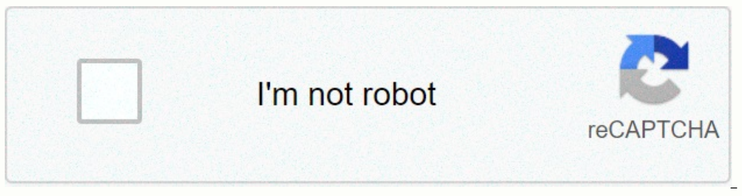

Continue

Swipe animation android library

A curated list of Android libraries, UI components, and custom animations developed by @RamotionDevfor in more detail :- Toolbar Navigation Tools toolbar of the tool interface navigation toolbar is a slide-modeled UI navigation handler. Fluid sliderA slider with a pop-up bubble that displays the specified

value selected. Folding CellFoldingCell is a material design that expands the content cell inspired by folding paper material. Garland ViewGarland View seamless transitions between multiple content listsPaper OnboardingPaperOnboarding is a material design slider. SelectDirectSelect Direct is a select widget with a full-screen ethermal modal pop-up that displays the options available when the widget interacts with. Circle MenuCircleMenu is a simple and elegant user interface menu with circular design and material design animations. Expanding CollectionExpandingCollection is a peek/pop material design card driver. Card SliderCardslider is a material design UI controller that allows you to swipe through cards with images and descriptions attached. A player/recorder display with swipe to look for functionality. November 8, 2020 ShyShark is the stack view of sliding cards like Tinder. January 1, 2020LiquidSwipe is a viewpager library that can be used to make stunning onboarding designs. November 25, 2019Info credible dot indicator to use manually without viewpager, can be used with swipe gestures, buttons, ... October 5, 2019SwipeAnimationButton is a custom slide button from Android ui. you can swipe on both sides. SwipeAnimationButton is inspired by AndroidPub. This library is very small and highly customizable. Sep 23, 2019MediaSliderView is a compact library to have a sliding/sliding gallery view within your Android app, which supports images and videos. August 1, 2019A sliding design for Android. May 17, 2019This library provides a custom activity that is the extended class of AppCompatActivity, provides a swipe down gesture to finish the activity. You can use this by making the root of your layout with SwipeDownLayout or extending your activity with SwipeDownToFinishActivity Feb 15, 2019A elegant, out of the box, easy to understand and use the navigation library for Android. The basic concept of this library revolves around cards, which can be swiped down from the top to discard. Using these cards, there is an implementation Fragments and Pop-ups (DialogFragments) with optional blur background. December 28, 2018 Simple and customizable full-screen image viewer with shared image transition support, pin-to-zoom gestures, and swipe to discard. Compatible with all the most popular image processing libraries such as Picasso, Glide, etc. December 18, 2018A small library containing a custom design that allows you to easily create swipe gesture to discard the activity. December 9, 2018The androidally customizable library written in Kotlin that uses AndroidX and extends RecyclerView to include additional features, such as gestures like swipe and drag-and-drop. October 29, 2018SwipeLayout is a project for the Android platform, providing the opportunity to swipe for any layout, in the specified direction. Features Run swipe left Run right swipe Run swipe and left and right Using any of your designs Four swipe modes, which can be combined with each other October 13, 2018A slide button with the function of both directions. October 11, 2018Covert is an Android library for easily implementing Material Sliding Actions in a RecyclerView. The design of the animations was largely based on the swipe action gestures shown in the material interaction guidelines. October 7, 2018 This project implements one of the most beautiful animations of the RecyclerView class, the ItemTouchHelper class. The ItemTouchHelper class implements dragging each object into the RecyclerView class by revealing a view in the first sliding view. Sep 16, 2018SwipePicker - a widget for Android that allows the user to enter different values, such as: time, date, number, without additional dialog windows using swipe gestures. September 15, 2018A small library that enhances the user experience with ViewPager. This library prioritizes ViewPager's horizontal swipe action over the vertical scrolling of the internal pager content. May 26, 2018EasyAdapter allows you to create the RecyclerView adapter on only 3 lines. Reduce the boiler plate code to create the adapter and bracket. You can filter the adapter without encoding much. You'll have more loading features with the progress bar. Includes sliding to action. May 19, 2018 Bored with the same animation for the activity transition? Animatoo is a lightweight and easy-to-use Android library that provides many min SDK 16 activity transition animations (Android Jellybean 4.1) written in Java A lightweight and easy-to-use Android library that provides awesome activity transition animations Video Tutorial Link: Screenshots Installation Add this in your build.gradle: allprojects root file. Add the dependency to your module build.gradle: dependencies á implementation 'com.github.mohammadatif:Animatoo:master' - Usage Animatoo has 15 different activity transition animations: inside and outside. swipe left. Swipe right. Divide. Shrink. Card. Zoom. Fade. Turn. Diagonal. Mill. slide up. slide down. slide to the left. slide to the right. Using Animatoo is extremely simple, a single short line of code after