[Android saving file to external storage](http://stackoverflow.com/questions/7887078/android-saving-file-to-external-storage)

|  |
| --- |
| I have a little issue with creating a directory and saving a file to it on my android application. I'm using this piece of code to do this :  String filename = "MyApp/MediaTag/MediaTag-"+objectId+".png";  File file = new File(Environment.getExternalStorageDirectory(), filename);  FileOutputStream fos;  fos = new FileOutputStream(file);  fos.write(mediaTagBuffer);  fos.flush();  fos.close();  But it's throwing an exception :  java.io.FileNotFoundException: /mnt/sdcard/MyApp/MediaCard/MediaCard-0.png (No such file or directory)  on that line : fos = new FileOutputStream(file);  If I set the filename to : "MyApp/MediaTag-"+objectId+" it's working, but If I try to create and save the file to an another directory it's throwing the exception. So any ideas what I'm doing wrong?  And another question: Is there any way to make my files private in external storage so user can't see them in gallery, only if he connect his device as Disk Drive? |

|  |
| --- |
| Use this function to save your bitmap in SD card  private void SaveImage(Bitmap finalBitmap) {  String root = Environment.getExternalStorageDirectory().toString();  File myDir = new File(root + "/saved\_images");  myDir.mkdirs();  Random generator = new Random();  int n = 10000;  n = generator.nextInt(n);  String fname = "Image-"+ n +".jpg";  File file = new File (myDir, fname);  if (file.exists ()) file.delete ();  try {  FileOutputStream out = new FileOutputStream(file);  finalBitmap.compress(Bitmap.CompressFormat.JPEG, 90, out);  out.flush();  out.close();  } catch (Exception e) {  e.printStackTrace();  }  }  and add this in manifest  <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  **EDIT:** By using this line you can able to see saved images in the gallery view.  sendBroadcast(new Intent(  Intent.ACTION\_MEDIA\_MOUNTED,  Uri.parse("file://" + Environment.getExternalStorageDirectory())));  look at this link also <http://rajareddypolam.wordpress.com/?p=3&preview=true> |
| |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | 9 |  | | You should still be using Environment.getExternalStorageDirectory() instead of /sdcard. – [Che Jami](http://stackoverflow.com/users/828060/che-jami" \o "4,205 reputation) [Oct 25 '11 at 10:41](http://stackoverflow.com/questions/7887078/android-saving-file-to-external-storage#comment9627093_7887114) | |

|  |
| --- |
| The code presented by RajaReddy no longer works for KitKat  This one does (2 changes):  private void saveImageToExternalStorage(Bitmap finalBitmap) {  String root = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY\_PICTURES).toString();  File myDir = new File(root + "/saved\_images");  myDir.mkdirs();  Random generator = new Random();  int n = 10000;  n = generator.nextInt(n);  String fname = "Image-" + n + ".jpg";  File file = new File(myDir, fname);  if (file.exists())  file.delete();  try {  FileOutputStream out = new FileOutputStream(file);  finalBitmap.compress(Bitmap.CompressFormat.JPEG, 90, out);  out.flush();  out.close();  }  catch (Exception e) {  e.printStackTrace();  }  // Tell the media scanner about the new file so that it is  // immediately available to the user.  MediaScannerConnection.scanFile(this, new String[] { file.toString() }, null,  new MediaScannerConnection.OnScanCompletedListener() {  public void onScanCompleted(String path, Uri uri) {  Log.i("ExternalStorage", "Scanned " + path + ":");  Log.i("ExternalStorage", "-> uri=" + uri);  }  });  } |
| \ |

|  |  |
| --- | --- |
|  | Probably exception is thrown because there is no MediaCard subdir. You should check if all dirs in the path exist.  About visibility of your files: if you put file named .nomedia in your dir you are telling Android that you don't want it to scan it for media files and they will not appear in the gallery. |
