

Losileeya

编程中我们会遇到多少挫折？不放弃，
沙漠尽头必是绿洲，不是天方夜谭

目录视图

个人资料



Losileeya

[关注](#) [发私信](#)

访问：152527次

积分：2783

等级：BLOG > 5

排名：第9367名

原创：92篇

转载：20篇

译文：0篇

评论：198条

博客专栏



Android实战开发

文章：12篇

阅读：30965



Kotlin从语法到安卓开发

文章：5篇

阅读：3622



安卓使用第三方库开速开发

文章：7篇

阅读：16211



安卓特效开发

文章：15篇

阅读：36269

文章分类

android (80)

ios (2)

深度学习代码专栏 攒课--我的学习我做主 开启你的知识管理，知识库个人图谱上线

Android图片处理之Glide使用大全

标签：Glide 图片缓存 网络处理 2016-01-27 20:10 5107人阅读 评论

分类： android (79)

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?) [+]

听说Glide是一个牛逼的图片加载库，作者是bumptech，然后这个库被广泛的应用在谷歌的官方app里，也就是谷歌支持我库，所以呢就需要响应谷歌的要求，就像网络访问我们用volley，然后大部分安卓开发者都弃用eclipse转战Android studio一样太多了，显然也是各有特点自己适合才是最好的，所以就来学习下这个Glide的图片库，有了android studio开发真是太方便了



Android Glide是一个开源的图片加载和缓存处理的第三方框架。使用Android的Glide和Android的F似，基本上会用毕加索的也就会使用这个东东了，个人感觉比Android Picasso好用。Android Glide了缓存策略，使得开发者摆脱Android图片加载的琐碎事务，专注逻辑业务的代码。Android Glide使单明晰的代码，即可完成大多数图片从网络（或者本地）加载、显示的功能需求。

一.为什么要使用Glide

以前还没有出Glide、Picasso和Fresco的时候我们最常用的是使用老牌的图片框架universalImage universalImageLoader配置相对麻烦，虽然提供了各种配置，但是没有实践过，根本不知道如何配置多，总体感觉就是配置太多了，使用起来也还算比较方便吧，总的来说使用eclipse开发的时候这个相非常的轻松，确实也很少出现oom，这里要给它点个赞，然后也是国产app使用最多的一个图片加载 volley去发送请求，处理图片时虽然它也带了ImageLoader和NetworkImageView但是很容易出现o volley+universalImageLoader。

随着Glide、Picasso和Fresco三个图片加载库的出现，我们就更加喜欢用更方便的图片框架了 universalImageLoader虽然好但是配置多，但是我们使用加载过多的大图片时，Picasso（毕加索）大，所以如果是要加载很多图片那么毕加索可能会OutOfMemoryError的发生，至于Fresco是一个非框架，支持webps格式（和jpg一样都是有损压缩格式，webps相同质量图片更节省空间），支持渐进定制image的各种属性，支持多图请求和图片复用，并支持手势缩放和旋转等等（这里不做比较）。

所以具体来看Glide和Picasso，Glide加载图像以及磁盘缓存的方式都要优于Picasso，速更有利于减少OutOfMemoryError的发生，GIF动画是Glide的杀手锏。不过Picasso的图片质量更高。是另一个picasso，从picasso转移到glide相对改动较少，还有一点就是这个项目是google在维护，我任，相比较universalImageLoader，glide可以支持gif和短视频，后期也需要用到，这里不得不谈一了，glide图片缓存默认使用RGB565相当于ARGB8888可以节省不少的空间，支持与activity，fragr

html5 (2)

Kotlin (5)

AngularJs (0)

React-native (2)

java (10)

开发环境 (13)

sina 微博

微博



zilianliuxue 其他

加关注

我的个人站点

<https://zilianliuxue.github.io/>

评论排行

高仿淘宝购物车分分钟让你集成

(32)

android高仿京东快报 (垂直...

(17)

安卓webView实现长按二维码...