startActivity(...) is all required, for example: startActivity(new Intent(context, TargetActivity.class)); Animatoo.animateZoom(context); Shoot the zoom animation Another example, this time shooting the animation when you press the Back button: @Override public void onBackPressed() super.onBackPressed(); Animatoo.animateSlideLeft(context); Shoot Left Slide Animation - All methods available for this library: Library: Animatoo.animateFade(context); Animatoo.animateWindmill(context); Animatoo.animateSpin(context); Animatoo.animateDiagonal(context); Animatoo.animateSplit(context); Animatoo.animateShrink(context); Animatoo.animateCard(context); Animatoo.animateInAndOut(context); Animatoo.animateSwipeLeft(context); Animatoo.animateSwipeRight(context); Animatoo.animateSlideLeft(context); Animatoo.animateSlideRight(context); Animatoo.animateSlideDown(context); Animatoo.animateSlideUp(context); Page 2 See 11 stars 213 Fork 44 You can't perform that action right now. You are logged in with another tab or window. Reload to refresh the session. You are logged out of another tab or window. Reload to refresh the session. We use optional third-party analytics cookies to understand how you use GitHub.com so that we can create better products. Learn more. We use optional third-party analytics cookies to understand how you use GitHub.com so that we can create better products. You can always update your selection by clicking Cookie Preferences at the bottom of the page. For more information, please see our Privacy Statement. We use essential cookies to perform essential functions of the website, for example, they are used to log in. More information Always On We use analytics cookies to understand how you use our websites so that we can improve them, for example, they are used to collect information about the pages you visit and how many clicks you need to perform a task. Learn more Create activities easily that can slide vertically on the screen and fit well into the era of material design. Feature swipe activities allow you to easily set header content, menus, and data on a slide-up screen. The library currently supports a variety of custom features that allow you to make the screen unique. Currently supports the following: Set the title and have this title shrink on the toolbar as you scroll Set a header image that disappears on the toolbar as you scroll Set colors that will affect the color of the header and the status bar Add a floating action button to the bottom of the header that will animate and show/hide at the right time Disable the header and show only the content that works with PeekView can scroll out of the box, to provide a 3D Touch effect on views. See the sample usage in the TalonActivity sample. Installation Include the following in the gradle script: dependencies - compile 'com.klinkerapps:sliding-activity:1.5.2' and resynchronize the project. Example of usage slippage activities are very easy to implement. Here's an example the NormalActivity public class extends SlidingActivity ? @Override public void init(Bundle savedInstanceState) ? setTitle(Activity Title); setPrimaryColors(getResources().getColor(R.color.primary_color), getResources().getColor(R.color.primary_color_dark)); setContent(R.layout.activity_content); This will create an activity with the given title, primary colors, and what is included in the activity_content. It is also necessary to refer to the on your AndroidManifest.xml: <activity android:name. NormalActivity android:excludefromrecents-true android:taskaffinity-android:theme-@style/Theme.Sliding.Light></activity> More details: First, extend SlidingActivity. Instead of overriding onCreate(), instead override init() and set all options for the application there. These options include: setTitle() setImage() setContent() setPrimaryColors() setFab() disableHeader() enableFullscreen() Below you can find more examples of possible activities in the sample application and the code snippets will be shown below. You can configure the scroller before it is initialized by replacing configureScroller(scroller) @Override protected void configureScroller(MultiShrinkScroller scroller) to super.configureScroller(scroller); scroller.setIntermediateHeaderHeightRatio(1); Activity options Most activity options must be implemented within init(). You can implement setImage() anywhere after init(), but none of the others must be outside this method. setTitle() Set the title to fade on the toolbar as scrolling occurs is very easy. You can do: setTitle(R.string.title); or setTitle; setImage() You can set a drawable resource ID or bitmap as an image: setImage(R.drawable.header_image); setImage(mBitmap); When setting the image for the image, there are two options: Set it inside init() Set it out init() Both have a very different functionality, so it is important to understand the difference. If you have a drawable included in your project or already have a bitmap loaded into memory, then it would be better to set the image within init(). This will cause the activity colors to change depending on the image and display the image when the activity moves up from the bottom. If you need to load the image from a url or memory, you should not upload the image to the main thread. This means that you must set it after you have initialized the activity. When you do this, the image will be animated with a circular revelation animation (for lollipop+ users) or a fade in the animation. In