

Android media projection screen capture

I'm not robot  reCAPTCHA

**Continue**





How to stop screen recording screenpresso. What is screen record android.

Step 1 Step MediaProtes MediaPpaning Object Projection Planning = (MediaPismal Mediation) GetSystemService (Context.media Project Service); Startactivityforresult (Projectmanager.CreatescreenCapture (, / \* your\_request\_code \* /); Step 2 When the user presents the authorization request dialog box, the MediaProct object in  
MediaPstrounion, projection = (mediapismal intermediary) GetSystemService (Content.media Project Service); MPROUNATION MediaProjects = PrructionMagger.Getmediaprojection (Result, Data); . Screenshot = wm.Getdefaultdisplay (); Metrikler Displaymetrics Final = New Displaymetrics (); Screen.Getmetrics; Point size = New Point ();  
Screen.GetSize (Size); Last int mgenislik = you.x; Final int Mheight = Size.y; int myntry = metric. Final images Mmagereader = Imageerereader.Newinstance (Mwidth, Mheight, Pixelformat.rgba 8888, 2); The processor last processor = new processor (); Int signs = displaymanager.virtual\_display\_flag\_own\_content\_only |  
Displaymanager.virtual\_display\_flag\_public; Mpronion.createvirtualdisplay (Screen.Firror, Mwidth, Mheight, Mensity, Flags, Mzeriradere.Getsurface (), Null, Manager); Step 4 Take the screenshot onimageerereader.Setonimemaneavalisteme (New Imageerereadeer.onimageabilsterne (New Imageerereadeer.onimageabilsterne ( (@override Public Voidavailable (Imageerereader Reader)  
(reader.Setonageageleeste (Null, operator); ; int wowstride = aircraft [0]. Getwowstride (); int rowpadding = lowstride = - pixelstride \* metrics.widthpixels; // creation of bitmap BMP = Bitmap.createbitmap (Metrics.widthpixels + (FLOAT) Rowpadding ( (Flatr) Pixel ), Metrics.heightpixels, bitmap.config.argb 8888); . Geheight (); bmp.recycle ();  
something containing); A detailed code example has been removed from use /src/main/java/tw/firmaples/onscreencr/screenshot/screenshoter.java kootlin..Step 1 Question for MediaProjection MediaProjectionManager ProjectManager = (MediaProjectManager) getSystemService (context.media\_project\_Service);  
StartActivityForresult (ProjectionManager.CreateScreenCaptureIntent (), \* Your\_proof\_Code \* /); Step 2 When the user sends a permission dialog box, you can get the mediaproject object in MediaprojectManager projectManager = (context.media\_project\_service); MediaProjection MPROROROPTION = ProjectManager.getMediaProject (ResultCode,  
Data); Step 3. Create an Imagelerader and a VirtualDisplay that matches the screen. WindowManager wm = (windowManager) context.getSystemService (context.window\_service); Display display = wm.getDefulTdisplay (); final displayMetrics = new displaymetrics (); Display.getMetrics (Metrics); point size = new point (); Display.getRealSize (size);  
Final int mwidth = size.x; Final int mHeight = size.y; int mDensity = metric.densityDPI; Final Imagelerader MimagereeeAder = ImageReader.Newinstance (MWIDTH, MHEIGHT, PIXFORMAT.RGBA 8888, 2); EndHandler = NewHandler (); Int flags = displaymanager.virtual\_display\_flag\_on\_content\_only | DisplayManager.virtual\_display\_flag\_public;  
Mprojection.CreateVirtualDisplay ("Mirror Screen", Mwidth, MHeight, MDensity, Flags, Mimagereader.getSurface (), Null, Handler); Step 4. Take a screenshot in OnimageAvaliablelistener mimagereader.setonimageavailelistener (new imagelerader. tbuffer (); int pixelstride = panels [0]. GetpixelStride (); int rowStride = panels [0]. GetRowStride (); int  
rowpadding = rowStride - pixelstride \* metrics.widthpixels + (int) ((float) rowpadding / (float) pixelstride), metrics.heightpixels, bitmap.config.argb 8888); copyPixelElsFrombuffer (buffer); Image.close (); reader.close (); bitmap realSiztbitmap = bitmap.createbitmap (bmp, 0, 0, metrics.widthpixels, bmp.getHeight ());  
something with (RealizeBitMap \* /); handler); A full sample code can be found at Remaples/evertranslator/blob/master/detected/src/main/java/tw/firmaples/onscreencr/screenshot/screenshoter.java botlin version.Reference: In a recent project I was responsible for allowing users to save this to the device. The way the solution works can be  
identified by two classes: Lollipop has introduced a multimedia projection that gives us the possibility of capturing what is on the screen. MediaCard is quite explicit, this class simplifies audio and video recording. Android SDK. Authorizations v21write\_external\_storage and record\_audio You must first request authorization to capture the screen.  
The call to this intent will present the user with a confirmation dialog to authorize the screenshot. SURITIKITYRESULT, you can create a media record. Then we have to define our media record. I'm not too concerned about details so I used a predefined high quality video camera, but you can create yours and specify anything from the bits outputs  
per second. A slightly different thing is that I define the height and the width with the window pointers so that it exactly matches the size of the screen. One thing about Mediarecard is that the methods need to be defined in the right order. For example, after defining the camcorder profile, the audio source cannot be changed without plants. Javadoc  
has more details on the states behind it. If you don't care about the recording aspect of this demo, you can stick to your surface. Status bar S. When ready, call Stop(). It is also important to publish media projection and virtual display. The recording is then saved to the location you defined in SETSOUPFILE. In Android 5 (API level 21), the Media  