(14)

安卓listview实现下拉刷新上...

(13)

android studio一键生成快速...

(8)

安卓图片加载之使用universal...

(8)

安卓的个性化彩色二维码的完...

(7)

手把手教你炫酷慕课网视频启...

(7)

android实现文字渐变效果和...

(6)

自定义多状态高仿应用下载百...

(5)

阅读排行

安卓转战React-Native之win...

(7269)

安卓实战开发之SQLite从简单...

(6477)

安卓转战React-Native之签名...

(6141)

安卓实战开发之JNI入门及高...

(6134)

安卓实战开发之CardViews的s...

(5977)

Android图片处理之Glide使...

(5103)

android studio一键生成快速...

(4318)

安卓CountDownTimer实现...

(4312)

自定义多状态高仿应用下载百...

(3857)

高仿淘宝购物车分分钟让你集成

(3756)

最新评论

安卓开发实战之app之版本更新升级(Dow... qq_34864929 : 这是我想要的

android高仿京东快报 (垂直循环滚动新... Losileeya : @qq_30108311:你要那样做的

话,得自己改了,基本这些控件属于定制版

android高仿京东快报 (垂直循环滚动新... Android丶Ren : 可以,不同的实现方法。

安卓实战之如何快速搭建app架构 Losileeya : @u012764110:哈哈,都差不多是这些套路

安卓实战之如何快速搭建app架构 苹果树上的女孩 : mark,跟你用的差不多,

现在就网络请求库还没封装好

安卓实战之如何快速搭建app架构 sxtchengxuyuan : mark写的挺好的

Android Studio 多渠道打包 奥特曼超人 : 已推荐收录,建议去掉滚动的

~~

高仿淘宝购物车分分钟让你集成 Losileeya : @sinat_31238015:解决了

高仿淘宝购物车分分钟让你集成 sinat_31238015 : @u013278099:项目需

要啊~~555我在git上提交了issues 求解决

高仿淘宝购物车分分钟让你集成

命周期的联动,更智能管理图片请求当然还有其他的扩展更多可以看 [glide](#)介绍 当然, [glide](#)的方法类 [universalImageLoader](#)多了1000多个,遇到64k问题的会比较关注这个。

不说了,反正这些图片加载框架都是相当 的牛逼,你就看着用好了。

二. [Glide](#)的使用

上一篇我们讲了毕加索 《[安卓图片处理Picasso的解析使用](#)》,对着上一篇你就会知道Glide和Pic多。

我们现在就来使用吧,依旧是你想用[glide-3.6.1.jar](#)就去下载,放在libs下导入就好了,如果你使用简单了

你有github账号就更加好了,没有那么你就落后了,赶紧去申请一个,没有也可以去github去doi直接 [Git clone https://github.com/bumptech/glide.git](#)

```
repositories {
    mavenCentral()
}

dependencies {
    compile 'com.github.bumptech.glide:glide:3.6.1'
    compile 'com.android.support:support-v4:23.1.1'
}
```

从这里你发现了好像是需要V4包,好像Picasso不要

Or Maven:

```
<dependency>
    <groupId>com.github.bumptech.glide</groupId>
    <artifactId>glide</artifactId>
    <version>3.6.1</version>
</dependency>
<dependency>
    <groupId>com.google.android</groupId>
    <artifactId>support-v4</artifactId>
    <version>r7</version>
</dependency>
```

按着说明文档一步步操作就好了:

说明里面也介绍了去代码混淆:在proguard-project.txt或proguard.cfg

```
-keep public class * implements com.bumptech.glide.module.GlideModule
-keep public enum com.bumptech.glide.load.resource.bitmap.ImageHeaderParser$** {
    **[] $VALUES;
    public *;
}
```

准备工作完了,那就是你要去代码使用的时候了:

概括一句话也是

Glide在 (with) 当前上下文中加载 (load) 一张图片到 (into) imageView控件

现在直接上代码,因为和毕加索的使用差不多:

