**ImageView из Файла**

Labeeb is right about why you need to set image using path if your resources are already laying inside the resource folder ,

This kind of path is needed only when your images are stored in SD-Card .

And **try the below code to set Bitmap images from a file stored inside a SD-Card .**

File imgFile = new File("/sdcard/Images/test\_image.jpg");

if(imgFile.exists()){

Bitmap myBitmap = BitmapFactory.decodeFile(imgFile.getAbsolutePath());

ImageView myImage = (ImageView) findViewById(R.id.imageviewTest);

myImage.setImageBitmap(myBitmap);

}

And include this permission in the manifest file:

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

I think you can use this

Bitmap bmImg = BitmapFactory.decodeFile("path of your img1");

imageView.setImageBitmap(bmImg);

Bitmap bmImg = BitmapFactory.decodeFile("res\\drawable-hdpi\\img1.png"); I set path like the above But i am not able to show the image – [Alex](http://stackoverflow.com/users/482568/alex) [Nov 15 '10 at 6:21](http://stackoverflow.com/questions/4181774/show-image-view-from-file-path#comment4518275_4181814)

Ho you cannot use that path of your res here. that can only be used to store image form sd card or other memory. I wondered, why u need to set image using path if your resource is there in resource folder – [Labeeb Panampullan](http://stackoverflow.com/users/477754/labeeb-panampullan" \o "18,335 reputation) [Nov 15 '10 at 6:28](http://stackoverflow.com/questions/4181774/show-image-view-from-file-path#comment4518320_4181814)

This is not working when I put "/storage/emulated/0/screen\_8.jpeg" in path. Can anyone help? – [Basim Majeed](http://stackoverflow.com/users/1900511/basim-majeed" \o "161 reputation) [Dec 9 '14 at 10:23](http://stackoverflow.com/questions/4181774/show-image-view-from-file-path#comment43202784_4181814)

add a comment

up vote18down vote You can also use:

File imgFile = new File(“filepath”);

if(imgFile.exists())

{

ImageView myImage = new ImageView(this);

myImage.setImageURI(Uri.fromFile(imgFile));

}

This does the bitmap decoding implicit for you.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | 4 |  | | work on UI thread so not preferable – [Vivek](http://stackoverflow.com/users/484073/vivek" \o "2,418 reputation) [Nov 19 '13 at 5:55](http://stackoverflow.com/questions/4181774/show-image-view-from-file-path#comment29886789_8065952) | | |  |  | | --- | --- | |  |  | | But avoid creating a memory hungry bitmap? wouldn't it be better if placed in an AsyncTask or thread?? – [Waza\_Be](http://stackoverflow.com/users/327402/waza-be" \o "22,446 reputation) [Apr 7 '15 at 8:55](http://stackoverflow.com/questions/4181774/show-image-view-from-file-path#comment47134407_8065952) | |

String path = Environment.getExternalStorageDirectory()+ "/Images/test.jpg";

File imgFile = new File(path);

if(imgFile.exists())

{

Bitmap myBitmap = BitmapFactory.decodeFile(imgFile.getAbsolutePath());

ImageView imageView=(ImageView)findViewById(R.id.imageView);

imageView.setImageBitmap(myBitmap);

}

+1 for Environment.getExternalStorageDirectory() – [Denis Kniazhev](http://stackoverflow.com/users/369317/denis-kniazhev) [Oct 4 '13 at 11:48](http://stackoverflow.com/questions/4181774/show-image-view-from-file-path#comment28376761_9920555)

add a comment

up vote7down vote All the answers are outdated. It is best to use [picasso](http://square.github.io/picasso/) for such purposes. It has a lot of features including background image processing.

Did I mention it is super easy to use:

Picasso.with(context).load(new File(...)).into(imageView);

I do this but sometimes my imgeview is empty.what can I do? – [saleh sereshki](http://stackoverflow.com/users/3986343/saleh-sereshki" \o "403 reputation) [Oct 22 '15 at 9:21](http://stackoverflow.com/questions/4181774/show-image-view-from-file-path#comment54355575_33052165)

@salehsereshki there was probably an error fetching the file. You can always display a error or placeholder as.placeholder(R.drawable.user\_placeholder) or .error(R.drawable.user\_placeholder\_error) – [vedant1811](http://stackoverflow.com/users/1396264/vedant1811) [Oct 22 '15 at 17:44](http://stackoverflow.com/questions/4181774/show-image-view-from-file-path#comment54374670_33052165)

ImageView imgView = new ImageView(this);

