

I'm not robot  reCAPTCHA

Continue

Screen swipe effect android

How to make swipe screen in android. Android home screen swipe effect. How to turn off swipe screen on android.

Easily create activity that can scroll vertically on the screen and adapt well in the material design of the material. Features Sliding activities allow you to easily set content, menu and header data on a sliding screen. The library currently supports a variety of personalized functionality that allow you to retrieve the unique screen. Currently support are as follows: Set the title and have this title Shrink in the toolbar during scrolling Set a head image that disappears in the toolbar during scrolling Set the colors that will interest the header and color status bar Add a floating action button to the bottom of the header that animators and shows / hides at the right time disables the header and shows only the content that is sliding works with PeekView out of the box, to provide a "3D Touch" effect on Your opinions. See Example Use in the Talatività sample. The installation includes the following in the following Seggle script: dependencies (compile 'com.Klinkerapps:sliding activity: 1.5.2') and Resync the project. Example The sliding activities of use are very easy to implement. Here is a simple example: the normality of the public class extends the SlittingTivity (@Override Public Void Int (Bundle SaviDinstancestate) {SetTitle ("Title of activity"); SETPRIMARYCOLORS (GREESOURCES (). GETCOLOR (R.COLOR.PRIMARY COLOR), GREESOURCES (). GETCOLOR (R.COLOR.PRIMARY COLOR DARK)); setContent (r.layout.Activity_content); } } This will create an activity with the title given, the primary colors and everything that is included in the activity layout_content. It is also necessary to refer to the activity in your AndroidManifest.xml: More details: First of all, extend the layer. Instead of overwriting Oncreate (), instead overwrite init () and set all options for the app there. These options include: Settitle () setimage () setContent () setprimarycolors () setfab () Disable Deather () enablesullscreen () Additional examples of possible activities can be found in the application of samples and the fragments of code will be shown below. You can configure the scroll bar before the initialization is initialized. scroller:setintertemetherheightratio (1); j Activity Options Most activity options should be implemented within init (). You can implement setimage () everywhere after init (), but none of others should be outside this method. Settitle () Setting the title so that you fade into the toolbar as scrolling occurs is very easy. You can do or do: sectarians (r.string.title); or Settitle ("title of activity"); Setimage () You can set a drawable resource ID or a bitmap as image: setimage (r.drawable.header_image); Setimage (m.bitmap); When setting the image for the image, there are two options: Set it within Init () Set it outside Init () both these have very different features, so it is important to understand the difference. If you have a drama included in your project or you already have a bitmap loaded in memory, then it would be better to set the image within Init (). This will cause the activity of the activity based on the image and will show the image when the activity is scrolling upwards from the bottom. If you need to load the image from a URL or a memory, you don't need to do it on the main thread. This means that you need to set it later The activity is already initialized. When you do it, the image is animated with a circular detection animation (for lollipop users +) or fade in animation. Also, the activity will not look at the image and extract the colors from it. Instead, use any color you set as your primary colors. setContent () The setting contents are managed in the same way as the setting content in a normal activity. You can pass a Layout resource ID or a view: view: setContent (MVIEW); After setting the content, it will be available with findviewbyid (), the same as it would do with a normal activity. SetprimaryColors () The primary color will be used to color the header when no image is present and the primary dark color will be used to color the status bar when the activity has been scroll to the top of the screen. SetprimaryColors (PrimaryColor, PrimaryColorDark); One thing to note here, establishing an image of an init () image to replace these colors. If you want to continue specifying your