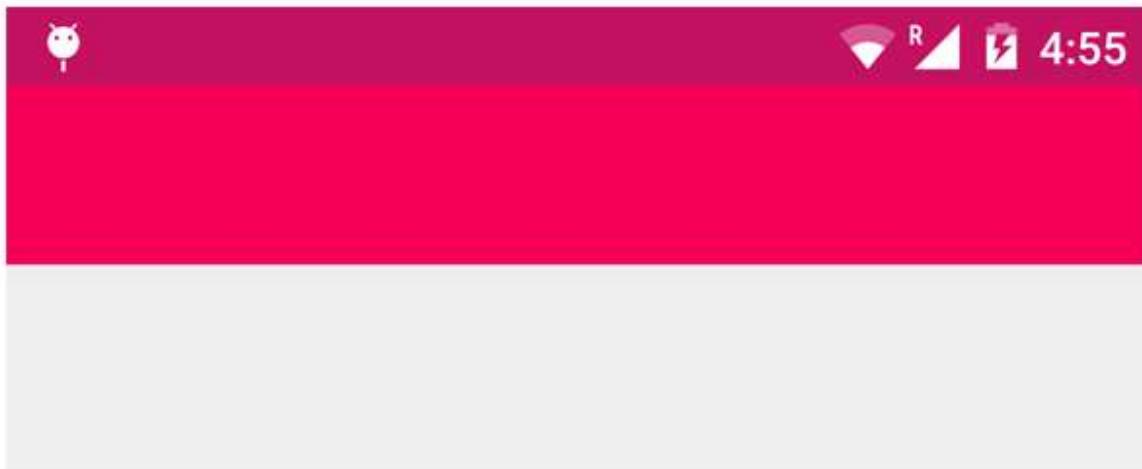
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:orientation="vertical">

        <include
            android:id="@+id/toolbar"
            layout="@layout/toolbar" />
    </LinearLayout>

</RelativeLayout>
```

Run the app and see if the toolbar displayed on the screen or not.

Android Material Design - Toolbar

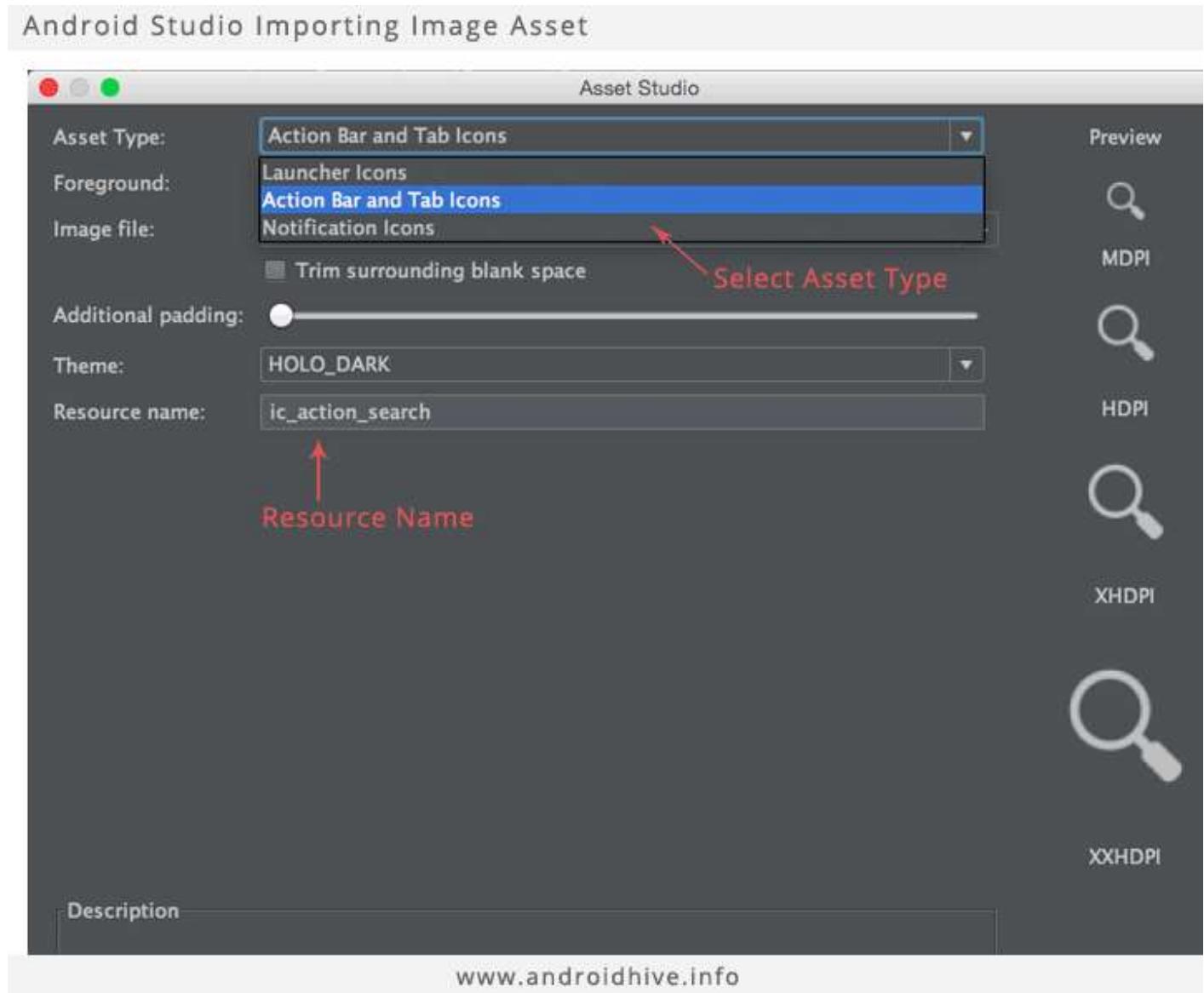


Now let's try to add a toolbar title and enable the action items.

10. Download this [search](#) icon and import it into Android Studio as a Image Asset.

11. To import the Image Asset in Android Studio, right click on **res** ⇒ **New** ⇒ **Image Asset**. It will show you a popup window to import the resource. Browse the search icon that you have downloaded in the

above step, select **Action Bar and Tab Icons** for Asset Type and give the resource name as **ic_search_action** and proceed.



12. Once the icon is imported, open **menu_main.xml** located under **res ⇒ menu** and add the search menu item as mentioned below.

