Capture Image from Camera and Display in Activity

I want to write a module where on a click of a button the camera opens and I can click and capture an image. If I don't like the image I can delete it and click one more image and then select the image and it should return back and display that image in the activity.

android image camera capture

edited May 13 '11 at 12:54
jengelsma
6,029 4 15 20

asked May 13 '11 at 11:40
Harsha M V
21.2k 91 278 441

3 you should open a camera intent, capture an image, save it on SDcard if want to, take an id through content provider, display it in dialog box with ok/cancel button. – Zoombie May 13 '11 at 12:27

1 @jengelsma i have one problem using camera i capture image in portraite mode and display in imageview then it's always display in landscape mode.have you nay idea?else can u slove this issue.plz reply i wait your answer. – Zala Janaksinh Feb 18 '13 at 7:19

@Harsha M V if u got the answer then reply me. – Zala Janaksinh Feb 18 '13 at 7:27

This Blog Can help you. startandroiddevelopment.blogspot.in/2013/10/… – user834900 Nov 1 '13 at 11:35

Refer the link stackoverflow.com/questions/13977245/… – Karthik Sridharan Oct 14 '16 at 10:57

10 Answers

Here's an example activity that will launch the camera app and then retrieve the image and display it.

```
package edu.gvsu.cis.masl.camerademo;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class MyCameraActivity extends Activity {
    private static final int CAMERA_REQUEST = 1888;
    private ImageView imageView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        this.imageView = (ImageView)this.findViewById(R.id.imageView1);
        Button photoButton = (Button) this.findViewById(R.id.button1);
        photoButton.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent cameraIntent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
                startActivityForResult(cameraIntent, CAMERA_REQUEST);
            }
        });
    }

    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (requestCode == CAMERA_REQUEST && resultCode == Activity.RESULT_OK) {
            Bitmap photo = (Bitmap) data.getExtras().get("data");
            imageView.setImageBitmap(photo);
        }
    }
}
```

Note that the camera app itself gives you the ability to review/retake the image, and once an image is accepted, the activity displays it.

Here is the layout that the above activity uses. It is simply a LinearLayout containing a Button with id button1 and an ImageView with id imageview1:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <Button android:id="@+id/button1" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="@string/photo"></Button>
    <ImageView android:id="@+id/imageView1" android:layout_height="wrap_content"
android:src="@drawable/icon" android:layout_width="wrap_content"></ImageView>

</LinearLayout>
```

And one final detail, be sure to add:

```
<uses-feature android:name="android.hardware.camera"></uses-feature>
```

and if camera is optional to your app functionality. make sure to set require to false in the permission. like this

```
<uses-feature android:name="android.hardware.camera" android:required="false"></uses-
feature>
```

to your manifest.xml.

edited Oct 3 '16 at 18:13
ErickBergmann
497 2 10

answered May 13 '11 at 12:23
jengelsma
6,029 4 15 20

16 Bitmap photo = (Bitmap) data.getExtras().get("data"); Its giving a Null pointer expection. When i open the app and click Capture.. it goes to the camera app and after 2-4 secs the whole thing crashes. – Harsha M V May 16 '11 at 10:04

6 I suppose to be on the safe side we should be checking to see if resultCode == Activity.RESULT_OK in the onActivityResult() method. As to why the camera is crashing, we'd have to see a stack trace. – jengelsma May 16 '11 at 11:56

4 @Harsha M V This is a known bug on Samsung Galaxy. Please see this answer stackoverflow.com/questions/7031374/… – Oh Danny Boy Sep 29 '11 at 19:16

7 @WillKru if the feature is not mandatory, you should add the `<uses-feature/>` with the `android:required="false"` attribute: `<uses-feature android:name="android.hardware.camera" android:required="false"></uses-feature>` – Mannaz Oct 8 '12 at 9:06

77 Also note that `Bitmap photo = (Bitmap) data.getExtras().get("data");` does not grab the image that was taken. It grabs a thumbnail of the image that was taken. – Billy Coover Feb 4 '14 at 6:19

It took me some hours to get this working. The code it's almost a copy-paste from developer.android.com, with a minor difference.

Request this permission on the `AndroidManifest.xml` :

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

On your `Activity` , start by defining this:

```
static final int REQUEST_IMAGE_CAPTURE = 1;
private Bitmap mImageBitmap;
private String mCurrentPhotoPath;
private ImageView mImageView;
```

Then fire this `Intent` in an `onClick` :

```
Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
if (cameraIntent.resolveActivity(getPackageManager()) != null) {
    // Create the File where the photo should go
    File photoFile = null;
    try {
        photoFile = createImageFile();
    } catch (IOException ex) {
        // Error occurred while creating the File
        Log.i(TAG, "IOException");
    }
    // Continue only if the File was successfully created
    if (photoFile != null) {
        cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(photoFile));
        startActivityForResult(cameraIntent, REQUEST_IMAGE_CAPTURE);
    }
}
```

Add the following support method:

```
private File createImageFile() throws IOException {
    // Create an image file name
    String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
    String imageFileName = "JPEG_" + timeStamp + "_";
    File storageDir = Environment.getExternalStoragePublicDirectory(
            Environment.DIRECTORY_PICTURES);
    File image = File.createTempFile(
```