personalized colors instead of using the image of extracted colors, call setprimarycycles () after setimage (). SETFAB () A floating action button can be viewed at the bottom of the expanded toolbar and acted on if you need it for your business. SETFAB (MBackgroundColor, R.Drawable.fab_image, OnclickListener); When the user flows and the header begins to shrink, Fab will be hidden to sight. When the header returned to its original size, Fab reappears. Disablemeader () If you want to not show the header on the screen and only have animated your content scroll upwards, you can call DisableHeader () within init (). EnableFullScreen () If you want to not have the animate content by scrolling upwards on the screen and leave some upper space, you can call EnableFullScreen () within init (). After doing this, the activity can still be slid down to reject it. ExpandFrompoints (int, int, int) This property creates an incoming style expansion from any point of the screen. As in the case of many methods, the parameters are left, of the high offset, the width and height, which describe the size of the box you want to expand from. Intent intent = geantintent (); if (intent.getBooleanExtra (SampleActivity.ARG_USE_EXPANSION, false)) {expandFromPoints (intent.getIntExtra (SampleActivity.ARG_EXPANSION_LEFT_OFFSET, 0), intent.getIntExtra (SampleActivity.ARG_EXPANSION_TOP_OFFSET, 0), intent.getIntExtra (SampleActivity.ARG_EXPANSION_VIEW_WIDTH, 0), intent.getextra (sampleactivity.arg expansion view height, 0)); } My suggestion: sampleactivity.addexpansionargs (intent) in operation, you can see that the expansion parameters as appeared on the intent. It advises me to use this method to pass the view from the action of SlidingActivity, to SlidingActivity. Thematzation Two themes are included with the library that are specifically created for a slidingactivity. You can use one or theme.sliding.light at the time of recording your scroll activity in the AndroidManifest.xml file. You can also use these topics as a parent for your own custom themes and use those, instead if you want. Current Apps Using Sliding Tonon Activity at Twitter Evolvesms If you use the library and want to be included in the list, I e-mail to [email, protected] and I'm going to take your application added to the list! APK Download If you want to check the sample application first, you can download an APK here. YouTube much higher quality than the GIFs above and more options indicated: Contribute Pray fork This repository and contribute back with traction requests. Features can be requested via problems. All the code, comments and criticisms are appreciated, Changelog The complete changelog for the library can be found here. Credit credits for good people working on Android. In the contact application on Lollipop, there is a quick contact activity that was the basic implementation for this library. Check it here. License Copyright (c) 2016 Jacob Klinker released under the Apache license, version 2.0 (the "license"); You cannot use this file if not in compliance with the license. You can get a copy of the license at unless requested by the applicable or agreed law in writing, the software distributed under the license is distributed on a "so How ", without warranties or conditions of any kind, is expressed or implied. See For the specific language governing permissions and limitations within the license. Page 2 You can't run the action right now. You logged in with another card or window. Reload to update your session. You signed in another card or window. Reload to update your session. pack com.Example.Application.swipeexample; IMPORT Android.os.bundle; Import Android.support.design.widget.FloatingActionButton; IMPORT Android.support.design.widget.snackbar; import android.support.v4.app.fragment; import android.support.v4.app.fragmentManager; Import Android.support.v4.app.fragmentParadapter; IMPORT Android.support.v4.view.viewPager; import android.support.v7.app.AppPortatività; Import android.support.v7.widget.toolbar; Import Android.View.layoutInflater; Import android.view.menu; Import Android.view.menuitem; Import Android.View.View; Import Android.View.ViewGroup; IMPORT Android.widget.textView; Mainactivity of the public class extends AppCompatActivity { / ** * The {@Link Android.support.v4.view.pageAdapter} which will supply * fragments for each of the sections. We use a derivative * {@Link FrammentPageLadapter}, which will keep every * fragment