для отображения ListView с изображения и текста, я использовал класс Simpleadapter, но я не могу отобразить элемент управления ListView Может ли один помочь мне, что я сделал не так ?

Here My Code:

public class HostsActivity extends ListActivity {

public final static String ITEM\_TITLE = "title";

public final static String ITEM\_IMAGE = "Image";

private List<HostsProfile> hostsProfile1;

List<Map<String, ?>> security = new LinkedList<Map<String, ?>>();

@Override

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.hostslist);

Bundle recdData = getIntent().getExtras();

hostsProfile1 = recdData.getParcelableArrayList("hostsProfile");

List<Map<String, ?>> security1 = new LinkedList<Map<String, ?>>();

for (HostsProfile msg1 : hostsProfile1) {

// Log.v("image","Image path"+msg.getImage());

try {

URL url = new URL(msg1.getMediathumbnail());

HttpURLConnection con = (HttpURLConnection) url.openConnection();

InputStream is = con.getInputStream();

Bitmap img = BitmapFactory.decodeStream(is);

security.add(createItem(msg1.getTitle(), img));

} catch (Throwable t) {

}

}

/\*for(int i=0; i<security.size(); i++){

security.remove(0);

}

\*/

SimpleAdapter adapter1 = new SimpleAdapter(this, security1,

R.layout.image\_text\_layout1, new String[] { ITEM\_IMAGE,

ITEM\_TITLE }, new int[] { R.id.feed\_image,

R.id.job\_text });

adapter1.setViewBinder(new MyViewBinder());

this.setListAdapter(adapter1);

}

public Map<String, ?> createItem(String title, Bitmap Image) {

Map<String, Object> item = new HashMap<String, Object>();

item.put(ITEM\_TITLE, title);

item.put(ITEM\_IMAGE, Image);

return item;

}

//for ImageView

class MyViewBinder implements ViewBinder {

public boolean setViewValue(View view, Object data,

String textRepresentation) {

if ((view instanceof ImageView) & (data instanceof Bitmap)) {

ImageView image1 = (ImageView) view;

Bitmap bm = (Bitmap) data;

image1.setImageBitmap(bm);

return true;

}

return false;

}

}

}

Вот мои макета XML файлов:

image\_text\_layout1 :

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="horizontal" android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content">

<ImageView android:id="@+id/feed\_image"

android:layout\_width="60dp"

android:layout\_height="60dp"

android:gravity="center\_vertical"

android:paddingRight="3dp"

android:paddingLeft="2dp"/>

<TextView android:id="@+id/job\_text"

android:layout\_width="wrap\_content"

android:layout\_height="60dp"

android:textColor="#000000"

android:textStyle="bold"

android:gravity="center\_vertical"/>

<!--android:paddingTop="5dip"

android:paddingBottom="28dip"

android:paddingLeft="8dip"

android:paddingRight="8dip" />-->

</LinearLayout>

hostslist.xml:

<?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"

android:background="@drawable/background"

>

<RelativeLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent" >

<LinearLayout

android:id="@+id/linearlayout01"

android:layout\_width="300dp"

android:layout\_height="45dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="52dp"

android:orientation="vertical" >

</LinearLayout>

<ListView

android:id="@android:id/list"

android:layout\_width="fill\_parent"

android:layout\_height="230dp"

android:layout\_alignParentRight="true"

android:layout\_marginTop="107dp"

android:cacheColorHint="#00000000">

</ListView>

</RelativeLayout>

</LinearLayout>

Top 5 ответ

**1**[Raghunandan](http://stackoverflow.com/users/653856/)@2013-04-13 05:23:58

Я хотел бы предложить вам использовать пользовательский элемент управления ListView с пользовательским адаптера.

Пример доступен по ссылке ниже.

http://www.androidhive.info/2012/02/android-custom-listview-with-image-and-text/

Используйте viewholder для плавной прокрутки и производительности.

Вы также можете использовать ленивый списка или универсальный образ загрузчика .

Оба используют кэширование.

http://stackoverflow.com/questions/15621936/whats-lazylist/15622374#15622374 .

Проверьте ниже ссылки для viewholder и загрузки растровых изображений эффективно

http://developer.android.com/training/improving-layouts/smooth-scrolling.html .

http://www.youtube.com/watch?v=wDBM6wVEO70 . Речь идет о viewholder и список характеристик вида.

http://developer.android.com/training/displaying-bitmaps/load-bitmap.html . Загрузка растровых изображений эффективно.

