Reduce Bitmap size using BitmapFactory.Options.inSampleSize

In the post "[Load ImageView with JPG file in SD Card](http://android-coding.blogspot.com/2011/06/load-imageview-with-jpg-file-in-sd-card.html)", the ImageView is loaded with bitmap in full-size. It cost too much resources for a mobile device, and easy to make the app closed unexpectly.  
  
We can generate a shrinked bitmap using [BitmapFactory.Options](http://developer.android.com/reference/android/graphics/BitmapFactory.Options.html) with inSampleSize; such that to reduce the needed resources greatly.

[?](http://android-coding.blogspot.ru/2011/06/reduce-bitmap-size-using.html)

|  |
| --- |
| package com.AndroidLoadImageView;    import android.app.Activity;  import android.graphics.Bitmap;  import android.graphics.BitmapFactory;  import android.os.Bundle;  import android.widget.ImageView;    public class AndroidLoadImageViewActivity extends Activity {     String imagefile ="/sdcard/IMG\_9331.JPG";        /\*\* Called when the activity is first created. \*/      @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.main);            ImageView image = (ImageView)findViewById(R.id.image);          //Bitmap bm = BitmapFactory.decodeFile(imagefile);          Bitmap bm = ShrinkBitmap(imagefile, 300, 300);          image.setImageBitmap(bm);      }     Bitmap ShrinkBitmap(String file, int width, int height){         BitmapFactory.Options bmpFactoryOptions = new BitmapFactory.Options();          bmpFactoryOptions.inJustDecodeBounds = true;          Bitmap bitmap = BitmapFactory.decodeFile(file, bmpFactoryOptions);            int heightRatio = (int)Math.ceil(bmpFactoryOptions.outHeight/(float)height);          int widthRatio = (int)Math.ceil(bmpFactoryOptions.outWidth/(float)width);            if (heightRatio > 1 || widthRatio > 1)          {           if (heightRatio > widthRatio)           {            bmpFactoryOptions.inSampleSize = heightRatio;           } else {            bmpFactoryOptions.inSampleSize = widthRatio;           }          }            bmpFactoryOptions.inJustDecodeBounds = false;          bitmap = BitmapFactory.decodeFile(file, bmpFactoryOptions);       return bitmap;      }  } |