|  | add a comment |
|  |  |
|  | Try This :   1. Check External storage device 2. Write File 3. Read File   public class WriteSDCard extends Activity {  private static final String TAG = "MEDIA";  private TextView tv;  @Override  public void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.main);  tv = (TextView) findViewById(R.id.TextView01);  checkExternalMedia();  writeToSDFile();  readRaw();  }  /\*\*  \* Method to check whether external media available and writable. This is  \* adapted from  \* http://developer.android.com/guide/topics/data/data-storage.html  \* #filesExternal  \*/  private void checkExternalMedia() {  boolean mExternalStorageAvailable = false;  boolean mExternalStorageWriteable = false;  String state = Environment.getExternalStorageState();  if (Environment.MEDIA\_MOUNTED.equals(state)) {  // Can read and write the media  mExternalStorageAvailable = mExternalStorageWriteable = true;  } else if (Environment.MEDIA\_MOUNTED\_READ\_ONLY.equals(state)) {  // Can only read the media  mExternalStorageAvailable = true;  mExternalStorageWriteable = false;  } else {  // Can't read or write  mExternalStorageAvailable = mExternalStorageWriteable = false;  }  tv.append("\n\nExternal Media: readable=" + mExternalStorageAvailable  + " writable=" + mExternalStorageWriteable);  }  /\*\*  \* Method to write ascii text characters to file on SD card. Note that you  \* must add a WRITE\_EXTERNAL\_STORAGE permission to the manifest file or this  \* method will throw a FileNotFound Exception because you won't have write  \* permission.  \*/  private void writeToSDFile() {  // Find the root of the external storage.  // See http://developer.android.com/guide/topics/data/data-  // storage.html#filesExternal  File root = android.os.Environment.getExternalStorageDirectory();  tv.append("\nExternal file system root: " + root);  // See  // http://stackoverflow.com/questions/3551821/android-write-to-sd-card-folder  File dir = new File(root.getAbsolutePath() + "/download");  dir.mkdirs();  File file = new File(dir, "myData.txt");  try {  FileOutputStream f = new FileOutputStream(file);  PrintWriter pw = new PrintWriter(f);  pw.println("Hi , How are you");  pw.println("Hello");  pw.flush();  pw.close();  f.close();  } catch (FileNotFoundException e) {  e.printStackTrace();  Log.i(TAG, "\*\*\*\*\*\*\* File not found. Did you"  + " add a WRITE\_EXTERNAL\_STORAGE permission to the manifest?");  } catch (IOException e) {  e.printStackTrace();  }  tv.append("\n\nFile written to " + file);  }  /\*\*  \* Method to read in a text file placed in the res/raw directory of the  \* application. The method reads in all lines of the file sequentially.  \*/  private void readRaw() {  tv.append("\nData read from res/raw/textfile.txt:");  InputStream is = this.getResources().openRawResource(R.raw.textfile);  InputStreamReader isr = new InputStreamReader(is);  BufferedReader br = new BufferedReader(isr, 8192); // 2nd arg is buffer  // size  // More efficient (less readable) implementation of above is the  // composite expression  /\*  \* BufferedReader br = new BufferedReader(new InputStreamReader(  \* this.getResources().openRawResource(R.raw.textfile)), 8192);  \*/  try {  String test;  while (true) {  test = br.readLine();  // readLine() returns null if no more lines in the file  if (test == null) break;  tv.append("\n" + " " + test);  }  isr.close();  is.close();  br.close();  } catch (IOException e) {  e.printStackTrace();  }  tv.append("\n\nThat is all");  }  } |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | 2 |  | | This looks very similar to the code from here: [stackoverflow.com/a/8330635/19679](http://stackoverflow.com/a/8330635/19679) . If it was drawn from there, you should probably cite that in your answer. – [Brad Larson**♦**](http://stackoverflow.com/users/19679/brad-larson) [May 29 '14 at 17:17](http://stackoverflow.com/questions/7887078/android-saving-file-to-external-storage#comment36872547_13620774) |   add a comment |