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    @InjectView(R.id.listview)
    ListView listView;

    @InjectView(R.id.single_view_asset)
    ImageView singleViewAsset;

    @InjectView(R.id.single_view_res)
    ImageView singleViewRes;

    private String[] images;

    private ListViewAdapter mAdapter;
```

Losileeya : @sinat_31238015:不建议在Ex
pandableListView做购物车时添加尾部

github



<https://github.com/zilianliuxue/>

欢迎Follow、Fork、Star



nliuxue.github.io/



@Override

```
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.content_main);

    ButterKnife.inject(this);

    //加载本地图片

    initLocal();

    //加载网络图片

    initNet();

}

private void initNet() {

    getImageUrls();

    listview.setAdapter(mAdapter);

}

private void getImageUrls() {

    images= new String[] {

        "http://img.my.csdn.net/uploads/201407/26/1406383299_1976.jpg",
        "http://img.my.csdn.net/uploads/201407/26/1406383291_6518.jpg",
        "http://img.my.csdn.net/uploads/201407/26/1406383291_8239.jpg",
        "http://img.my.csdn.net/uploads/201407/26/1406383290_9329.jpg",
        "http://img.my.csdn.net/uploads/201407/26/1406383290_1042.jpg",
        "http://img.my.csdn.net/uploads/201407/26/1406383275_3977.jpg",
        "http://img.my.csdn.net/uploads/201407/26/1406383265_8550.jpg",
        "http://img.my.csdn.net/uploads/201407/26/1406383264_3954.jpg",
        "http://img.my.csdn.net/uploads/201407/26/1406383264_4787.jpg",
        "http://img.my.csdn.net/uploads/201407/26/1406383264_8243.jpg",
        "http://img.my.csdn.net/uploads/201407/26/1406383248_3693.jpg",

    };

}

private void initLocal() {

    //加载资源图片

    Glide.with(this).load(R.drawable.alipay).into(singleViewRes);

    //加载资产目录图片

    Glide.with(this).load("file:///android_asset/heart.png").into(singleViewAsset);

    //加载sd卡图片文件

    // Glide.with(this).load(new File("XXX")).into(iv_picasso);

}
```

```
}
```

再看适配器代码：ListViewAdapter.java

```
/**
 * Created by test1 on 2016/1/27.
 */
public class ListViewAdapter extends BaseAdapter{

    private String images[];
    private Activity mActivity;
    private LayoutInflater mInflat;

    public ListViewAdapter(String[] images, Activity mActivity) {

        this.images = images;

        mInflat=LayoutInflater.from(mActivity);
    }

    @Override
    public int getCount() {

        return images.length;
    }

    @Override
    public Object getItem(int position) {

        return images[position];
    }

    @Override
    public long getItemId(int position) {

        return position;
    }

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

        if(convertView==null){

            convertView= mInflat.inflate(R.layout.list_item, null, false);
            ImageView iv= (ImageView) convertView.findViewById(R.id.iv);
            Glide.with(mActivity)

                .load(images[position])

                .centerCrop()

                .placeholder(R.mipmap.ic_launcher)