```
MENU_MAIN.XML
<menu 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"
    tools:context=".MainActivity">

    <item
        android:id="@+id/action_search"
        android:title="@string/action_search"
        android:orderInCategory="100"
        android:icon="@drawable/ic_action_search"
        app:showAsAction="ifRoom" />

```

```

<item
    android:id="@+id/action_settings"
    android:title="@string/action_settings"
    android:orderInCategory="100"
    app:showAsAction="never" />
</menu>

```

13. Now open your **MainActivity.java** and do the below changes.

- > Extend the activity from **AppCompatActivity**
- > Enable the toolbar by calling **setSupportActionBar()** by passing the toolbar object.
- > Override **onCreateOptionsMenu()** and **onOptionsItemSelected()** methods to enable toolbar action items.

MAINACTIVITY.JAVA

```

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {

    private Toolbar mToolbar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mToolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(mToolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will

```

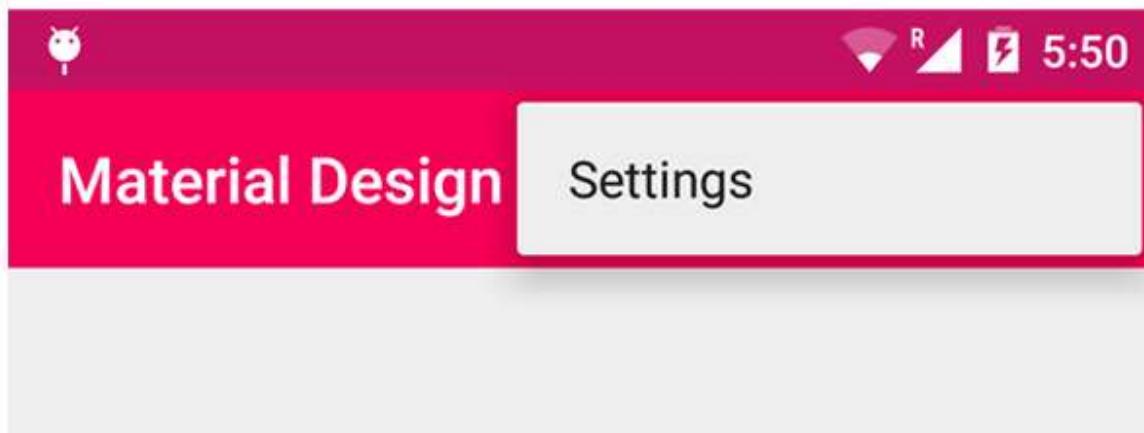
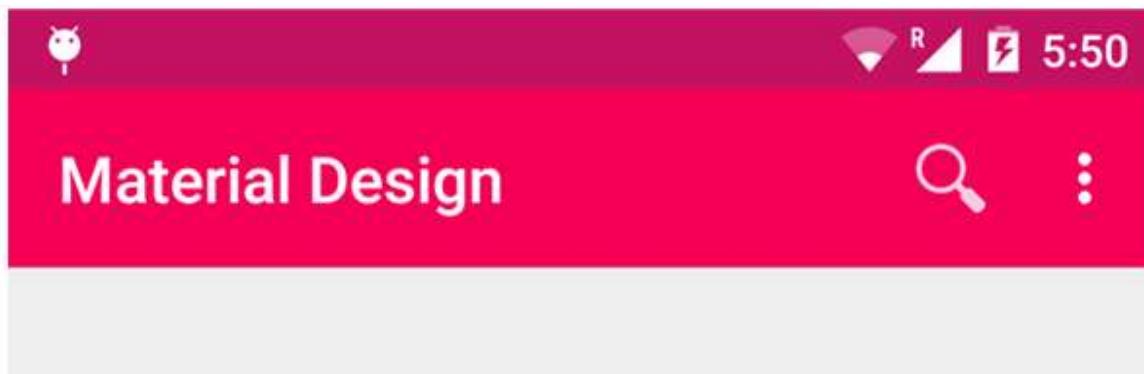
```
// automatically handle clicks on the Home/Up button, so long
// as you specify a parent activity in AndroidManifest.xml.
int id = item.getItemId();

//noinspection SimplifiableIfStatement
if (id == R.id.action_settings) {
    return true;
}

return super.onOptionsItemSelected(item);
}
```

After doing the above changes, if you run the app, you should see the search icon and action overflow icon.

Android Material Design Toolbar Action Items



3.2 Adding Navigation Drawer

Adding navigation drawer is same as that we do before lollipop, but instead if using ListView for menu items, we use [RecyclerView](#) in material design. So let's see how to implement the navigation drawer with RecyclerView.

14. In your project's java folder, create three packages named **activity**, **adapter**, **model** and move your **MainActivity.java** to activity package. This will keep your project organized.

15. Open **build.gradle** located under your **app** module and add below dependencies. After adding the dependencies, goto **Build ⇒ Rebuild Project** to download required libraries.

BUILD.GRADLE

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.2.0'
    compile 'com.android.support:recyclerview-v7:22.2.+'
}
```

16. Under **model** package, create a class named **NavDrawerItem.java** with the below code. This model class is POJO class that defines each row in navigation drawer menu.

NAVDRAWERITEM.JAVA

```
package info.androidhive.materialdesign.model;

/**
 * Created by Ravi on 29/07/15.
 */
public class NavDrawerItem {
    private boolean showNotify;
    private String title;

    public NavDrawerItem() {
    }

    public NavDrawerItem(boolean showNotify, String title) {
        this.showNotify = showNotify;
        this.title = title;
    }

    public boolean isShowNotify() {
        return showNotify;
    }
}
```

```

public void setShowNotify(boolean showNotify) {
    this.showNotify = showNotify;
}

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}
}

```

- 17.** Under **res ⇒ layout**, create an xml layout named **nav_drawer_row.xml** and add the below code. The layout renders each row in navigation drawer menu. If you want to customize the navigation drawer menu item, you have to do the changes in this file. For now it has only one TextView.

```

NAV_DRAWER_ROW.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="wrap_content"
    android:clickable="true">

    <TextView
        android:id="@+id/title"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingLeft="30dp"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:textSize="15dp"
        android:textStyle="bold" />

</RelativeLayout>

```

- 18.** Download this profile [icon](#) and paste it in your drawable folder. This step is optional, but this icon used in the navigation drawer header part.

- 19.** Create another xml layout named **fragment_navigation_drawer.xml** and add the below code. This layout renders the complete navigation drawer view. This layout contains a header section to display the user information and a RecyclerView to display the list view.

FRAGMENT_NAVIGATION_DRAWER.XML

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/white">

    <RelativeLayout
        android:id="@+id/nav_header_container"
        android:layout_width="match_parent"
        android:layout_height="140dp"
        android:layout_alignParentTop="true"
        android:background="@color/colorPrimary">

        <ImageView
            android:layout_width="70dp"
            android:layout_height="70dp"
            android:src="@drawable/ic_profile"
            android:scaleType="fitCenter"
            android:layout_centerInParent="true" />

    </RelativeLayout>

    <android.support.v7.widget.RecyclerView
        android:id="@+id/drawerList"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/nav_header_container"
        android:layout_marginTop="15dp" />

</RelativeLayout>

```

20. As the RecyclerView is customized, we need an adapter class to render the custom xml layout. So under adapter package, create a class named **NavigationDrawerAdapter.java** and paste the below code. This adapter class inflates **nav_drawer_row.xml** and renders the **RecycleView** drawer menu.

