

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free.

Take the 2-minute tour

Android 5.0 material design style navigation drawer for KitKat



I see that Android introduced new navigation drawer icons, drawer icon and back arrow icon. How can we use that in Kitkat supported apps. See Google's latest version of Newsstand app, which has the latest navigation drawer icons and animations. How can we implement that?

I have tried setting the minSDK to 19 and compileSDK to 21 but it's using the old style icons. Is that self implemented?

android navigation-drawer android-5.0-lollipop

edited Oct 29 '14 at 11:33



tomrozb

5,875 ● 13 ● 37 ● 59

asked Oct 20 '14 at 23:58

nomongo

724 ● 3 ● 14 ● 23

What theme is your activity using? – [alanv](#) Oct 21 '14 at 1:03

right now, holo. Want to switch to Material style and yet be compatible with kitkat – [nomongo](#) Oct 21 '14 at 3:00

Use the Appcompat theme and check the answer below – [Gabriele Mariotti](#) Oct 21 '14 at 6:02

Google keeps on changing the Drawer design in every app , the simple way they did in Google IO is simple and Overlay pattern [mobilewebwizard.in/2015/01/...](#) – [Tarun Chaudhary](#) Feb 14 at 3:18

3 Answers

You need to use the new Toolbar in the appcompat v21 and the new `ActionBarDrawerToggle` that is in this library as well.

Add the gradle dependency to your gradle file:

```
compile 'com.android.support:appcompat-v7:21.0.0'
```

Your `activity_main.xml` layout would look something like that:

```
<!--I use android:fitsSystemWindows because I am changing the color of the statusbar as well-->
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main_parent_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:fitsSystemWindows="true">

    <include layout="@layout/toolbar"/>

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

        <!-- Main layout -->
        <FrameLayout
            android:id="@+id/main_fragment_container"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

        <!-- Nav drawer -->
        <fragment
            android:id="@+id/fragment_drawer"
            android:name="com.example.packagename.DrawerFragment"
            android:layout_width="@dimen/drawer_width"
            android:layout_height="match_parent"
            android:layout_gravity="left|start" />
    </android.support.v4.widget.DrawerLayout>
</LinearLayout>
```

Your Toolbar layout would look something like that:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<android.support.v7.widget.Toolbar
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/toolbar"
    app:theme="@style/ThemeOverlay.AppCompat.ActionBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:minHeight="?attr/actionBarSize"
    android:background="?attr/colorPrimary"/>
```

Your activity must extend from:

`ActionBarActivity`

When you find your views (drawer and toolbar) in the activity the set the toolbar as the support action bar and set the `setDrawerListener`:

```
setSupportActionBar(mToolbar);
mDrawerToggle= new ActionBarDrawerToggle(this, mDrawerLayout,mToolbar, R.string.app_name,
R.string.app_name);
mDrawerLayout.setDrawerListener(mDrawerToggle);
```

After that you just need to take care of the menu items and drawerToggle state:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = new MenuInflater(this);
    inflater.inflate(R.menu.menu_main,menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (mDrawerToggle.onOptionsItemSelected(item)) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}

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

@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    mDrawerToggle.onConfigurationChanged(newConfig);
}

@Override
public void onBackPressed() {
    if(mDrawerLayout.isDrawerOpen(Gravity.START|Gravity.LEFT)){
        mDrawerLayout.closeDrawers();
        return;
    }
    super.onBackPressed();
}
```

The implementation is the same as It was before the Toolbar and you receive the arrow animation for free. No headaches. For more information follow:

- [The documentation.](#)
- [The ChrisBanes post](#)
- [The official android blog post.](#)

If you want to display the drawer over the Toolbar and under the status bar, please refer to [this question](#).