Projection API captures the content of your device's display in the form of a multimedia flow that you can read, store or diffuse on other devices such as TVs. The multimedia projection contains three representations of the device display: Figure 1. The actual display of the device projected on the virtual display. Virtual display content is  
writtenSurface. The multimedia projection intercepts the content of the device screen, then reflects the image intercepted on the virtual screen that creates the image on the surface. The app provides an area via surfaceview or imageler, which both consume the content of the intercepted screen. Onimageavailelistener of images allows you to  
direct the images created by the surface in real time. You can save images as registration or use them on TV or other devices. Start a multimedia projection session occupying a room to acquire the contents of the screen, the audio of the device or both. The part is represented by an example of the MediaProject class. After starting a new event, you  
can create an example of this class. To obtain a piece of multimedia projection using an old approach, old approach calls the mediapropracymanager system (MediaProjectManager. Approval window of the images dialog box showing the user the acquisition of all the information displayed, including information Reserved or personal,  
including sensitive information, including sensitive information or personal information osobiste info in informed. (Sound.) StartMediaproject.LaunchFinal Media Projection [] Media Projection = new Media Project [1]; ActivityResultLauncher startMedEAProiect = RegisterFityResuxt = RegisterFityResult (new startActivityForResule (), result -> {if (result.getResultCode ()  
== Activity.Result\_OK) {MediaProjection [0] = MediaProjectionManager. }); Wirtualny wyswietlacz Centralnym elementem projekcji multimedialnej jest wirtualny wyswietlacz tworzony przez wywołanie method createVirtualDisplay () w instancji MediaProjection: null) virtualDisplay = mediaProjection.createVirtualDisplay ("ScreenCapture", width,  
height, screendensity, DisplayManager.VIRTUAL\_DISPLAY\_FLAG\_AUTO\_MIRROR, zero, surface); The width and height parameters determine the width and height of the virtual display. To get values corresponding to the width and height of the media projection, use the Window API introduced in Android 11 (API level 30). WindowsMetrics media  
projection captures the entire screen whether the media projection application is running in full screen mode or many windows mode. To get the size of the media projection, use the WindowManager #getMaximumWindowMetrics () method, which returns WindowsMetrics for the full screen, even though the media projection application works in many  
Windows modes and only takes up a portion of the screen. To ensure compatibility on API 14, use WindowMetryScalCulator #ComputeMaximumWindowMetrics () from WindowManager Jetpack library. Call WindowMetrics #getBounds () to get the correct width and height of the virtual media projection view (see virtual view). Warning. Always change  
the size of the media projection application. Applications of variable size support changes in device configuration and many-window mode (see how many windows work). If the application size can't be changed, it should send a question to the screen boundaries from the window context and get WindowsMetrics of the maximum screen area available  
to the application using WindowManager #getMaximumWindowMetrics (); Val WindowContext = context.CoreAtewindowContext (context.Display !! type application, null) Val ProjectMetrics = windowContext.getSystemService (WindowManager :: class.java). MaximumWindowMetrics Context WindowContext =  
context.CoreAtewindowContext (context.getDisplay () ProjectMetrics = windowContext.getSystemService (windowManager.class). getMaximumWindowMetrics (); Note: Screen Density comes with a createEVirtualDisplay () method for you to customize the resulting screen resolution for external screen sharing or lower resolution. Use the #DensyDPI  
configuration instead of #GetRealMetrics () to get a density theme from CreenEVirTaldisplay (). The area must have a supporting surface to get the output at the desired resolution. Make a large size (low resolution) for screen transmission to TV or computer monitors (high resolution) to display the recording device. From 12L (API level 32), when the  
system forms a virtual display on the surface, reduced the virtual display on the surface using a process similar to the ImageView Center option. A new approach to reduction improves screen transmission for TV and other large screens, maximizing surface image size and ensuring the correct aspect ratio at the same time. Follow the following  
recommendations on how to get the best results with a multimedia projection: you can modify the modification of the application. Size apps support changes to device configuration and multi-finger mode (see multi-finals). Set the applications resizeBeBeActivity="True" in the manifest. Android 7.0 (API 24) and later This setting is true by default. Let  
your apps support both horizontal and vertical orientation, as both directives are common in phones, tablets, and typesetting. Use WindowManager #getMaximumWindowMetrics () to get multimedia projection limitations. Use Jetpack WindowManager to track level 14. (See WindowMetrics) If your application is not variable, get a media projection  
limit from the window context. (See WindowMetrics.) For more information on multimedia projection, see additional resources, see Video Playback and Audio Playback. The reproduction.