```

import android.content.Context;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import java.util.Collections;
import java.util.List;

/**
 * Created by Ravi Tamada on 12-03-2015.
 */
public class NavigationDrawerAdapter extends RecyclerView.Adapter<NavigationDrawerAdapter.NavDrawerItem> {
    List<NavDrawerItem> data = Collections.emptyList();
    private LayoutInflater inflater;
    private Context context;

```

```

public NavigationDrawerAdapter(Context context, List<NavDrawerItem> data) {
    this.context = context;
    inflater = LayoutInflater.from(context);
    this.data = data;
}

public void delete(int position) {
    data.remove(position);
    notifyItemRemoved(position);
}

@Override
public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View view = inflater.inflate(R.layout.nav_drawer_row, parent, false);
    MyViewHolder holder = new MyViewHolder(view);
    return holder;
}

@Override
public void onBindViewHolder(MyViewHolder holder, int position) {
    NavDrawerItem current = data.get(position);
    holder.title.setText(current.getTitle());
}

@Override
public int getItemCount() {
    return data.size();
}

class MyViewHolder extends RecyclerView.ViewHolder {
    TextView title;

    public MyViewHolder(View itemView) {
        super(itemView);
        title = (TextView) itemView.findViewById(R.id.title);
    }
}
}

```

21. Under **activity** package, create a fragment named **FragmentDrawer.java**. In Android Studio, to create a new fragment, **Right click on activity** ⇒ **New** ⇒ **Fragment** ⇒ **Fragment (Blank)** and give your fragment class name.

FRAGMENTDRAWER.JAVA

```

/**
 * Created by Ravi on 29/07/15.
 */

import android.content.Context;
import android.os.Bundle;
import android.support.v4.app.Fragment;

```

```
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.view.GestureDetector;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;

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

import info.androidhive.materialdesign.R;
import info.androidhive.materialdesign.adapter.NavigationDrawerAdapter;
import info.androidhive.materialdesign.model.NavDrawerItem;

public class FragmentDrawer extends Fragment {

    private static String TAG = FragmentDrawer.class.getSimpleName();

    private RecyclerView recyclerView;
    private ActionBarDrawerToggle mDrawerToggle;
    private DrawerLayout mDrawerLayout;
    private NavigationDrawerAdapter adapter;
    private View containerView;
    private static String[] titles = null;
    private FragmentDrawerListener drawerListener;

    public FragmentDrawer() {

    }

    public void setDrawerListener(FragmentDrawerListener listener) {
        this.drawerListener = listener;
    }

    public static List<NavDrawerItem> getData() {
        List<NavDrawerItem> data = new ArrayList<>();

        // preparing navigation drawer items
        for (int i = 0; i < titles.length; i++) {
            NavDrawerItem navItem = new NavDrawerItem();
            navItem.setTitle(titles[i]);
            data.add(navItem);
        }
        return data;
    }

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

        // drawer labels
        titles = getActivity().getResources().getStringArray(R.array.nav_drawer_);
    }
}
```

```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    // Inflating view layout
    View layout = inflater.inflate(R.layout.fragment_navigation_drawer, container);
    recyclerView = (RecyclerView) layout.findViewById(R.id.drawerList);

    adapter = new NavigationDrawerAdapter(getActivity(), getData());
    recyclerView.setAdapter(adapter);
    recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));
    recyclerView.addOnItemTouchListener(new RecyclerTouchListener(getActivity(),
        @Override
        public void onClick(View view, int position) {
            drawerListener.onDrawerItemSelected(view, position);
            mDrawerLayout.closeDrawer(containerView);
        }
    });

    @Override
    public void onLongClick(View view, int position) {

    }
});

return layout;
}

public void setUp(int fragmentId, DrawerLayout drawerLayout, final Toolbar toolbar) {
    containerView = getActivity().findViewById(fragmentId);
    mDrawerLayout = drawerLayout;
    mDrawerToggle = new ActionBarDrawerToggle(getActivity(), drawerLayout, toolbar, toolbar.getOverflowIcon());
    @Override
    public void onDrawerOpened(View drawerView) {
        super.onDrawerOpened(drawerView);
        getActivity().invalidateOptionsMenu();
    }

    @Override
    public void onDrawerClosed(View drawerView) {
        super.onDrawerClosed(drawerView);
        getActivity().invalidateOptionsMenu();
    }

    @Override
    public void onDrawerSlide(View drawerView, float slideOffset) {
        super.onDrawerSlide(drawerView, slideOffset);
        toolbar.setAlpha(1 - slideOffset / 2);
    }
};

mDrawerLayout.setDrawerListener(mDrawerToggle);
mDrawerLayout.post(new Runnable() {
    @Override
    public void run() {
        mDrawerToggle.syncState();
    }
});
}
}

```

```
public static interface ClickListener {
    public void onClick(View view, int position);

    public void onLongClick(View view, int position);
}

static class RecyclerTouchListener implements RecyclerView.OnItemTouchListener {

    private GestureDetector gestureDetector;
    private ClickListener clickListener;

    public RecyclerTouchListener(Context context, final RecyclerView recyclerView) {
        this.clickListener = clickListener;
        gestureDetector = new GestureDetector(context, new GestureDetector.SimpleOnGestureListener() {
            @Override
            public boolean onSingleTapUp(MotionEvent e) {
                return true;
            }

            @Override
            public void onLongPress(MotionEvent e) {
                View child = recyclerView.findChildViewUnder(e.getX(), e.getY());
                if (child != null && clickListener != null) {
                    clickListener.onLongClick(child, recyclerView.getChildPosition(child));
                }
            }
        });
    }

    @Override
    public boolean onInterceptTouchEvent(RecyclerView rv, MotionEvent e) {

        View child = rv.findChildViewUnder(e.getX(), e.getY());
        if (child != null && clickListener != null && gestureDetector.onTouchEvent(e)) {
            clickListener.onClick(child, rv.getChildPosition(child));
        }
        return false;
    }

    @Override
    public void onTouchEvent(RecyclerView rv, MotionEvent e) {
    }

    @Override
    public void onRequestDisallowInterceptTouchEvent(boolean disallowIntercept) {
    }
}

public interface FragmentDrawerListener {
    public void onDrawerItemSelected(View view, int position);
}
}
```

22. Finally open main activity layout (**activity_main.xml**) and modify the layout as below. In this layout we are adding **android.support.v4.widget.DrawerLayout** to display the navigation drawer menu.

Also you have to give the correct path of your FragmentDrawer in <fragment> element.

ACTIVITY_MAIN.XML

```
<android.support.v4.widget.DrawerLayout 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:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:id="@+id/container_toolbar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <include
                android:id="@+id/toolbar"
                layout="@layout/toolbar" />
        </LinearLayout>

        <FrameLayout
            android:id="@+id/container_body"
            android:layout_width="fill_parent"
            android:layout_height="0dp"
            android:layout_weight="1" />
    </LinearLayout>

    <fragment
        android:id="@+id/fragment_navigation_drawer"
        android:name="info.androidhive.materialdesign.activity.FragmentDrawer"
        android:layout_width="@dimen/nav_drawer_width"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        app:layout="@layout/fragment_navigation_drawer"
        tools:layout="@layout/fragment_navigation_drawer" />
</android.support.v4.widget.DrawerLayout>
```

Now we have all the layout files and java classes ready in place. Let's do the necessary changes in **MainActivity** to make the navigation drawer functioning.

23. Open your **MainActivity.java** and do the below changes.

- > Implement the activity from **FragmentDrawer.FragmentDrawerListener** and add the **onDrawerItemSelected()** override method.
- > Create an instance of **FragmentDrawer** and set the drawer selected listeners.

MAINACTIVITY.JAVA