|  |  |  |
| --- | --- | --- |
| since android 4.4 file saving has been changed. there is  ContextCompat.getExternalFilesDirs(context, name);  it retuns an array.  when name is null  the first value is like /storage/emulated/0/Android/com.my.package/files  the second value is like /storage/extSdCard/Android/com.my.package/files  android 4.3 and less it retuns a single item array  parts of little messy code but it demonstrates how it works:  /\*\* Create a File for saving an image or video  \* @throws Exception \*/  private File getOutputMediaFile(int type) throws Exception{  // Check that the SDCard is mounted  File mediaStorageDir;  if(internalstorage.isChecked())  {  mediaStorageDir = new File(getFilesDir().getAbsolutePath() );  }  else  {  File[] dirs=ContextCompat.getExternalFilesDirs(this, null);  mediaStorageDir = new File(dirs[dirs.length>1?1:0].getAbsolutePath() );  }  // Create the storage directory(MyCameraVideo) if it does not exist  if (! mediaStorageDir.exists()){  if (! mediaStorageDir.mkdirs()){  output.setText("Failed to create directory.");  Toast.makeText(this, "Failed to create directory.", Toast.LENGTH\_LONG).show();  Log.d("myapp", "Failed to create directory");  return null;  }  }  // Create a media file name  // For unique file name appending current timeStamp with file name  java.util.Date date= new java.util.Date();  String timeStamp = new SimpleDateFormat("yyyyMMdd\_HHmmss",Locale.ENGLISH) .format(date.getTime());  File mediaFile;  if(type == MEDIA\_TYPE\_VIDEO) {  // For unique video file name appending current timeStamp with file name  mediaFile = new File(mediaStorageDir.getPath() + File.separator + slpid + "\_" + pwsid + "\_" + timeStamp + ".mp4");  }  else if(type == MEDIA\_TYPE\_AUDIO) {  // For unique video file name appending current timeStamp with file name  mediaFile = new File(mediaStorageDir.getPath() + File.separator + slpid + "\_" + pwsid + "\_" + timeStamp + ".3gp");  } else {  return null;  }  return mediaFile;  }  /\*\* Create a file Uri for saving an image or video  \* @throws Exception \*/  private Uri getOutputMediaFileUri(int type) throws Exception{  return Uri.fromFile(getOutputMediaFile(type));  }  //usage:  try {  file=getOutputMediaFileUri(MEDIA\_TYPE\_AUDIO).getPath();  } catch (Exception e1) {  e1.printStackTrace();  return;  }   |  |  | | --- | --- | |  |  | |
| add a comment |
| I have created an AsyncTask for saving bitmaps.  public class BitmapSaver extends AsyncTask<Void, Void, Void>  {  public static final String TAG ="BitmapSaver";  private Bitmap bmp;  private Context ctx;  private File pictureFile;  public BitmapSaver(Context paramContext , Bitmap paramBitmap)  {  ctx = paramContext;  bmp = paramBitmap;  }  /\*\* Create a File for saving an image or video \*/  private File getOutputMediaFile()  {  // To be safe, you should check that the SDCard is mounted  // using Environment.getExternalStorageState() before doing this.  File mediaStorageDir = new File(Environment.getExternalStorageDirectory()  + "/Android/data/"  + ctx.getPackageName()  + "/Files");  // This location works best if you want the created images to be shared  // between applications and persist after your app has been uninstalled.  // Create the storage directory if it does not exist  if (! mediaStorageDir.exists()){  if (! mediaStorageDir.mkdirs()){  return null;  }  }  // Create a media file name  String timeStamp = new SimpleDateFormat("ddMMyyyy\_HHmm").format(new Date());  File mediaFile;  String mImageName="MI\_"+ timeStamp +".jpg";  mediaFile = new File(mediaStorageDir.getPath() + File.separator + mImageName);  return mediaFile;  }  protected Void doInBackground(Void... paramVarArgs)  {  this.pictureFile = getOutputMediaFile();  if (this.pictureFile == null) { return null; }  try  {  FileOutputStream localFileOutputStream = new