**2**[ishu](http://stackoverflow.com/users/2148099/)@2013-04-13 06:06:54

Сделать проект CustomListview

========= CustomListView.java

package com.example.customlistview;

import java.util.ArrayList;

import android.os.Bundle;

import android.widget.ListView;

import android.app.Activity;

public class CustomListView extends Activity {

ArrayList<Contact> imageArry = new ArrayList<Contact>();

ContactImageAdapter adapter;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

// add image and text in arraylist

imageArry.add(new Contact(R.drawable.facebook, "FaceBook"));

imageArry.add(new Contact(R.drawable.google, "Google"));

imageArry.add(new Contact(R.drawable.ical, "Ical"));

imageArry.add(new Contact(R.drawable.outlook, "Outlook"));

imageArry.add(new Contact(R.drawable.twitter, "Twitter"));

// add data in contact image adapter

adapter = new ContactImageAdapter(this, R.layout.list, imageArry);

ListView dataList = (ListView) findViewById(R.id.list);

dataList.setAdapter(adapter);

}

}

========================== Contact.java

package com.example.customlistview;

public class Contact

{

int image;

String name;

public int getImage() {

return image;

}

public void setImage(int image) {

this.image = image;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Contact(int image, String name) {

super();

this.image = image;

this.name = name;

}

}

=========== ContactImageAdapter.java

package com.example.customlistview;

import java.util.ArrayList;

import android.app.Activity;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ArrayAdapter;

import android.widget.ImageView;

import android.widget.TextView;

public class ContactImageAdapter extends ArrayAdapter<Contact> {

Context context;

int layoutResourceId;

ArrayList<Contact> data=new ArrayList<Contact>();

public ContactImageAdapter(Context context, int layoutResourceId, ArrayList<Contact> data) {

super(context, layoutResourceId, data);

this.layoutResourceId = layoutResourceId;

this.context = context;

this.data = data;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

View row = convertView;

ImageHolder holder = null;

if(row == null)

{

LayoutInflater inflater = ((Activity)context).getLayoutInflater();

row = inflater.inflate(layoutResourceId, parent, false);

holder = new ImageHolder();

holder.txtTitle = (TextView)row.findViewById(R.id.txtTitle);

holder.imgIcon = (ImageView)row.findViewById(R.id.imgIcon);

row.setTag(holder);

}

else

{

holder = (ImageHolder)row.getTag();

}

Contact myImage = data.get(position);

holder.txtTitle.setText(myImage.name);

int outImage=myImage.image;

holder.imgIcon.setImageResource(outImage);

return row;

}

static class ImageHolder

{

ImageView imgIcon;

TextView txtTitle;

}

}

=======main.xml

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<ListView

android:id="@+id/list"

android:layout\_width="0dp"

android:layout\_height="fill\_parent"

android:layout\_weight="0.55" >

</ListView>

</LinearLayout>

============================== list.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="horizontal"

android:padding="10dp" ><ImageView

android:id="@+id/imgIcon"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="0.71"

android:gravity="center\_vertical"/>

<TextView

android:id="@+id/txtTitle"

android:layout\_width="80dp"

android:layout\_height="fill\_parent"

android:gravity="center\_vertical"

android:textSize="14dp"

android:layout\_marginLeft="7dp" />

</LinearLayout>

=================

попробуйте этот код . положить икона в Drwable, которое используется при основной массив активность изображений.

**3**[ishu](http://stackoverflow.com/users/2148099/)@2013-04-13 12:32:33

добавить класс для отложенной загрузки FileCache.java

========

public class FileCache {

private File cacheDir;

public FileCache(Context context){

//Find the dir to save cached images

if (android.os.Environment.getExternalStorageState().equals(android.os.Environment.MEDIA\_MOUNTED))

cacheDir=new File(android.os.Environment.getExternalStorageDirectory(),"LazyList");

else

cacheDir=context.getCacheDir();

if(!cacheDir.exists())

cacheDir.mkdirs();

}

public File getFile(String url){

//I identify images by hashcode. Not a perfect solution, good for the demo.