```
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;

public class MainActivity extends AppCompatActivity implements FragmentDrawer.Fra

    private Toolbar mToolbar;
    private FragmentDrawer drawerFragment;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mToolbar = (Toolbar) findViewById(R.id.toolbar);

        setSupportActionBar(mToolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        drawerFragment = (FragmentDrawer)
            getSupportFragmentManager().findFragmentById(R.id.fragment_navigation_drawer);
        drawerFragment.setUp(R.id.fragment_navigation_drawer, (DrawerLayout) findViewById(R.id.drawer_layout));
        drawerFragment.setDrawerListener(this);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
    }
}
```

```
int id = item.getItemId();

//noinspection SimplifiableIfStatement
if (id == R.id.action_settings) {
    return true;
}

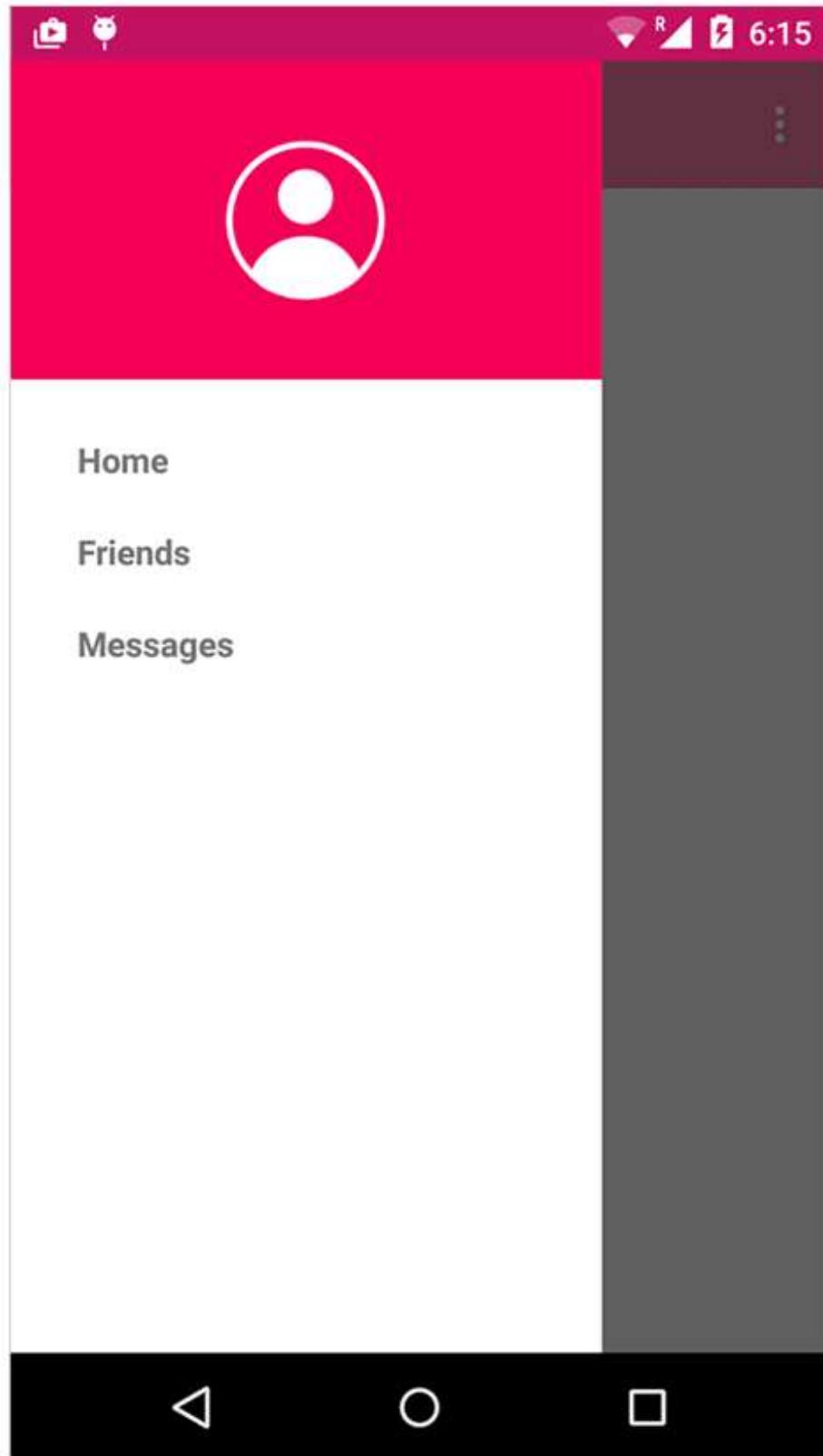
return super.onOptionsItemSelected(item);
}

@Override
public void onDrawerItemSelected(View view, int position) {

}
}
```

Now if you run the app, you can see the navigation drawer with a header and few list items in it.

Android Material Design Navigation Drawer



3.3 Implementing Navigation Drawer Item Selection

Although navigation drawer is functioning, you can see the selection of drawer list items not working. This is because we are yet to implement the click listener on RecyclerView items.

As we have three menu items in navigation drawer (**Home, Friends & Messages**), we need to create three separate fragment classes for each menu item.

24. Under res layout, create an xml layout named **fragment_home.xml** and add below code.

```
FRAGMENT_HOME.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"
    android:orientation="vertical"
    tools:context="info.androidhive.materialdesign.activity.HomeFragment">

    <TextView
        android:id="@+id/label"
        android:layout_alignParentTop="true"
        android:layout_marginTop="100dp"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:textSize="45dp"
        android:text="HOME"
        android:textStyle="bold"/>

    <TextView
        android:layout_below="@+id/label"
        android:layout_centerInParent="true"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="12dp"
        android:layout_marginTop="10dp"
        android:gravity="center_horizontal"
        android:text="Edit fragment_home.xml to change the appearance" />

</RelativeLayout>
```

25. Under **activity** package, create a fragment class named **HomeFragment.java** and add below code.

HOMEFRAGMENT.JAVA

```

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

public class HomeFragment extends Fragment {

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

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

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_home, container, false);

        // Inflate the layout for this fragment
        return rootView;
    }

    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
    }

    @Override
    public void onDetach() {
        super.onDetach();
    }
}

```

26. Create two more fragment classes named **FriendsFragment.java**, **MessagesFragment.java** and respected layout files named **fragment_friends.xml** and **fragment_messages.xml** and add the code from above two steps.

27. Now open **MainActivity.java** and do the below changes. In the below code

> **displayView()** method displays the fragment view respected the navigation menu item selection. This method should be called in **onDrawerItemSelected()** to render the respected view when a

navigation menu item is selected.

MAINACTIVITY.JAVA