FileOutputStream(this.pictureFile);  this.bmp.compress(Bitmap.CompressFormat.PNG, 90, localFileOutputStream);  localFileOutputStream.close();  }  catch (FileNotFoundException localFileNotFoundException)  {  return null;  }  catch (IOException localIOException)  {  }  return null;  }  protected void onPostExecute(Void paramVoid)  {  super.onPostExecute(paramVoid);  try  {  //it will help you broadcast and view the saved bitmap in Gallery  this.ctx.sendBroadcast(new Intent("android.intent.action.MEDIA\_MOUNTED", Uri  .parse("file://" + Environment.getExternalStorageDirectory())));  Toast.makeText(this.ctx, "File saved", 0).show();  return;  }  catch (Exception localException1)  {  try  {  Context localContext = this.ctx;  String[] arrayOfString = new String[1];  arrayOfString[0] = this.pictureFile.toString();  MediaScannerConnection.scanFile(localContext, arrayOfString, null,  new MediaScannerConnection.OnScanCompletedListener()  {  public void onScanCompleted(String paramAnonymousString ,  Uri paramAnonymousUri)  {  }  });  return;  }  catch (Exception localException2)  {  }  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  |  | | How i can save gif image?? – [ER Vishal Senjaliya](http://stackoverflow.com/users/6193960/er-vishal-senjaliya) [Oct 3 at 10:10](http://stackoverflow.com/questions/7887078/android-saving-file-to-external-storage#comment66948072_29795857) | | |  |  | | --- | --- | | 1 |  | | Gif image contains multiple images. you have to first seperate those frames then you can use this method. It's my opinion. – [Nepster](http://stackoverflow.com/users/3496570/nepster" \o "11,989 reputation) [4 hours ago](http://stackoverflow.com/questions/7887078/android-saving-file-to-external-storage#comment67184121_29795857) | | |  |  | | --- | --- | | 1 |  | | i did it from [stackoverflow.com/questions/39826400/…](http://stackoverflow.com/questions/39826400/how-to-save-gif-image-in-sdcard) – [ER Vishal Senjaliya](http://stackoverflow.com/users/6193960/er-vishal-senjaliya) [4 hours ago](http://stackoverflow.com/questions/7887078/android-saving-file-to-external-storage#comment67184486_29795857) |   add a comment |

|  |  |  |
| --- | --- | --- |
| up vote0down vote | This code is Working great & Worked on KitKat as well. Appreciate @RajaReddy PolamReddy Added few more steps here and also Visible on Gallery as well.  public void SaveOnClick(View v){  File mainfile;  String fpath;  try {  //i.e v2:My view to save on own folder  v2.setDrawingCacheEnabled(true);  //Your final bitmap according to my code.  bitmap\_tmp = v2.getDrawingCache();  File(getExternalFilesDir(Environment.DIRECTORY\_PICTURES)+File.separator+"/MyFolder");  Random random=new Random();  int ii=100000;  ii=random.nextInt(ii);  String fname="MyPic\_"+ ii + ".jpg";  File direct = new File(Environment.getExternalStorageDirectory() + "/MyFolder");  if (!direct.exists()) {  File wallpaperDirectory = new File("/sdcard/MyFolder/");  wallpaperDirectory.mkdirs();  }  mainfile = new File(new File("/sdcard/MyFolder/"), fname);  if (mainfile.exists()) {  mainfile.delete();  }  FileOutputStream fileOutputStream;  fileOutputStream = new FileOutputStream(mainfile);  bitmap\_tmp.compress(CompressFormat.JPEG, 100, fileOutputStream);  Toast.makeText(MyActivity.this.getApplicationContext(), "Saved in Gallery..", Toast.LENGTH\_LONG).show();  fileOutputStream.flush();  fileOutputStream.close();  fpath=mainfile.toString();  galleryAddPic(fpath);  } catch(FileNotFoundException e){  e.printStackTrace();  } catch (IOException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  }  This is Media scanner to Visible in Gallery.  private void galleryAddPic(String fpath) {  Intent mediaScanIntent = new Intent("android.intent.action.MEDIA\_SCANNER\_SCAN\_FILE");  File f = new File(fpath);  Uri contentUri = Uri.fromFile(f);  mediaScanIntent.setData(contentUri);  this.sendBroadcast(mediaScanIntent);  }   |  | | --- | |  | |