Ready image file using build-in Gallery, with Intent.ACTION\_PICK

We can start a activity with Intent.ACTION\_PICK, to start Android build-in Gallery app to select image.   
  
**Intent intent = new Intent(Intent.ACTION\_PICK, android.provider.MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI);  
startActivityForResult(intent, RQS\_GET\_IMAGE);**  
  
The selected image can be retrieved using data.getData() in onActivityResult().  
  
**Uri imageUri = data.getData();**  
  
In order to make use of ShrinkBitmap() in last post "[Reduce Bitmap size using BitmapFactory.Options.inSampleSize](http://android-coding.blogspot.com/2011/06/reduce-bitmap-size-using.html)", re-write ShrinkBitmap(Uri uri, int width, int height), instead of ShrinkBitmap(String file, int width, int height); to get a reduced size bitmap.

[?](http://android-coding.blogspot.ru/2011/06/ready-image-file-using-build-in-gallery.html)

|  |
| --- |
| package com.AndroidLoadImageView;    import java.io.FileNotFoundException;    import android.app.Activity;  import android.content.Intent;  import android.graphics.Bitmap;  import android.graphics.BitmapFactory;  import android.net.Uri;  import android.os.Bundle;  import android.widget.ImageView;  import android.widget.Toast;    public class AndroidLoadImageViewActivity extends Activity {     final static int RQS\_GET\_IMAGE = 1;     ImageView image;        /\*\* Called when the activity is first created. \*/      @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.main);            image = (ImageView)findViewById(R.id.image);          Intent intent = new Intent(Intent.ACTION\_PICK,            android.provider.MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI);          startActivityForResult(intent, RQS\_GET\_IMAGE);        }     Bitmap ShrinkBitmap(Uri uri, int width, int height){             BitmapFactory.Options bmpFactoryOptions = new BitmapFactory.Options();              bmpFactoryOptions.inJustDecodeBounds = true;                Bitmap bitmap = null;;     try {      bitmap = BitmapFactory.decodeStream(getContentResolver().openInputStream(uri));        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.decodeStream(getContentResolver().openInputStream(uri));       } catch (FileNotFoundException e) {      // TODO Auto-generated catch block      e.printStackTrace();     }       return bitmap;      }     @Override   protected void onActivityResult(int requestCode, int resultCode, Intent data) {    // TODO Auto-generated method stub    super.onActivityResult(requestCode, resultCode, data);      if (resultCode == RESULT\_OK){     if(requestCode == RQS\_GET\_IMAGE){      Uri imageUri = data.getData();      String imageFile = imageUri.toString();      Toast.makeText(AndroidLoadImageViewActivity.this,        imageFile,        Toast.LENGTH\_LONG).show();      Bitmap bm = ShrinkBitmap(imageUri, 300, 300);            image.setImageBitmap(bm);     }      }   }  } |

[?](http://android-coding.blogspot.ru/2011/06/ready-image-file-using-build-in-gallery.html)

|  |
| --- |
| <?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"      >  <TextView      android:layout\_width="fill\_parent"      android:layout\_height="wrap\_content"      android:text="@string/hello"      />  <ImageView   android:id="@+id/image"      android:layout\_width="fill\_parent"      android:layout\_height="wrap\_content"      />  </LinearLayout> |