```

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends ActionBarActivity implements FragmentDrawer.Fra
    private static String TAG = MainActivity.class.getSimpleName();

    private Toolbar mToolbar;
    private FragmentDrawer drawerFragment;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mToolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(mToolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        drawerFragment = (FragmentDrawer)
            getSupportFragmentManager().findFragmentById(R.id.fragment_navigation_drawer);
        drawerFragment.setUp(R.id.fragment_navigation_drawer, (DrawerLayout) findViewById(R.id.drawer_layout));
        drawerFragment.setDrawerListener(this);

        // display the first navigation drawer view on app launch
        displayView(0);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
    }
}

```

```
//noinspection SimplifiableIfStatement
if (id == R.id.action_settings) {
    return true;
}

if(id == R.id.action_search){
    Toast.makeText(getApplicationContext(), "Search action is selected!", Toast.LENGTH_SHORT).show();
    return true;
}

return super.onOptionsItemSelected(item);
}

@Override
public void onDrawerItemSelected(View view, int position) {
    displayView(position);
}

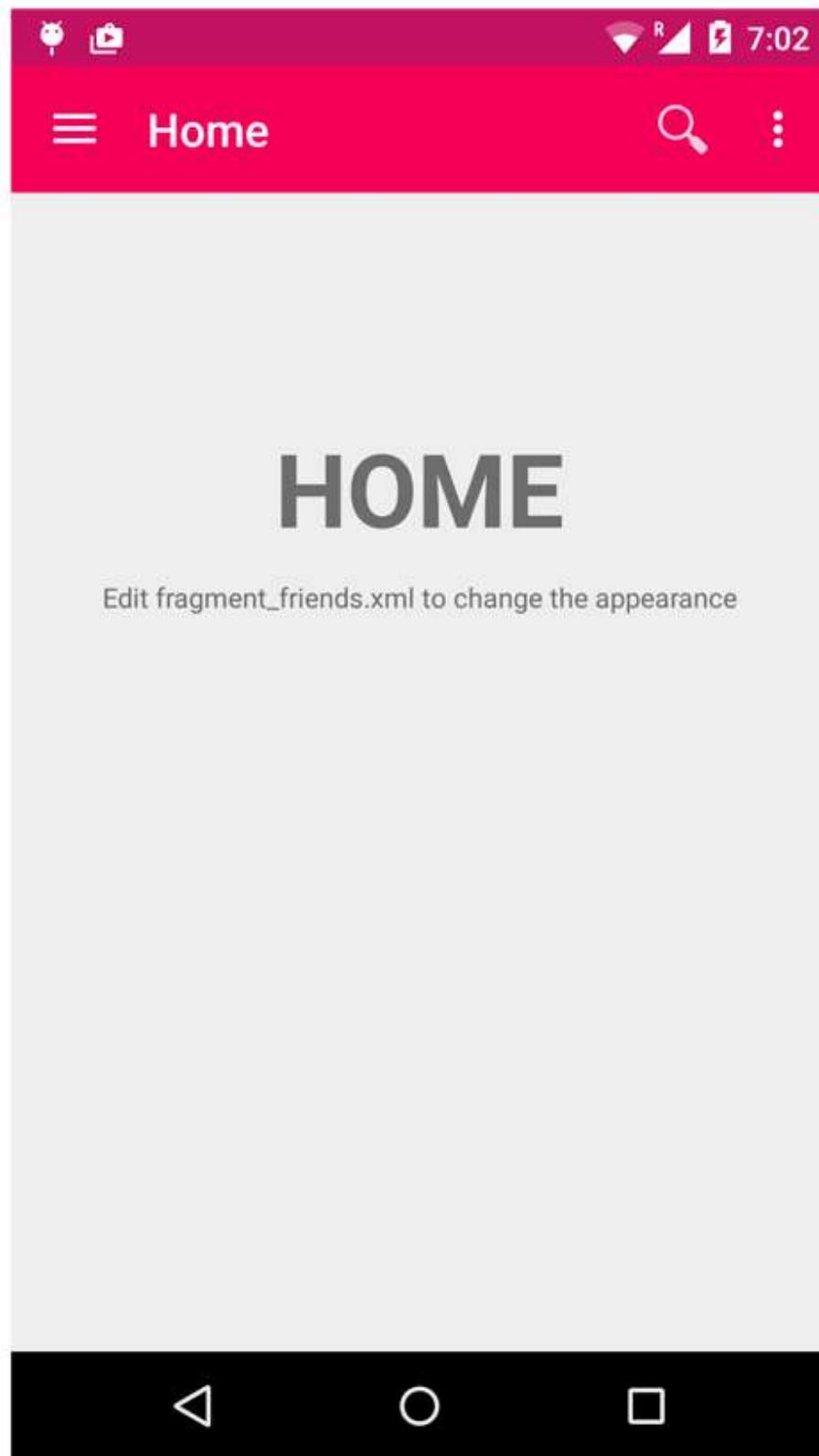
private void displayView(int position) {
    Fragment fragment = null;
    String title = getString(R.string.app_name);
    switch (position) {
        case 0:
            fragment = new HomeFragment();
            title = getString(R.string.title_home);
            break;
        case 1:
            fragment = new FriendsFragment();
            title = getString(R.string.title_friends);
            break;
        case 2:
            fragment = new MessagesFragment();
            title = getString(R.string.title_messages);
            break;
        default:
            break;
    }

    if (fragment != null) {
        FragmentManager fragmentManager = getSupportFragmentManager();
        FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
        fragmentTransaction.replace(R.id.container_body, fragment);
        fragmentTransaction.commit();