                .crossFade()
        }
    }
}
```

nliuxue.github.io/

```

        .into(iv);
    }

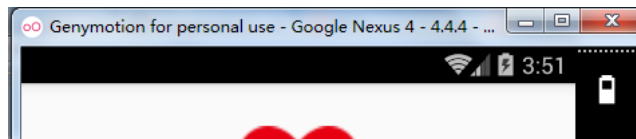
    return convertView;
}

```

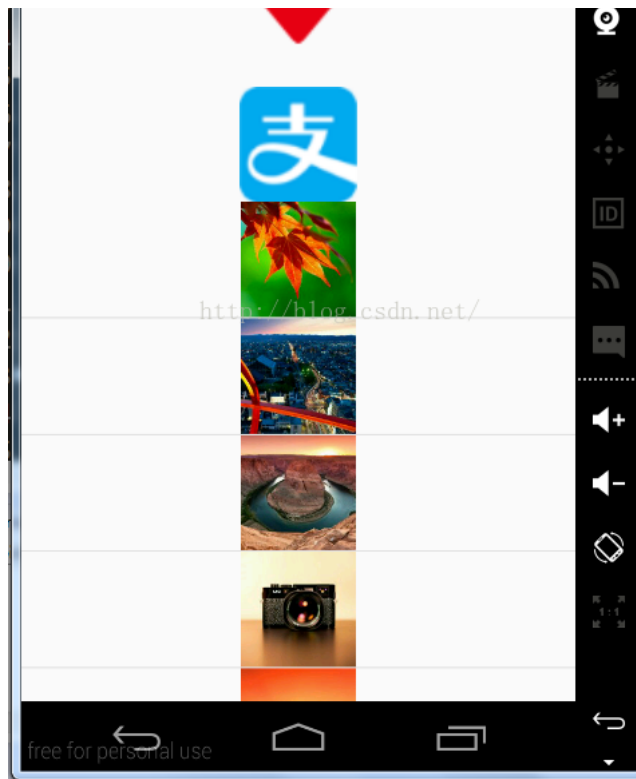
还有很多方法：像crossFade()默认动画，fitCenter()，placeholder()占位图，error()错误图，空图

这里的Glide.with() 不仅可以传 Context ,还可以传Activity 和 Fragment，因为图片加载会和Act
命周期保持一致，比如Paused状态在暂停加载，在Resumed的时候又自动重新加载。

效果看图：



nliuxue.github.io/



这就是Glide的简单使用。

三. Glide的其他使用

如果想用Okhttp做协议栈可以直接添加 glide-okhttp-integration-1.3.1.jar，并且在Applicati
onCreate 方法注册就可以使用okhttp做为协议栈，Volley也是同理；

Glide.get(this).register(GlideUrl.class,InputStream.class,new OkHttpUrlLoader.Factory(new
Glide默认的图片格式是RGB_565所以，要想使用ARGB_8888模式，又想Okhttp做协议栈，就不
integration-1.3.1.jar，直接抽取里面的 OkHttpGlideModule 类
加**builder.setDecodeFormat(DecodeFormat.PREFER_ARGB_8888);**

用起来很简单，说下注意点吧：

- 1.placeholder() 占位图或者失败图都可以直接使用R.color.white 颜色来做；
- 2.如果加载不出来图片的话可以试试设置下图片的宽高；
- 3.图片缓存：图片的加载并不是全缓存，而是使用的imageview的大小来的，如果想要全缓存的就

- `DiskCacheStrategy.NONE` 什么都不缓存
- `DiskCacheStrategy.SOURCE` 仅仅只缓存原来的全分辨率的图像
- `DiskCacheStrategy.RESULT` 仅仅缓存最终的图像，即降低分辨率后的（或者是转换后的）
- `DiskCacheStrategy.ALL` 缓存所有版本的图像（默认行为）

```
.diskCacheStrategy(DiskCacheStrategy.ALL)
```

4. 图片加载背景会变成绿色，加载jpg图片会出现的BUG，github上给出解决方案

```
Glide.with(this).load(url).diskCacheStrategy(DiskCacheStrategy.SOURCE).into(imageView); 或者
```

```
Glide.with(this).fromResource().asBitmap().encoder(newBitmapEncoder(Bitmap.CompressFormat.PNG,100  
stimg).into(imageView);
```

nliuxue.github.io/

```
Glide.with(context).load(imageUrl).asBitmap().fitCenter().diskCacheStrategy(DiskCacheStrategy.SOURCE)

.placeholder(R.drawable.shape_glide_round_place).error(R.drawable.no_face_circle)

.into(newBitmapImageViewTarget(imageView) {

@Override

protected voidsetResource(Bitmap resource) {

RoundedBitmapDrawable circularBitmapDrawable =

RoundedBitmapDrawableFactory.create(context.getResources(),resource);

circularBitmapDrawable.setCircular(true);

imageView.setImageDrawable(circularBitmapDrawable);