String filename=String.valueOf(url.hashCode());

//Another possible solution (thanks to grantland)

//String filename = URLEncoder.encode(url);

File f = new File(cacheDir, filename);

return f;

}

public void clear(){

File[] files=cacheDir.listFiles();

if(files==null)

return;

for(File f:files)

f.delete();

}

}

========= ImageLoader.java

public class ImageLoader {

MemoryCache memoryCache=new MemoryCache();

FileCache fileCache;

private Map<ImageView, String> imageViews=Collections.synchronizedMap(new WeakHashMap<ImageView, String>());

ExecutorService executorService;

Handler handler=new Handler();//handler to display images in UI thread

public ImageLoader(Context context){

fileCache=new FileCache(context);

executorService=Executors.newFixedThreadPool(5);

}

final int stub\_id=R.drawable.stub;

public void DisplayImage(String url, ImageView imageView)

{

imageViews.put(imageView, url);

Bitmap bitmap=memoryCache.get(url);

if(bitmap!=null)

imageView.setImageBitmap(bitmap);

else

{

queuePhoto(url, imageView);

imageView.setImageResource(stub\_id);

}

}

private void queuePhoto(String url, ImageView imageView)

{

PhotoToLoad p=new PhotoToLoad(url, imageView);

executorService.submit(new PhotosLoader(p));

}

private Bitmap getBitmap(String url)

{

File f=fileCache.getFile(url);

//from SD cache

Bitmap b = decodeFile(f);

if(b!=null)

return b;

//from web

try {

Bitmap bitmap=null;

URL imageUrl = new URL(url);

HttpURLConnection conn = (HttpURLConnection)imageUrl.openConnection();

conn.setConnectTimeout(30000);

conn.setReadTimeout(30000);

conn.setInstanceFollowRedirects(true);

InputStream is=conn.getInputStream();

OutputStream os = new FileOutputStream(f);

Utils.CopyStream(is, os);

os.close();

conn.disconnect();

bitmap = decodeFile(f);

return bitmap;

} catch (Throwable ex){

ex.printStackTrace();

if(ex instanceof OutOfMemoryError)

memoryCache.clear();

return null;

}

}

//decodes image and scales it to reduce memory consumption

private Bitmap decodeFile(File f){

try {

//decode image size

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

o.inJustDecodeBounds = true;

FileInputStream stream1=new FileInputStream(f);

BitmapFactory.decodeStream(stream1,null,o);

stream1.close();

//Find the correct scale value. It should be the power of 2.

final int REQUIRED\_SIZE=70;

int width\_tmp=o.outWidth, height\_tmp=o.outHeight;

int scale=1;

while(true){

if(width\_tmp/2<REQUIRED\_SIZE || height\_tmp/2<REQUIRED\_SIZE)

break;

width\_tmp/=2;

height\_tmp/=2;

scale\*=2;

}

//decode with inSampleSize

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

o2.inSampleSize=scale;

FileInputStream stream2=new FileInputStream(f);

Bitmap bitmap=BitmapFactory.decodeStream(stream2, null, o2);

stream2.close();

return bitmap;

} catch (FileNotFoundException e) {

}

catch (IOException e) {

e.printStackTrace();

}

return null;

}

//Task for the queue

private class PhotoToLoad

{

public String url;

public ImageView imageView;

public PhotoToLoad(String u, ImageView i){

url=u;

imageView=i;

}

}

class PhotosLoader implements Runnable {

PhotoToLoad photoToLoad;

PhotosLoader(PhotoToLoad photoToLoad){

this.photoToLoad=photoToLoad;

}

public void run() {

try{

if(imageViewReused(photoToLoad))

return;

Bitmap bmp=getBitmap(photoToLoad.url);

memoryCache.put(photoToLoad.url, bmp);

if(imageViewReused(photoToLoad))

return;

BitmapDisplayer bd=new BitmapDisplayer(bmp, photoToLoad);

handler.post(bd);

}catch(Throwable th){

th.printStackTrace();

}

}

}

boolean imageViewReused(PhotoToLoad photoToLoad){

String tag=imageViews.get(photoToLoad.imageView);

if(tag==null || !tag.equals(photoToLoad.url))

return true;

return false;

}

//Used to display bitmap in the UI thread

class BitmapDisplayer implements Runnable

{

Bitmap bitmap;