InputStream is = getClass().getResourceAsStream("/drawable/" + fileName);

imgView.setImageDrawable(Drawable.createFromStream(is, ""));

public static Bitmap decodeFile(String path) {

Bitmap b = null;

File f = new File(path);

// Decode image size

BitmapFactory.Options o = new BitmapFactory.Options();

o.inJustDecodeBounds = true;

FileInputStream fis = null;

try {

fis = new FileInputStream(f);

BitmapFactory.decodeStream(fis, null, o);

fis.close();

int IMAGE\_MAX\_SIZE = 1024; // maximum dimension limit

int scale = 1;

if (o.outHeight > IMAGE\_MAX\_SIZE || o.outWidth > IMAGE\_MAX\_SIZE) {

scale = (int) Math.pow(2, (int) Math.round(Math.log(IMAGE\_MAX\_SIZE / (double) Math.max(o.outHeight, o.outWidth)) / Math.log(0.5)));

}

// Decode with inSampleSize

BitmapFactory.Options o2 = new BitmapFactory.Options();

o2.inSampleSize = scale;

fis = new FileInputStream(f);

b = BitmapFactory.decodeStream(fis, null, o2);

fis.close();

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return b;

}

public static Bitmap showBitmapFromFile(String file\_path)

{

try {

File imgFile = new File(file\_path);

if(imgFile.exists()){

Bitmap pic\_Bitmap = decodeFile(file\_path);

return pic\_Bitmap;

}

} catch (Exception e) {

MyLog.e("Exception showBitmapFromFile");

return null;

}

return null;

}

if you are using image loading in List view then use Aquery concept .

<https://github.com/AshishPsaini/AqueryExample>

AQuery aq= new AQuery((Activity) activity, convertView);

//load image from file, down sample to target width of 250 pixels .gi

File file=new File("//pic/path/here/aaaa.jpg");

if(aq!=null)

aq.id(holder.pic\_imageview).image(file, 250);

From the official site: <http://developer.android.com/training/displaying-bitmaps/load-bitmap.html>

ImageView image = (ImageView) findViewById(R.id.imagePreview);

try {

image.setImageBitmap(decodeSampledBitmap(picFilename));

} catch (Exception e) {

e.printStackTrace();

}

Here the methods:

private int calculateInSampleSize(

BitmapFactory.Options options, int reqWidth, int reqHeight) {

// Raw height and width of image

final int height = options.outHeight;

final int width = options.outWidth;

int inSampleSize = 1;

if (height > reqHeight || width > reqWidth) {

final int halfHeight = height / 2;

final int halfWidth = width / 2;

// Calculate the largest inSampleSize value that is a power of 2 and keeps both

// height and width larger than the requested height and width.

while ((halfHeight / inSampleSize) > reqHeight

&& (halfWidth / inSampleSize) > reqWidth) {

inSampleSize \*= 2;

}

}

return inSampleSize;

}

private Bitmap decodeSampledBitmap(String pathName,

int reqWidth, int reqHeight) {

// First decode with inJustDecodeBounds=true to check dimensions

final BitmapFactory.Options options = new BitmapFactory.Options();

options.inJustDecodeBounds = true;

BitmapFactory.decodeFile(pathName, options);

// Calculate inSampleSize

options.inSampleSize = calculateInSampleSize(options, reqWidth, reqHeight);

// Decode bitmap with inSampleSize set

options.inJustDecodeBounds = false;

return BitmapFactory.decodeFile(pathName, options);

}

//I added this to have a good approximation of the screen size:

private Bitmap decodeSampledBitmap(String pathName) {

Display display = getWindowManager().getDefaultDisplay();

Point size = new Point();

display.getSize(size);

int width = size.x;

int height = size.y;

return decodeSampledBitmap(pathName, width, height);

}

You may use this to access a specific folder and get particular image

public void Retrieve(String path, String Name)

{

File imageFile = new File(path+Name);

if(imageFile.exists()){

Bitmap myBitmap = BitmapFactory.decodeFile(path+Name);

myImage = (ImageView) findViewById(R.id.savedImage);

myImage.setImageBitmap(myBitmap);

Toast.makeText(SaveImage.this, myBitmap.toString(), Toast.LENGTH\_LONG).show();

}

}

And then you can call it by

Retrieve(Environment.getExternalStorageDirectory().toString()+

"/Aqeel/Images/","Image2.PNG");

Toast.makeText(SaveImage.this, "Saved", Toast.LENGTH\_LONG).show();