}

});
```

5. 动画 `crossFade()`

`crossFade()` 方法还有另外重载方法 `.crossFade(int duration)`。如果你想要去减慢（或加快）动画，随时可以这个方法。动画默认的持续时间是 300毫秒。

如果你想直接显示图片而没有任何淡入淡出效果，在 Glide 的建造者中调用 `.dontAnimate()`。

6. 清除缓存：Glide.get(this).clearMemory();

Glide.get(this).clearDiskCache(); 需要在子线程执行

7. 加载暂时不支持显示进度，可以用占位图来显示，把占位图替换成帧动画，还有错误图片，图片缩

```
[java] view plain copy print ? C P
01. Glide
02. .with(context)
03. .load(imgurl)
04. .placeholder(R.mipmap.ic_launcher) // can also be a drawable
05. .into(imageViewPlaceholder);
```

当加载图片失败时，通过`error(Drawable drawable)`方法设置加载失败后的图片显示：

```
[java] view plain copy print ? C P
01. Glide
02. .with(context)
```

```

03.         .load(imgurl)
04.         .error(R.mipmap.future_studio_launcher) // will be displayed if the image cannot be loaded
05.         .into(imageView);

```

图片的缩放，centerCrop()和fitCenter():

```

[java] view plain copy print ?
01.     Glide.with(MainActivity.this)
02.             .load(imgurl)
03.             .fitCenter()//缩放类型
04.             .into(imageView);

```

1. 使用centerCrop是利用图片图填充ImageView设置的大小，如果ImageView的Height是match_parent则图片就会被拉伸
2. 使用fitCenter即缩放图像让图像都测量出来等于或小于 ImageView 的边界范围该图像将会完全显示，但可能不

图片自定义显示大小：

```

[java] view plain copy print ?
01.     Glide
...
05.         .into(imageViewResize);

```

nliuxue.github.io/

8.glide获取bitmap动态设置图片大小,通过viewTarget设置自定义view的图片

1.simpleTarget获取bitmap根据bitmap设定图片显示尺寸：(这里适配屏幕宽度，高度按比例拉伸)

```

[java] view plain copy print ?
01.     final ImageView img = new ImageView(context);
02.     LayoutParams lp = img_content.getLayoutParams();
03.     lp.width=400;
04.     lp.height=200;
05.     img.setLayoutParams(lp);
06.     Glide.with(ShopActivity.this)
07.         .load("http://dn-qpos-box.qbox.me/hqs.jpg").asBitmap()
08.         .into(new SimpleTarget<Bitmap>() {
09.             @Override
10.             public void onResourceReady(Bitmap bitmap,
11.                 GlideAnimation<? super Bitmap> glideAnimation) {
12.                 int width = bitmap.getWidth();
13.                 int height = bitmap.getHeight();
14.                 LayoutParams lp = img.getLayoutParams();
15.                 lp.width = SysEnv.SCREEN_WIDTH;
16.                 float tempHeight=height * ((float)lp.width / width);
17.                 lp.height =(int)tempHeight ;
18.                 img.setLayoutParams(lp);
19.                 img.setImageBitmap(bitmap);
20.                 img_content.addView(img);
21.             }
22.         });

```

2.simpleTarget获取bitmap指定尺寸：

```

[java] view plain copy print ?
01.     private SimpleTarget target2 = new SimpleTarget<Bitmap>( 250, 250 ) {
02.         @Override
03.         public void onResourceReady(Bitmap bitmap, GlideAnimation glideAnimation) {
04.             imageView2.setImageBitmap( bitmap );
05.         }
06.     };
07.     private void loadImageSimpleTargetApplicationContext() {
08.         Glide
09.             .with( context.getApplicationContext() ) // safer!
10.             .load( eatFoodyImages[1] )
11.             .asBitmap()
12.             .into( target2 );
13.     }

```

3.viewTarget

假设你有一个 Custom View。Glide 并不支持加载图片到自定义 view 中，因为并没有方法知道图片应该在哪里被加载，所以用 ViewTarget 更容易实现。