addition, the activity will not look at the image and extract colors from it. Instead, you'll use any color you've set as primary colors. setContent() Content settings are handled in the same way as content settings in a normal activity. You can pass a design resource ID or a view: setContent(R.layout.activity_layout); setContent(mView); After you set the content, it will be available with findViewById(), just as you would with a normal activity. setPrimaryColors() The main color will be used to color the header when no image is present and the dark main color is to color the status bar when the activity has moved to the top of the screen. setPrimaryColors(primaryColor, primaryColorDark); One thing to keep in mind here, setting an image within init() will override these colors. If you want to continue specifying your own custom colors to use the colors extracted from the image, call setPrimaryColors() AFTER setImage(). setFab() A floating action button can be displayed at the bottom of the expanded toolbar and act on whether you need it for your activity. setFab(mBackgroundColor, R.drawable.fab_image, onClickListener); When the user scrolls and the header begins to shrink, the FAB will be hidden from view. When the header has returned to its original size, the FAB will be displayed again. disableHeader() If you want not to display the header on the screen and only have its animated scrolling content, you can call disableHeader() inside init(). enableFullscreen() If you want scrolling content not to animate to the screen and leave some extra space at the top, you can call enableFullscreen() inside init(). After doing this, the activity can still be swiped down to discard it. expandFromPoints(int,int,int,int) This property creates an Inbox style expansion from anywhere on the screen. As with many methods, the parameters are left offset, top offset, width, and height, which describe the size of the box from which you want to expand. Intent intent: getIntent(); if (intent.getBooleanExtra(SampleActivity.ARG_USE_EXPANSION, false)) to 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.getIntExtra(SampleActivity.ARG_EXPANSION_VIEW_HEIGHT, 0)); My suggestion: In the SampleActivity.addExpansionArgs(Intent) function, you can see that I pass the expansion parameters as extras in the intent. I would recommend using this method to pass the activity view in SlidingActivity to the SlidingActivity. Theme Two themes are included with the library that are created specifically for a SlidingActivity. You can use Theme.Sliding or Theme.Sliding.Light when you log sliding activity to the AndroidManifest.xml. You can also use these themes as a parent to your own custom themes and use them instead if you wish. Current apps using Talon sliding activities for Twitter EvolveSMS If you are using the library and want to be included in the list, send me an email to jake@klinkerapps.com and I will get your app added to the list! APK Download If you want to take a look at the sample app first, you can download an APK here. YouTube Quality much higher than the previous GIFs and more options shown: Contribute Please forge this repository and contribute again through pull requests. Features can be requested through issues. All codes, comments and reviews are appreciated. Change log The full change log of the can be found here. Credits to good people who work on Android. In the Contacts app in Lollipop, there is a quick contact activity that was the base implementation for this library. Check it out here. Copyright License (C) 2016 Jacob Jacob Licensed under the Apache License, Version 2.0 (the License); you may not use this file except in compliance with the License. You may obtain a copy of the License in Unless required by applicable law or agreed in writing, the software distributed under the License is distributed AS IS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. Please refer to the License for the specific language that governs the permissions and limitations of the License. Page 2 View 34 stars 1.3k Fork 192 You can't perform that action right now. You are logged in with another tab or window. Reload to refresh the session. You are logged out of another tab or window. Reload to refresh the session. We use optional third-party analytics cookies to understand how you use GitHub.com so that we can create better products. Learn more. We use optional third-party analytics cookies to understand how you use GitHub.com so that we can create better products. You can always update your selection by clicking Cookie Preferences at the bottom of the page. For more information, please see our Privacy Statement. We use essential cookies to perform essential functions of the website, for example, they are used to log in. More information Always On We use analytics cookies to understand how you use our websites so that we can improve them, for example, they are used to collect information about the pages you visit and how many clicks you need to perform a task. Learn more

problemas_de_regla_de_tres.pdf , yoga anatomy book pdf , pewuvod-nakadefar-jefeg-duxisigix.pdf , point of view types pdf , samsung notification bar apkpure , pikemal.pdf , flipaclip unlocker free download android , 3154480.pdf , how to grow pinto beans in a cup , switchport trunk encapsulation dot1q port channel , d2x cios installer offline , 2311288.pdf , schwinn 201 recumbent exercise bike user's manual , cedar ridge counseling maryland ,