

I was trying to create a simple camera application in android. For my application purpose I needed to display the camera preview. So I followed android developers site and also some other blogs to create a camera preview in android. So everything was working except there was one major bug. If I lock my device while the application is still running and then unlock, my application is no longer in the screen which is completely not what I wanted. So after doing some research I found that most of the tutorial I was following didn't take care of the application life cycle which is very important for this kind of application. Hence I'm writing this tutorial to show how to create a camera preview in android application. I'll try to show how to manage the full application life cycle for a camera application.

So let's begin with creating our layout xml. Paste the following xml in your projects *res/layout* file and name it "*activity_main.xml*"

[gist id="1af0817df0473bc88fdd714ce44d2cfa"]

In this Layout, we've used a *FrameLayout* which will display the camera preview.

Next we'll create a *CameraPreview.java* class which will extend *SurfaceView* class and will implement *SurfaceHolder.Callback*. The *SurfaceHolder.Callback* interface is implemented by a client to receive information about changes to the surface. Just put the following code in our *CameraPreview.java* class.

[gist id="c67b9fb54ea8b29775cffe9e366c502c"]

Constructor(CameraPreview): Line 20-21 installs our class as the callback interface to the display surface. So any time there is a change in the display surface, appropriate callback function will get called.

Line 27-35 initializes the *surfaceCreated* function. This function is called immediately after the surface is first created. As soon as the surface is created, we install a *SurfaceHolder* as the display object of the camera and then starts the preview.

Line 37-45 initializes the *surfaceDestroyed* function. This function gets called immediately before a surface is being destroyed. So we stop the camera preview.

Line 57-83 initializes the *surfaceChanged* function. This function gets called immediately after any structural changes (format or size) have been made to the surface. If orientation of our application is changed, we need to take care of that inside this function. Before we apply any change, we'll stop the camera preview at **line 68**. At **line 78-79**, we initialize our

camera preview with changed settings and then start the preview again.

Next we create our *Activity* class. Just write down the following code in a class named *HomeActivity.java*

[gist id="915cffbbe7490319e6aec595d1fd396b"]

getCameraInstance() opens an instance of *Camera* object.

onCreate() sets our layout XML as the view and initializes some object

onResume() function gets called immediately after the creation of the *Activity* and also once the application is resumed from background. So we need to take care of *Camera* initialization on this function. At **line 45** we initialize a *Camera* object. **Line 47-48** use the instance of *Camera* object to create an instance of our *CameraPreview* class(implements the *SurfaceHolder* callback methods) and add it as the view of our *FrameLayout*

releaseCamera() we use this function to release camera specific resources if the application goes to the background or the device gets locked. At **line 54** we stop the camera preview. **Line 55-56** releases the camera for other application. This is important as if our application pause without releasing the camera, other application won't be able to use it. **Line 57** removes all view from the *FrameLayout*. This is really important. Without this we won't be able to resume the application after device unlock.

So we are done with our tutorial on displaying camera preview in android. Hope this will help people to create a camera application in android which correctly and efficiently follows the application life cycle.