        // set the toolbar title
        getSupportActionBar().setTitle(title);
    }
}
}
```

Now if you run the app, you can see the selection of navigation drawer menu is working and respected view displayed below the toolbar.

Android Material Design Navigation Drawer



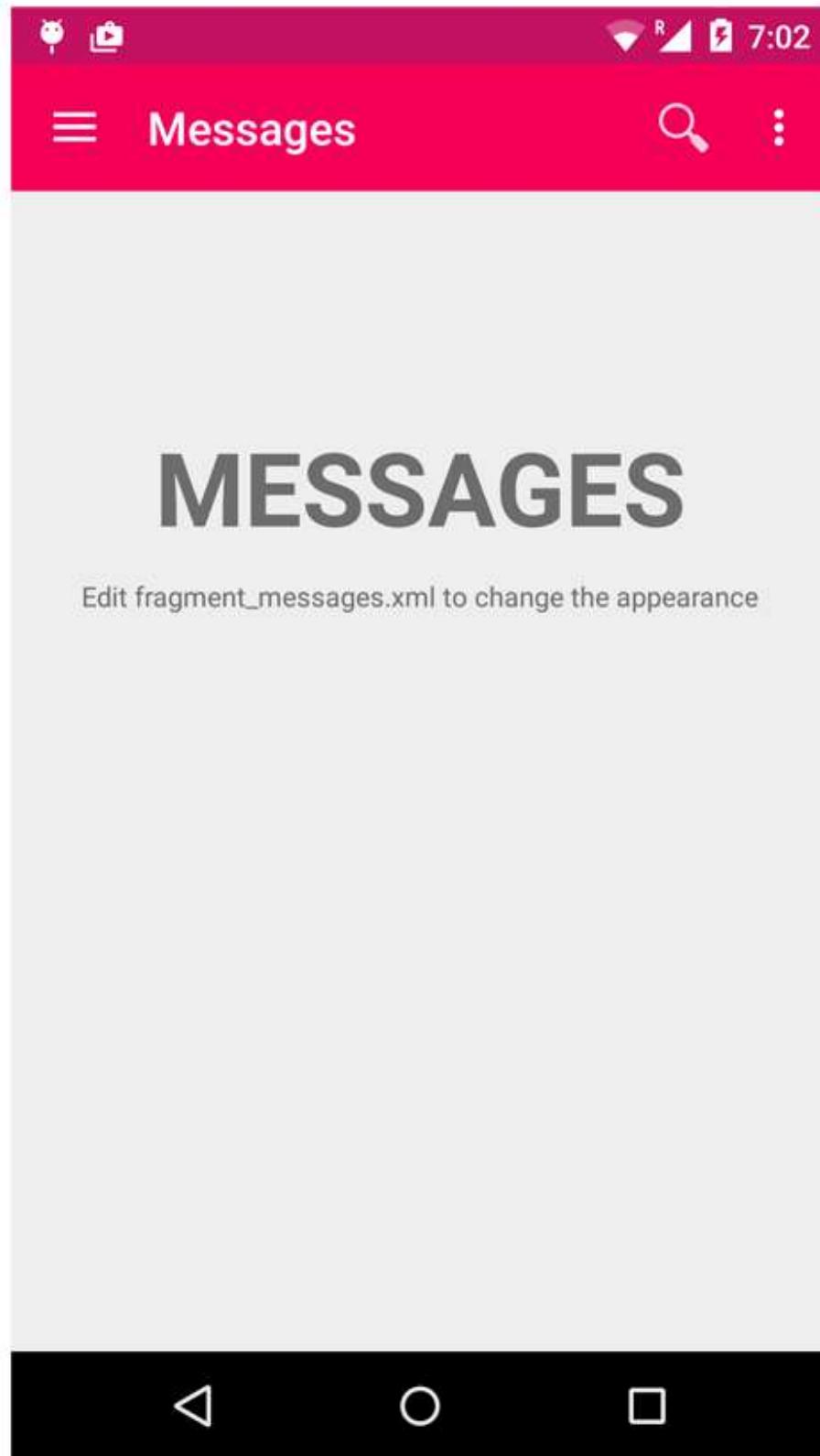
www.androidhive.info

Android Material Design Navigation Drawer



ANDROIDHIVE

Android Material Design Navigation Drawer



www.androidhive.info

What's Next?

Below are few more material components you can add to your app. These were implemented using recent Android Design Support Library.

1. Material Design Tab Layout

If you want to add tabs to your app, [Android Material Design Tabs](#) covers different aspects of Tab Layout.

2. Floating Labels for EditText

Learn how [floating labels](#) works on EditText with a simple form validation example.

3. Floating Action Button (FAB)

Add the [Floating Action Button](#) to your which displays in circular shape floating on the top of the UI.

4. Snackbar

Add the [Snackbar](#) to your app to give immediate feedback about any operation that user performed.

Change Log

Updated On	29th July 2015 (Latest support library)
------------	---

Share this article on



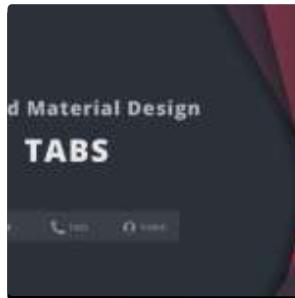
Tweet



Like

G+1

You May Also Like



Android Material Design
working with Tabs



Android Building Free
Wallpapers App – Part 2



Android Building Free
Wallpapers App – Part 1



Android Material Design
Floating Labels for
EditText

424 Comments [Android Hive](#)

1 August ▾

Recommend 47

Share

Sort by Newest ▾



Join the discussion...



raveena • 17 hours ago

Failed to find the style corresponding to the id 2130772027

Failed to convert ?attr/colorPrimary into a drawable

"?attr/actionBarSize" in attribute "minHeight" is not a valid format.

Exception details are logged in Window > Show View > Error Log
come in error in layout of main

• Reply • Share >



Iis Afriyanti • 21 hours ago

This is really helpful! Thanks!

• Reply • Share >



Rishi Mishra • 8 days ago

heyy boss i am new in android development would you please tell me how can i create a app using material design and fetch data of blogger.

or is there any way that i download the xml file from blogger and insert it into my android app

but i want to create a fully automatic app whenever i post on blogger my app automaticall updated

Please suggest me I will wait for your answer

• Reply • Share >



mernid • 8 days ago



Android · 5 days ago

hi, It's really great job.

i have a question. how can i add a separator or a section to the drawer?
for example i have 5 items and i need a separator after third item.

^ | v · Reply · Share >



Arihant Jain · a month ago

hello, ur tutorial is really great. i want a help how to add icons to row item , i am unable to do it. pls help asap

^ | v · Reply · Share >



Rick → Arihant Jain · 16 days ago

Arihant Jain I'm not so expert with Android yet but if I'm not wrong in understanding your question, then you might need to do something like this:

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

so prepare your icons in drawable folder and link them there.
Hope this will help.

For more info, go to [@Ravi Tamada](#) <http://www.androidhive.info/20...> tutorial.. you will find details there..

Cheers!

^ | v · Reply · Share >



hamid reza · a month ago

thanks alot

^ | v · Reply · Share >



Shraddha · a month ago

hello, ur tutorial is really great. i want a help how to add icons to row item , m unable to do it. pls help asap

^ | v · Reply · Share >



Rick → Shraddha · 16 days ago

Shraddha I'm not so expert with Android yet but if I'm not wrong in understanding your question, then you might need to do something like this:

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

so prepare your icons in drawable folder and link them there.

Hope this will help.

For more info, go to [@Ravi Tamada](#) <http://www.androidhive.info/20...> tutorial.. you will find details there..

Cheers!

^ | v • Reply • Share >



amine • a month ago

thanks man, very nice job

^ | v • Reply • Share >



Ashwath • a month ago

whenever i click on home, it gives "Unfortunately, theapp has stopped".Why so?

^ | v • Reply • Share >



revat3260 • a month ago

Hello ,

I download my fragment list "tiles" from Firebase.
and make a string like this

```
titles = new String[5];
titles[0] = title0; (title0 is a variable containing a value of the database)
titles[1] = title1;
titles[2] = title2;
titles[3] = title3;
```

If I make my String array in [FragmentDrawer.java](#) - onCreate Void I take Fatal Exception error.

I would like to know how I could make this array in MainActivity and take the data in [FragmentDrawer.java](#).

Actually I would like to replace the following command

