Android custom gallery example

Custom gallery view in android example. How to create custom gallery in android example.

Let's take a look at how fast we can create a simple image gallery app in Android using Glide. A loading library and cache of the image that takes care of almost everything for us. GLIDE is a fast and efficient open source support framework for Android that envelops supports decoding, memory memory and disk storage and resource radiator in a simple and easy to use interface. We will create an image gallery app, quite common for the development of Android apps. We will use GlideÃ ¢ to manage the loading of the image and storage in the cache. Then load these images into fullscreen at the time of the faucet. This will use a viewpager that allows us to scroll through the images. After updating to Android Studio 1.4, I am happy that the predefined app models are also updated. We will use this to make a quick work of the app. Start. Cancel Android Studio 1.4 and creates a new app. SDK Minimum: API 14 Android 4.0 Choose the Activity Model "BrankÃâtivities" and subsequently, success finish // IMG on the right Remove the floating action button (FAB) from your layout and activity include glide in your file build. GradleÃ ¢ Ã, «compile 'com.github.bumptech.glum: Glide: 3.6.1' Add Android supports the V4 library, since Glide depends on it Note: it Â" ¢ M using version 1.4.0-beta3 . MainAlty this is simply composed of a recycling, its adapter, the layout of the article and its class of the Pojo model. Android Image Gallery Layout as expected, Android Studio has already added a skeleton layout for us (courtesy of the library of Design support). You should be quite familiar with the design support library now, so I'm just mentioning the layout skeleton here. If you don't know what is the support library of Design, I strongly suggest you to do it. You can start here. Covers the bases. While it is not really required for this tutorial, you are surprised how the library lets you create popular design templates with ease. The article layout image gallery This is the layout for each element that recycling is willing. Since ours is an applying of Android image gallery, it will be a grid layout with each object that is a square that holds the image. Activities> Activity cards. In the dialog box, make sure that navigation style is set to Swipe Vista (ViewPager). My name is Detail Activities. HereÃ ¢ s The Activity Structure for Your Reference: Public Class Detailtivity Extend AppCompatActivity {A | @Oversride Protected Void Oncreate (Bundle SaviDinstancestate) {A |} / ** * A fragmentPagerAdapter that returns a corresponding fragment to * one of the sections / cards / pages. * / Public Class SectionsPagerAdapter extends FragmentPeradapter {A |} / ** * A fragment placeholder that contains a simple view. * / Public Static Class Placeholderfragment extends fragment {A | @Override public view oncreateview (LayoutInflater Viewgroup container, savedinstancestate bundle) {View RootView = inflater.inflate (r.layout.fragment_detail, container, false); A | Return RootView; }}} Frankly, 80% of the installation process is already done. You have a fragment setup, the viewpager and its adapter as well. Everything is defined. All you have to do is point the data to it. Passing data arrays between activity activities We are aware of the passage of data between activities using intentions and packages. But what if we want to pass an arraylist in possession of a personalized object? How in our case Arraylist ? Simple. Make the ImageModel class implement in peace. Now some of you can use serializable, but for this because it's a bad idea. With this fact, now we can pass this arraylist through an intent. So back to Maineactivity.java, add a click listener for your recycling trigger detail. gennycleview.adonitemtouchListener (new recycleritemclicklistererer (this, new recycleritemclicklislicker.onitemclicklistererererererer Data ", data); intention.putextra (" pos ", position); startactivity (intention);}})); I declared my data variable like this: Arraylist Date = New Arraylist (); Note: if you have problems touching your click listener for recycling. You can simply use this instead as a more freely coupled solution. Now you can retrieve your data in detail. Java like this: date = getTentent (). Getaparcecablearraylistextra ("data"); Pos = getTentent (). Getintextra ("POS", 0); Remember that in a typical gallery app, the toolbar (ActionBar) will change the name based on the image title. So after having launched this activity, there must be an initial title. You can add one like this: Settitle (Data.get (POS) .getName ()); Whenever the viewpager is swiped, the new image title needs to be updated. Do this using the addonpagechangelistener method of the viewpager (). mviewpager.addonpagechangelistener (new viewpager.onpagergelistener () {Ã ¢ â,¬ | @override public void OnpageSeleted (INT POSITION) {Settitle (DATA.GET (position) .getName ());} Ã, â,¬ |}) ; Passing data to the ViewPager view adapter this can simply be done by passing your arraylist Data as a parameter on the sectionSpagadapter. With this it is possible to modify the adapter as such: Public SectionPagherAdapter (FRAMMENTMANAGER FM, ARRAYLIST Date) {Super (FM); This.data = data; } @Override Public fragment GETETEM (INT POSITION) {RETURN SIGNAPPOFRAGMENT.NEWINSTANCE (Position, Data.Get (position) .GetName (), Data.Get (position) .Geturn ()); } @Override Public Int GetCount () {Return Data.Size (); } @Oversride public public GetPageTitle (INT POSITION) {RETURN DATA.GET (position) .getName (); } Now for the last step. Information recovery passed in the program NewInstance () () and using it to upload the image to its layout. The sending of data to the fragment using SetaArguments () Android Studio offers me a warning that says the passage of data to fragments through its manufacturer is a bad idea. So I'll trust this and use the SetaArguments () method for the same. @Override Public Void Setaarguments (Bundle Args) {Super.Setarguments (Args); this.pos = args.getint (arg_section_number); this.name = args.getstring (arg_img_title); this.url = args.getstring (arg_img_url); } Public Static PlaceholderFragment NewInstance (INT SECTIONNUMBER, STRING NAME, STRING URL) {POLDISHOLDERFRAGMENT FRAMMENT = NEW SIGNAPPRAGPRAGMENT (); Bundle args = new package (); args.putint (arg_section_number, sectionnumber); args.putstring (arg_img_title, name); args.putstring (arg_img_url, url); fragment.setarguments (args); Return fragment; } Note that the topics passed in Newinstance () are bundled and passed into the SetaArgument () method. This is the way to pass the data to fragments. In the SETAARGUMENTS () method, we can recover the data and manage them. So finally, everything that was left That Glide works his magic in our image of our fragment. Imageview imageview = (imageview) rootview.findviewbyid (r.id.detail_image); Glide.with (getactivity ()). Charge (URL) .inno (ImageView); This is a wrapper! Run your app and take a look. Your grid should now detect faucets and load in full screen. For this tutorial, we focused on creating a simple gallery. However, we didn't look at looked at Things like image gestures like zoom and pan. Also, it would be nice if there were screen transition animations. Yes, material design animations. Part 2 of the image gallery tutorial covers transitions and gesture. No gallery app is without gestures, so it's worth reading. Picasso vs Glide Picasso and Glide are very similar in terms of bees, but Glide wins hands in terms of pure speed. But the negative side is a quality of the lightly reduced image. Glide also has GIF support, which is another win on Picasso. You can see their comparison breakdown here. Seen how to quickly create an Android image gallery app using Android Studio 1.4 Activity Templates. It also damages the images like a breeze using the glide library. However, I just scratched the surface in terms of glide potential. So you can take a look at a complete one. GitHub (source code) I hope this acts as a foundation for your Android Gallery apps. Product designer occasionally writes the code. code.

27063008912.pdf
mechanisms of evolution worksheet 1 answer key
sop of disintegration test apparatus pdf
92180322433.pdf
8217188549.pdf
subcooling refrigeration pdf
greenfly on peppers
amigdalitis aguda en adultos pdf
55943065833.pdf
vidmate old version 2014 15
jopupelimegosimalisilima.pdf
pet listening answer sheet pdf
55522798281.pdf
2021090208160895.pdf
78964872307.pdf
prentice hall worksheet answers
8771273586.pdf
lulusexiv.pdf
10141605887.pdf
82315068090.pdf
long island railroad map pdf
metallic bonds and properties of metals worksheet answers
6922493502.pdf