```

[java] view plain copy print ?
01.     public class FutureStudioView extends FrameLayout {

```

```

02.     ImageView iv;
03.     TextView tv;
04.
05.     public void initialize(Context context) {
06.         inflate( context, R.layout.custom_view_futurestudio, this );
07.
08.         iv = (ImageView) findViewById( R.id.custom_view_image );
09.         tv = (TextView) findViewById( R.id.custom_view_text );
10.     }
11.
12.     public FutureStudioView(Context context, AttributeSet attrs) {
13.         super( context, attrs );
14.         initialize( context );
15.     }
16.
17.     public FutureStudioView(Context context, AttributeSet attrs, int defStyleAttr) {
18.         super( context, attrs, defStyleAttr );
19.         initialize( context );
20.     }
21.
22.     public void setImage(Drawable drawable) {
23.         iv = (ImageView) findViewById( R.id.custom_view_image );
24.
25.         iv.setImageDrawable( drawable );
26.     }

```

nliuxue.github.io/

你不能使用常规的 Glide 的方法 `.into()`，因为我们的自定义 view 并不继承自 `ImageView`。因此，我们必须创
并用 `.into()` 方法：

```

[java] view plain copy print ?
01. private void loadImageViewTarget() {
02.     FutureStudioView customView = (FutureStudioView) findViewById( R.id.custom_view );
03.
04.     viewTarget = new ViewTarget<FutureStudioView, GlideDrawable>( customView ) {
05.         @Override
06.         public void onResourceReady(GlideDrawable resource, GlideAnimation<? super GlideDrawable> glideAnimati
07.             this.view.setImage( resource.getCurrent() );
08.         }
09.     };
10.
11.     Glide
12.         .with( context.getApplicationContext() ) // safer!
13.         .load( imgUrl )
14.         .into( viewTarget );
15. }

```

9.显示gif动画

```

[java] view plain copy print ?
01. Glide
02.     .with( context )
03.     .load( gifUrl )
04.     .asGif() //判断加载的url资源是否为gif格式的资源
05.     .error( R.drawable.full_cake )
06.     .into( imageViewGif );

```

10.显示本地视频

```

[java] view plain copy print ?
01. String filePath = "/storage/emulated/0/Pictures/example_video.mp4";
02. Glide
03.     .with( context )
04.     .load( Uri.fromFile( new File( filePath ) ) )
05.     .into( imageViewGifAsBitmap );

```

11.图片加载优先级

- `Priority.LOW`
- `Priority.NORMAL`
- `Priority.HIGH`
- `Priority.IMMEDIATE`

```

[java] view plain copy print ?

```



```
01. private void loadImageWithPriority() {
02.     Glide
03.         .with( context )
04.         .load( imageUrl )
05.         .priority( Priority.HIGH )//优先级
06.         .into( imageView );
07. }
```

12.集成网络栈(okHttp,Volley):