```
titles = getActivity().getResources().getStringArray(R.array.nav_drawer_labels);
```

and make the titles list dynamically.

Thanx

RE

^ | v • Reply • Share >



Yashwant Chawhan • a month ago

How to make navigation drawer right to left, any idea??

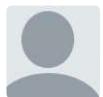
^ | v • Reply • Share >



Nijesh Narayanan ➔ [Yashwant Chawhan](#) • 23 days ago

just set navigation drawer alignment to left

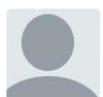
^ | v • Reply • Share >



eduardo · a month ago

Building "MaterialDesign" Gradle project info and building, building and building :)

^ | v · Reply · Share >



Roy Zack · a month ago

How to change the Title Text color i change the "textColorPrimary" but it still remain the same color white.... did the tutorial missing some variable for the theme?

^ | v · Reply · Share >



Jordan · a month ago

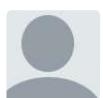
I'm on step 23 and I got this error :

Attempt to invoke virtual method 'void android.widget.TextView.setText(java.lang.CharSequence)' on a null object reference

at

mypackage.adapter.NavigationDrawerAdapter\$override.onBindViewHolder(NavigationDrawerAdapter.

^ | v · Reply · Share >



aman · a month ago

how to change text color of navigation item or background of that item when selected.

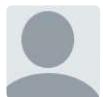
^ | v · Reply · Share >



vinod · a month ago

sir, can we write navigation drawer item's in telugu language????

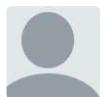
^ | v · Reply · Share >



aslam · 2 months ago

hi sir how to add icon in navigation drawer

^ | v · Reply · Share >



Manny264 · 2 months ago

Ravi, please help. How can I group the items on the nav drawer? I mean I already have a menu resource file with different groups. How and where do I set this group?

^ | v · Reply · Share >



Uyi Oriaghan · 2 months ago

Hello guys pls i need your help at step 20. After creating a class named

NavigationDrawerAdapter.java and pasted the codes i had this error:

Error:(16, 10) error: cannot find symbol class NavDrawerItem

Error:(20, 58) error: cannot find symbol class NavDrawerItem

Error:(33, 39) error: package R does not exist

Error:(40, 9) error: cannot find symbol class NavDrawerItem

Error:(54, 55) error: package R does not exist

Error:Execution failed for task ':app:compileDebugJava'.

> Compilation failed; see the compiler error output for details.

Information:BUILD FAILED

Please help me, this tutorial has been really interesting..

Thanks.

[^](#) [v](#) • Reply • Share >



fadejimi → Uyi Oriaghan • a month ago

At first you have to create the NavDrawerItem class and it should be in your model class, if it is there then you have to import the package name.

If you are using android studio just right click on the error(33,39) and import the package of your application like uyi.app.materialdesign depending on how you named your application.

[^](#) [v](#) • Reply • Share >



Uyi Oriaghan • 2 months ago

Hello guys pls i need your help at step 20.

after creating a class named **NavigationDrawerAdapter.java** and pasted the codes i had this error:

```
Error:(16, 10) error: cannot find symbol class NavDrawerItem
Error:(20, 58) error: cannot find symbol class NavDrawerItem
Error:(33, 39) error: package R does not exist
Error:(40, 9) error: cannot find symbol class NavDrawerItem
Error:(54, 55) error: package R does not exist
Error:Execution failed for task ':app:compileDebugJava'.
> Compilation failed; see the compiler error output for details.
Information:BUILD FAILED
```

Please help me, this tutorial has been really interesting..

Thanks.

[^](#) [v](#) • Reply • Share >



JosephAb → Uyi Oriaghan • 2 months ago

Do not copy and paste the entire code. NavDrawerItem must be in model subpackage inside your application main package file. So when importing that java class ,you have to give the correct path to it also. Eg: if you created the java class NavDrawerItem inside Model package then you must import as- import "yourpackagename".model.NavDrawerItem , Hope this helps :)

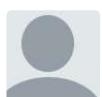
[^](#) [v](#) • Reply • Share >



Uyi Oriaghan → JosephAb • 2 months ago

Thanks so much, ive been able to fix it. but i also wanna ask the items in the recycleView in the NavDrawer i.e the HOME, FRIENDS ,MESSAGES, how do we make it highlight when its touched

[^](#) [v](#) • Reply • Share >



Abhin • 2 months ago

I got an error like "Resource not found colorPrimary"

"Resource not found colorPrimaryDark"

[^](#) [v](#) • Reply • Share >



JosephAb → Abhin • 2 months ago

check you colors.xml file inside values folder. inside that create a new resource as

```
<color name="colorPrimary">#F44336(hex code of color you want)</color>
```

Hope this helps :)

^ | v • Reply • Share >



Rick → Abhin • 2 months ago

same thing here... plus there are so many other things which is giving me errors like Version, minSdkRequired and others.. don't know how to resolve them.. what to do??

Suggestions needed.... Anybody?? or **@Ravi Tamada** ??

^ | v • Reply • Share >

sagarS → Rick • 2 months ago

try with older appcompat version:

```
compile 'com.android.support:appcompat-v7:22.2.1'
```

^ | v • Reply • Share >



sagarS • 2 months ago

Hello Ravi Sir,

This code is really helpful for me and very easy to understand.

I want to hide navigation drawer icon from some fragments or from main activity.

How to achieve it?

please guide me.

^ | v • Reply • Share >



Clay24g • 2 months ago

This is a tutorial any android developer should follow. Material design is amazing. Thanks, man!

^ | v • Reply • Share >



Chirag thaker • 2 months ago

Nice Tutorial about Nav.Drawer . One Side Question in android studio what "Invalid cache and restart does"

^ | v • Reply • Share >



Appkart In • 2 months ago

```
<fragment android:id="@+id/fragment_navigation_drawer"
    android:name="info.androidhive.materialdesign.activity.FragmentDrawer"
    android:layout_width="@dimen/nav_drawer_width" android:layout_height="match_parent"
    android:layout_gravity="start" app:layout="@layout/fragment_navigation_drawer"
    tools:layout="@layout/fragment_navigation_drawer"/>
```

Hi Ravi, Why android:layout_gravity="start" is used here. Please help because i did not understand

^ | v • Reply • Share >



Chirag thaker → Appkart In • 2 months ago

Because Navigation Drawer Appear from Left side, after Android L update they upgrade

Gravity : Left to Start and Right to End

^ | v • Reply • Share >



Indio John • 2 months ago

Very well explained guide on material design finally I found.

^ | v • Reply • Share >



Arihant Jain • 2 months ago

hello ravi sir ,

I implement above code in eclipse and android studio both but in both i get this error please help me how to resolve this issue

02-03 23:49:13.499: E/AndroidRuntime(1829): FATAL EXCEPTION: main

02-03 23:49:13.499: E/AndroidRuntime(1829): Process: com.example.materialdesign, PID: 1829

02-03 23:49:13.499: E/AndroidRuntime(1829): java.lang.RuntimeException: Unable to start activity ComponentInfo{com.example.materialdesign/com.example.materialdesign.MainActivity}: android.view.InflateException: Binary XML file line #35: Error inflating class fragment

02-03 23:49:13.499: E/AndroidRuntime(1829): at
android.app.ActivityThread.performLaunchActivity([ActivityThread.java:2184](#))

02-03 23:49:13.499: E/AndroidRuntime(1829): at
android.app.ActivityThread.handleLaunchActivity([ActivityThread.java:2233](#))

[02-03 23:49:13.499: E/AndroidRuntime\(1829\): at](#)

[see more](#)

^ | v • Reply • Share >



Arihant Jain ➔ Arihant Jain • 2 months ago

Thanks for the example it works fine in android studio

^ | v • Reply • Share >



Ravi Tamada Mod ➔ Arihant Jain • 2 months ago

Clean the project from Build -> Clean and try again.

^ | v • Reply • Share >



Cyril Ferrand • 3 months ago

Nice job, Thanks !

But...I try to share drawer in a base activiy Class but I have a strange error :

java.lang.IllegalArgumentException: No view found for id 0x7f0e006c

(com.facades_de_nice.fdnapplication:id/container_body) for fragment HomeFragment

^ | v • Reply • Share >



Ravi Tamada Mod ➔ Cyril Ferrand • 3 months ago

Clean the project and run again.

[^](#) [|](#) [v](#) • Reply • Share >**Cyril Ferrand** → Ravi Tamada • 3 months ago

Same :(

But I make code modification to have MainActivity extends BaseActivity, maybe a problem of lifecycle...My new MainActivity only contain the OnCreate method :