PhotoToLoad photoToLoad;

public BitmapDisplayer(Bitmap b, PhotoToLoad p){bitmap=b;photoToLoad=p;}

public void run()

{

if(imageViewReused(photoToLoad))

return;

if(bitmap!=null)

photoToLoad.imageView.setImageBitmap(bitmap);

else

photoToLoad.imageView.setImageResource(stub\_id);

}

}

public void clearCache() {

memoryCache.clear();

fileCache.clear();

}

}

====== MemoryCache.java

public class MemoryCache {

private static final String TAG = "MemoryCache";

private Map<String, Bitmap> cache=Collections.synchronizedMap(

new LinkedHashMap<String, Bitmap>(10,1.5f,true));//Last argument true for LRU ordering

private long size=0;//current allocated size

private long limit=1000000;//max memory in bytes

public MemoryCache(){

//use 25% of available heap size

setLimit(Runtime.getRuntime().maxMemory()/4);

}

public void setLimit(long new\_limit){

limit=new\_limit;

Log.i(TAG, "MemoryCache will use up to "+limit/1024./1024.+"MB");

}

public Bitmap get(String id){

try{

if(!cache.containsKey(id))

return null;

//NullPointerException sometimes happen here http://code.google.com/p/osmdroid/issues/detail?id=78

return cache.get(id);

}catch(NullPointerException ex){

ex.printStackTrace();

return null;

}

}

public void put(String id, Bitmap bitmap){

try{

if(cache.containsKey(id))

size-=getSizeInBytes(cache.get(id));

cache.put(id, bitmap);

size+=getSizeInBytes(bitmap);

checkSize();

}catch(Throwable th){

th.printStackTrace();

}

}

private void checkSize() {

Log.i(TAG, "cache size="+size+" length="+cache.size());

if(size>limit){

Iterator<Entry<String, Bitmap>> iter=cache.entrySet().iterator();//least recently accessed item will be the first one iterated

while(iter.hasNext()){

Entry<String, Bitmap> entry=iter.next();

size-=getSizeInBytes(entry.getValue());

iter.remove();

if(size<=limit)

break;

}

Log.i(TAG, "Clean cache. New size "+cache.size());

}

}

public void clear() {

try{

//NullPointerException sometimes happen here http://code.google.com/p/osmdroid/issues/detail?id=78

cache.clear();

size=0;

}catch(NullPointerException ex){

ex.printStackTrace();

}

}

long getSizeInBytes(Bitmap bitmap) {

if(bitmap==null)

return 0;

return bitmap.getRowBytes() \* bitmap.getHeight();

}

}

=====Utils.java

public class Utils {

public static void CopyStream(InputStream is, OutputStream os)

{

final int buffer\_size=1024;

try

{

byte[] bytes=new byte[buffer\_size];

for(;;)

{

int count=is.read(bytes, 0, buffer\_size);

if(count==-1)

break;

os.write(bytes, 0, count);

}

}

catch(Exception ex){}

}

}

==== ContactImageAdapte.java

public class ContactImageAdapter extends ArrayAdapter<Contact>{

Context context;

int layoutResourceId;

public ImageLoader imageLoader;

ArrayList<Contact> data=new ArrayList<Contact>();

public ContactImageAdapter(Context context, int layoutResourceId, ArrayList<Contact> data) {

super(context, layoutResourceId, data);

this.layoutResourceId = layoutResourceId;

this.context = context;

this.data = data;

this.imageLoader=new ImageLoader(context);

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

View row = convertView;

ImageHolder holder = null;

if(row == null)

{

LayoutInflater inflater = ((Activity)context).getLayoutInflater();

row = inflater.inflate(layoutResourceId, parent, false);

holder = new ImageHolder();

holder.txtTitle = (TextView)row.findViewById(R.id.txtTitle);

holder.imgIcon = (ImageView)row.findViewById(R.id.imgIcon);

row.setTag(holder);

}

else

{

holder = (ImageHolder)row.getTag();

}

Contact myImage = data.get(position);

holder.txtTitle.setText(myImage.name);

int outImage=myImage.image;

imageLoader.DisplayImage(listres.ItemImage,holder.ItamImage);

return row;

}

static class ImageHolder

{

ImageView imgIcon;

TextView txtTitle;

}

}