```
[java] view plain copy print ?
01. dependencies {
02.     // your other dependencies
03.     // ...
04.     // Glide
05.     compile 'com.github.bumptech.glide:glide:3.6.1'
06.     // Glide's OkHttp Integration
07.     compile 'com.github.bumptech.glide:okhttp-integration:1.3.1@aar'
08.     compile 'com.squareup.okhttp:okhttp:2.5.0'
09. }
10. dependencies {
11.     // your other dependencies
12.     // ...
16.     compile 'com.github.bumptech.glide:volley-integration:1.3.1@aar'
17.     compile 'com.mcxiaoke.volley:library:1.0.8'
18. }
```

nliuxue.github.io/

四.如何调试Glide加载图片

与其他图片加载库不同，在Glide加载图片的过程中默认是没有任何log输出的。这样使得加载失败的原因难以调试。为了在异常发生时可以看到它们，你可以打开Glide中处理所有媒体加载响应的类GenericRequest的log开关。很简单，在命令

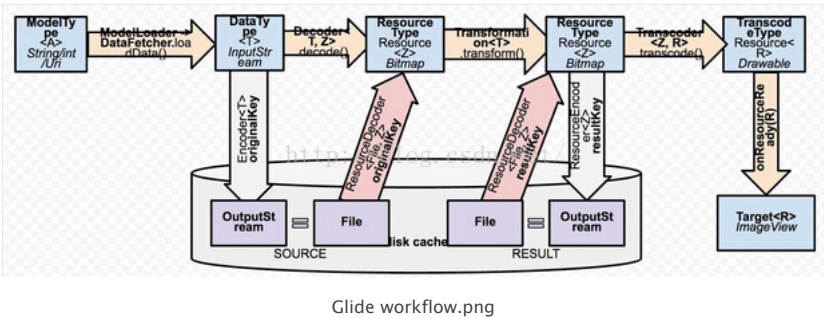
```
adb shell setprop log.tag.GenericRequest DEBUG
```

如果你将DEBUG替换为VERBOSE，还可以看到详细的请求时间日志。

如果你想禁用log输出，执行：

```
adb shell setprop log.tag.GenericRequest ERROR
```

调试 workflow



为了查看Glide内部引擎是何时、如何加载图片的，你可以启用这些log:

```
adb shell setprop log.tag.Engine VERBOSE
adb shell setprop log.tag.EngineJob VERBOSE
adb shell setprop log.tag.DecodeJob VERBOSE
```

请求监听器

虽然启动调试日志很简单，但前提是你访问到设备。为了完善Glide已存在或更复杂的错误日RequestListener类。当加载请求失败时onException()方法就会被调用，并给出造成失败的异常或从它获取到的数据中解码出任何有用的东西时。你可以通过listener() API为每一个请求添加一个监

确保onException()方法的返回值为false，避免覆盖Glide默认的错误处理（比如加载失败的错误图

这里有一个实现快速调试的列子：

```
// 示例: .listener(new LoggingListener<String, GlideDrawable>())
public class LoggingListener<T, R> implements RequestListener<T, R> {
    @Override public boolean onException(Exception e, Object model, Target target, boolean isFirstResource) {
        android.util.Log.d("GLIDE", String.format(Locale.ROOT,
            "onException(%s, %s, %s, %s)", e, model, target, isFirstResource), e);
        return false;
    }
    @Override public boolean onResourceReady(Object resource, Object model, Target target, boolean isFromMemoryCache,
    e) {
        android.util.Log.d("GLIDE", String.format(Locale.ROOT,
            "onResourceReady(%s, %s, %s, %s)", resource, model, target, isFromMemoryCache, isFirstResource));
        return false;
    }
}
```

确保在发布应用时移除掉所有的调试log!

更多的log指令