```
protected void onCreate(Bundle savedInstanceState) {  
  
    super.onCreate(savedInstanceState);  
  
    setContentView(R.layout.activity_main);  
  
    // display the first navigation drawer view on app launch  
  
    displayView(0);  
  
}
```

[^](#) [|](#) [v](#) • Reply • Share >**Cyril Ferrand** → Cyril Ferrand • 3 months ago

I find it !

I forget super.setContentView(layoutResID); in my override setContentView...
Shame on me[^](#) [|](#) [v](#) • Reply • Share >**sanju davariya** • 3 months ago

please more elaborate step no:-12

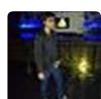
[^](#) [|](#) [v](#) • Reply • Share >**Khiem Nguyen Dang** → sanju davariya • 3 months ago

u go to in your project. Create manual folder menu when it will show on Android Studio

[^](#) [|](#) [v](#) • Reply • Share >**indrajit narvekar** • 3 months ago

Very first time I used this tutorial and made my Material design successfully...

Thanks alotwhoever made this

.....
thanks...[^](#) [|](#) [v](#) • Reply • Share >**Deepak Sharma** • 3 months ago

How Can I lock the drawer when on item click ?

[^](#) [|](#) [v](#) • Reply • Share >**ponco robbi** • 3 months ago

very helpful, thanks!

[^](#) [|](#) [v](#) • Reply • Share >



Osinnowo Emmanuel • 4 months ago

I don't how much I should thank you, this is the greatest tutorial ever seen, it is very comprehensive and concise compare to other tutorials I have been seeing, I came here out of frustration and I got it all, thank you once again, you're a great Teacher. Thanks.

^ | v • Reply • Share >



ARASHI Funsho → Osinnowo Emmanuel • 3 months ago

Yup..... trained me up to intermediate level in no time.....

^ | v • Reply • Share >

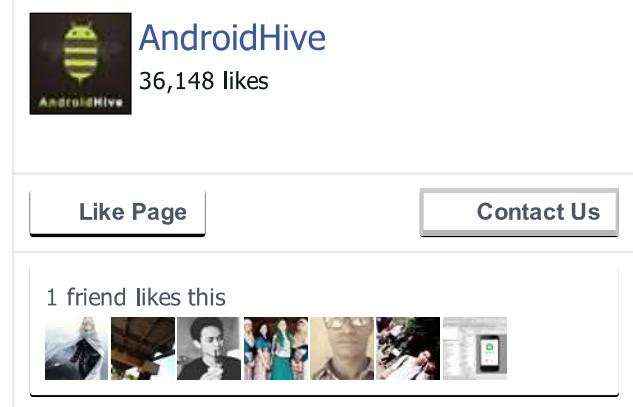
[Load more comments](#)

[Subscribe](#)

[Add Disqus to your site](#) [Add Disqus Add](#)

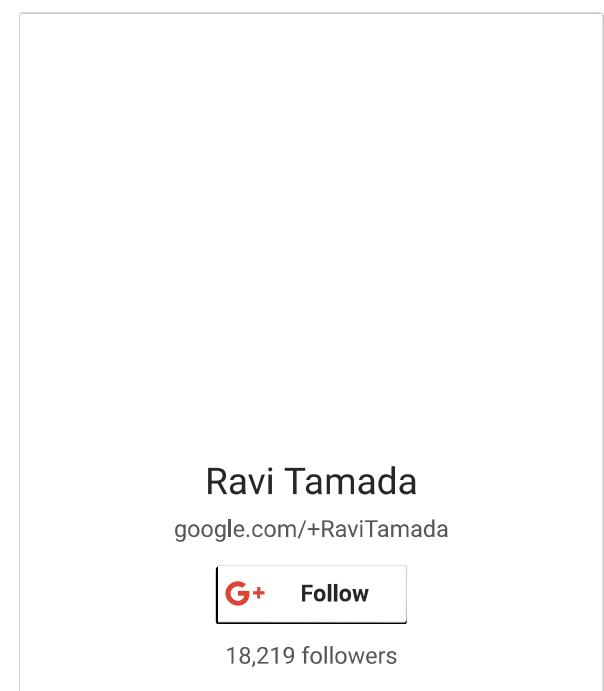
[Privacy](#)

Ravi Tamada
Hyderabad, INDIA



Tag Cloud

Action Bar Adapter AdMob Analytics
Animation API Apps Async
Beginner Camera Chat Dashboard
Database facebook File Upload
Fragments GCM GDK Gestures
Google Google Glass Google Plus
GPS Grid Grid View HTTP
Intermediate json Libstreaming
List View Locale Location Maps
Material Design MySQL
Navigation Drawer PayPal PHP Pinch
Progress Bar Push Notifications
Quick Tips RecyclerView REST
sessions Slim SMS Sockets
Speech Input Spinner sponsored
SQLite Swipe Tab View Twitter UI
Video Video Streaming View Pager
Volley Wearable xml YouTube



Most Popular

- 1 Android SQLite Database Tutorial - 1,363,127 views
- 2 How to connect Android with PHP, MySQL - 1,339,196 views
- 3 Android JSON Parsing Tutorial - 1,195,969 views
- 4 Android Push Notifications using Google Cloud Messaging (GCM), PHP and MySQL - 1,124,346 views
- 5 Android Sliding Menu using Navigation Drawer - 1,025,624 views
- 6 Android Custom ListView with Image and Text - 946,771 views
- 7 Android Login and Registration with PHP, MySQL and SQLite - 896,677 views

8 Android GPS, Location Manager Tutorial -

683,712 views

9 Android Tab Layout with Swipeable Views -

652,762 views

10 Android Tab Layout Tutorial - 579,269 views

Copyright © AndroidHive

[Advertise](#) . [Privacy Policy](#) . [Terms & Conditions](#)