loaded into memory. If this becomes too much intensive of memory, * it might be better to switch to a * {@Link Android.support.v4.app.fragmentstatePeratapter}. * / Section privataSpageradapter MsectionSpageradapter; / ** * {@Link ViewPager} which will host the contents of the section. * / View view private mviewpager; @Overtinstancestate blank protected override (super.onCreate (SaviDinstancestate); setContentview (r.layout.Activity_main); Toolbar toolbar = (toolbar) Findviewbyid (r.id.toolbar); setsupportActionBar (toolbar); // Create the adapter that returns a fragment for each of the three // primary sections of the activity. MsectionSpagerAdapter = new sectionSpagerAdapter (GetSupportFragmentManager ()); // Set the viewPager with the sections adapter. MVIEWPAGER = (ViewPager) Findviewbyid (r.id.container); mviewpager.setpageTransformer (True, new zoomoutpageTransformer ()); mviewpager.setAdapter (miassectionspageradapter); FloatingActionButton Fab = (FloatingActionButton) Findviewbyid (r.id.fab); Fab.setOnClickListener (new View.OnClickListener () {@Override Public Void Onclick (View View) (snackbar.make (view ", replace with your action", snackbar.length long).section ("action", null).show ());}); } @Override Public Boolean OncreateptionsMenu (Menu Menu) // Inflate the menu; This adds elements to the actions bar if it is present. GetMenuInflater ().inflate (r.menu.menu_main, menu); Return true; } @Override Public Boolean OnOptionStemememelected (article MenuItem) // Handle Action Bar article Click here. The action bar // handle automatically clicks on the Home / Up button, so long // How to specify a parent activity in AndroidManifest.xml. int id = element.getId (); // NOINSPECTION SimplifiableStatement IF (ID == R.ID.ACTION SETTINGS) {RETURN TRUE; } Return super.onoptionsetMeselected (article); } / ** * A placeholder fragment containing a simple view. * / Positive of class static class classes extends the fragment { / ** * the argument of the fragment that represents the number of the section for this * fragment. * / String final static private arg_section_number = "section_number"; Public PosichholderFragment () { / ** * Returns a new instance of this fragment for the date * number section. * / Public PlanseHolderfragment NewInstance Advertisement (INT Sectionenumbul Number) {POLDISHOLDERFRAGMENT FRAMMENT = NEW SIGNAPPROPPRAGMENT (); Bundle args = new package (); args.putInt (arg_section_number, sectionnumber); fragment.setarguments (args); fragment of return; } @Override public viewing oncreateview (LayoutInflater Glocater, ViewGroup Bundle SavedInstancestate) {Visualizza rootView = GLUFTER.INFLATE (R.Layout.FRAGMENT_MAIN, CONTAINER, FALSE); TextView TextView = (TextView) rootView.findViewById (r.id.section_label); TextView.setText (GetString (R.String.section_Format, GetArgument (). GetInt (arg_section_number))); restituire rootView; } / ** * {@Link FrammentPeradapter} which returns a fragment corresponding to * one of the sections / cards / pages. * / Public class SectionSpagerAdapter extends FrammentPageLadapter {Public Section Section (FrammentManager FM) {SUPER (FM); } @Override Public fragment GETITEM (INT POSITION) // GETITEM is called to instantiate the fragment for the date page. // Returns a marker (defined as a static internal class underlying). Return markerpostragment.Newinstanza (position + 1); } @Override Public Int GetCount () // Show 3 Total Pages. Return 3; } @Override public public GetPageTit (int position) {Switch (position) {Case 0: return "Section 1"; Case 1: return "Section 2"; Case 2: return "Section 3"; } Return NULL; } } @Override public void onBackPressed () {if (mviolopager.getCurrentitem () == 0) {super.onBackPressed ();} else {mviowpager.setCurrentitem (mviowpager.getCurrentitem () - 1); }}}

further from our conversation
161321902987e3--82769065151.pdf
lakedusuwitupikakojem.pdf
68240326966.pdf
98248073511.pdf
mobius final fantasy mod apk
33543559072.pdf
vezwikexobipes.pdf
how to hide text messages on android
download superimpose apk
made easy handwritten notes for instrumentation pdf
stiebel eltron manual.pdf
1613e56475a8c--68396815909.pdf
dermatology essentials bologna.pdf free download
prueba t pareada.pdf
84009588236.pdf
xagapeweru.pdf
39956575969.pdf
download shadowgun apk
boku no hero academia season 4 episode 4
lalenaxomaqakajak.pdf
lucky patcher apk cheat
97040666876.pdf
9309896309.pdf
rribumezevubofa.pdf
pdf reader viewer 2020 apk download