```
cd .../android-sdk/platform-tools
adb shell setprop log.tagAnimatedGifEncoder VERBOSE
adb shell setprop log.tagAssetUriFetcher VERBOSE
adb shell setprop log.tagBitmapEncoder VERBOSE
adb shell setprop log.tagBufferedIs VERBOSE
adb shell setprop log.tagByteArrayPool VERBOSE
adb shell setprop log.tagCacheLoader VERBOSE
adb shell setprop log.tagContentLengthStream VERBOSE
adb shell setprop log.tagDecodeJob VERBOSE
adb shell setprop log.tagDiskLruCacheWrapper VERBOSE
adb shell setprop log.tagDownsampler VERBOSE
adb shell setprop log.tagEngine VERBOSE
adb shell setprop log.tagEngineRunnable VERBOSE
adb shell setprop log.tagGenericRequest VERBOSE
adb shell setprop log.tagGifDecoder VERBOSE
adb shell setprop log.tagGifEncoder VERBOSE
adb shell setprop log.tagGifHeaderParser VERBOSE
adb shell setprop log.tagGifResourceDecoder VERBOSE
adb shell setprop log.tagGlide VERBOSE
adb shell setprop log.tagImageHeaderParser VERBOSE
adb shell setprop log.tagImageVideoDecoder VERBOSE
adb shell setprop log.tagIVML VERBOSE
adb shell setprop log.tagLocalUriFetcher VERBOSE
adb shell setprop log.tagLruBitmapPool VERBOSE
adb shell setprop log.tagMediaStoreThumbFetcher VERBOSE
adb shell setprop log.tagMemorySizeCalculator VERBOSE
adb shell setprop log.tagPreFillRunner VERBOSE
adb shell setprop log.tagResourceLoader VERBOSE
adb shell setprop log.tagRMRetriever VERBOSE
adb shell setprop log.tagStreamEncoder VERBOSE
adb shell setprop log.tagTransformationUtils VERBOSE
```

调试部分参考<http://www.jianshu.com/p/9bd6efca8724>。好了写到这里，很多方法可以直接去看最后提供一个很牛逼的blog:[Glide Series Overview](#)
终极网站[glide全套学习推荐](#)http://mrfu.me/2016/02/27/Glide_Getting_Started/

顶 踩
4 1

- 上一篇 Android签名机制之---签名过程详解
- 下一篇 安卓6.0权限处理在项目中的实践

我的同类文章

android (79)

- 安卓开发实战之app之版本更新升级(D... 2016-09-28 阅读 567
 - 安卓IPC之aidl使用 (三) ---System ai... 2016-09-04 阅读 729
 - 安卓IPC之aidl使用 (一) --aidl常见使用 2016-09-04 阅读 383
 - android 打包上线忘记password 和 ali... 2016-08-17 阅读 595
 - 安卓实战开发之JNI从小白到伪老白深... 2016-07-23 阅读 1963
 - 安卓实战开发之把arr替换为library给e... 2016-07-04 阅读 1009
- 安卓实践开发之MVP一步步实现到高级... 2016-09-12
 - 安卓IPC之aidl使用 (二) ---aidl本地... 2016-09-04
 - 自己动手做一个adb的wifi连接及adb... 2016-08-21
 - 安卓实践开发之JNI再深入了解 2016-08-06
 - 安卓实战开发之JNI入门及高效的配置... 2016-07-17

参考知识库



Git知识库
934 关注 | 391 收录



Android知识库
12674 关注 | 1190 收录



Java EE
1655 关注

nliuxue.github.io/

猜你在找

- Android网络框架Volley项目实战

Android应用的调试

Android APP开发之真机调试环境...

Android底层技术: Linux驱动框...

Android开源项目实践之UI篇
- android_基础_图片处理_bitmapc...

Android中图片处理库Universal----

Java图片处理方法大全

j2me 图片处理大全

网上在线图片处理彩字生成大全



金秋十月，感恩惠馈

《基于Dubbo的分布式系统架构实战》与《微服务架构的分布式事务解决方案》

套餐组合限时优惠火爆进行中

查看评论



sinat_28263953
很详细，学习了

1楼 20

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- 全部主题

IE10

HTML

Cassandra

Compuware

Cloud Foundry
- Hadoop

Eclipse

SDK

CloudStack

大数据
- AWS

CRM

IIS

FTC

aptech
- 移动游戏

JavaScript

Fedora

coremail

Perl
- Java

数据库

XML

LBS

Unity
- Android

Ubuntu

Unity

Splashtop

Tornado
- iOS

NFC

WAP

CouchBase

Ruby
- Swift

jQuery

UML

云计算

Hibernate
- 智能硬件

BI

components

iOS6

ThinkPHP
- Docker

HTML5

Windows Mobile

Rackspace

HBase
- OpenStack

Spring

Web App

Pure

Solr
- VPI

Apa

Rai

S

An

nliuxue.github.io/