EDIT: Use `NavigationView` from the support design library. Tutorial to learn how to use in here: <http://antonioleiva.com/navigation-view/>

edited Jun 30 at 13:57

answered Oct 24 '14 at 15:35



[jpardogo](#)

3,263 ● 1 ● 9 ● 20

35 I have a question... what is the purpose of this awesome ArrowDrawable when Google recommends in Material Spec that Drawer should be above ActionBar/Toolbar? – [Michał Z.](#) Oct 24 '14 at 15:46

To be honest, I don't know. But you can find a good discussion about it in this google+ post: plus.google.com/102272971619910906878/posts/CQW5Wejx44g – [jpardogo](#) Oct 26 '14 at 15:17

1 @MichałZ. Actually there's a pretty easy explanation for this: "Ink on paper doesn't transfer itself between pieces of paper." [quote from Adam Powell \(click\)](#). And as you may know: Pieces of paper that are moving over each other is one of the core-concepts of Material Design. – [reVerse](#) Oct 31 '14 at 12:24

2 Google's Apps has different Navigation Drawers: Play store has this fancy arrowdrawable,newsstand has the material design which moves actionbar. androidpolice.com/2014/10/30/... – [Dory](#) Nov 3 '14 at 5:16

I followed your answer but it does not show any icon with or without animation – [Ravi](#) Nov 5 '14 at 9:29



There are plenty of libraries now that can achieve this.

Choice 1 - <https://github.com/neokree/MaterialNavigationDrawer>

Others

- <https://github.com/HeinrichReimer/material-drawer>
- <https://github.com/kanytu/android-material-drawer-template>
- <https://github.com/balysv/material-menu>
- <https://github.com/ikimuhendis/LDrawer>
- <https://github.com/Zlate87/material-navigation-drawer-example>

edited Dec 30 '14 at 6:58

 [Jonathan727](#)
460 ● 3 ● 12

answered Dec 27 '14 at 6:20

 [SSR](#)
7,069 ● 2 ● 14 ● 39

I would strongly advise Reimer's project first, and not neokree's. The former correctly puts the navigation drawer behind the task bar, includes all the necessary source and is able to build the project, which the latter fails on all accounts. Though Reimer's is currently not working on 5.1 and still contains bugs. – [RedGlyph](#) Apr 24 at 11:59

- 1 Scratch that, Reimer's code isn't mature enough and probably wants to tackle too many things that still need debugging, which doesn't make it a didactic example. github.com/kanytu/android-material-drawer-template is a better example but it's not maintained anymore, in favour of his latest version (github.com/kanytu/Android-studio-material-template), which is actually a template for Android Studio! Much better to start from there IMHO. – [RedGlyph](#) Apr 24 at 13:42

first choice is questionable — having to extend a class other than ActionBarActivity is PITA, because then you can't have a common base class for activities – [Sarge Borsch](#) May 16 at 20:01

If you want the real navigation drawer with material design style (defined [here](#))

I have implemented a custom library that do exactly that.

You can find it [here](#)

answered Nov 18 '14 at 23:25

 [NeoKree](#)
202 ● 1 ● 8

can i do it eclipse rather than android studio..? because i need to this navigation drawer – [gvsharma](#) Jan 21 at 14:14

Sure, but you have to import manually all dependencies and codes. – [NeoKree](#) Jan 22 at 19:53

It seems to have quite a few issues, have you tested the build on a fresh clone? From where I stand, the project directories being on several levels, you need at least to add this to your example settings.gradle:

```
project(':MaterialNavigationDrawerModule').projectDir = new File('..../MaterialNavigationDrawerModule')
```

 Then it tries to import a `import it.neokree.example.backpattern.BackPatternCustom;` that doesn't exist. Still 93 open issues. It's nice to provide a library ... if it's solid. – [RedGlyph](#) Apr 24 at 9:18

protected by [Community](#) ♦ May 18 at 21:22

Thank you for your interest in this question. Because it has attracted low-quality answers, posting an answer now requires 10 [reputation](#) on this site.

Would you like to answer one of these [unanswered questions](#) instead?