# **[Android图片缓存之初识Glide](http://www.cnblogs.com/whoislcj/p/5558168.html)**

### **Glide特点**

##### **使用简单**

##### **可配置度高，自适应程度高**

##### **支持常见图片格式 Jpg png gif webp**

##### **支持多种数据源  网络、本地、资源、Assets 等**

##### **高效缓存策略    支持Memory和Disk图片缓存 默认Bitmap格式采用RGB\_565内存使用至少减少一半**

##### **生命周期集成   根据Activity/Fragment生命周期自动管理请求**

##### **高效处理Bitmap  使用Bitmap Pool使Bitmap复用，主动调用recycle回收需要回收的Bitmap，减小系统回收压力**

### **Glide简单使用**

#### **1.）添加引用 build.gradle 中添加配置**

compile 'com.github.bumptech.glide:glide:3.7.0'

#### **2.）设置绑定生命周期**

我们可以更加高效的使用Glide提供的方式进行绑定，这样可以更好的让加载图片的请求的生命周期动态管理起来

Glide.with(Context context);// 绑定Context

Glide.with(Activity activity);// 绑定Activity

Glide.with(FragmentActivity activity);// 绑定FragmentActivity

Glide.with(Fragment fragment);// 绑定Fragment

#### **3. ）简单的加载图片实例**

Glide.with(this).load(imageUrl).into(imageView);

#### **4.）设置加载中以及加载失败图片**

api里面对placeholder()、error()函数中有多态实现 用的时候可以具体的熟悉一下

Glide.with(this).load(imageUrl).placeholder(R.mipmap.ic\_launcher).error(R.mipmap.ic\_launcher).into(imageView);

#### **5.）设置跳过内存缓存**

Glide.with(this).load(imageUrl).skipMemoryCache(true).into(imageView);

#### **6.）设置下载优先级**

Glide.with(this).load(imageUrl).priority(Priority.NORMAL).into(imageView);

#### **7.）设置缓存策略**

Glide.with(this).load(imageUrl).diskCacheStrategy(DiskCacheStrategy.ALL).into(imageView);

策略解说：

all:缓存源资源和转换后的资源

none:不作任何磁盘缓存

source:缓存源资源

result：缓存转换后的资源

#### **8.）设置加载动画**

api也提供了几个常用的动画：比如crossFade()

Glide.with(this).load(imageUrl).animate(R.anim.item\_alpha\_in).into(imageView);

R.anim.item\_alpha\_in

[IMG_256](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

<?xml version="1.0" encoding="utf-8"?><set xmlns:android="http://schemas.android.com/apk/res/android">

<alpha

android:duration="500"

android:fromAlpha="0.0"

android:toAlpha="1.0"/></set>

[IMG_257](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

#### **9.）设置缩略图支持**

这样会先加载缩略图 然后在加载全图

Glide.with(this).load(imageUrl).thumbnail(0.1f).into(imageView);

#### **10.）设置加载尺寸**

Glide.with(this).load(imageUrl).override(800, 800).into(imageView);

#### **11.）设置动态转换**

Glide.with(this).load(imageUrl).centerCrop().into(imageView);

    api提供了比如：centerCrop()、fitCenter()等函数也可以通过自定义Transformation，举例说明：比如一个人圆角转化器

[IMG_258](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

public class GlideRoundTransform extends BitmapTransformation {

private float radius = 0f;

public GlideRoundTransform(Context context) {

this(context, 4);

}

public GlideRoundTransform(Context context, int dp) {

super(context);

this.radius = Resources.getSystem().getDisplayMetrics().density \* dp;

}

@Override

protected Bitmap transform(BitmapPool pool, Bitmap toTransform, int outWidth, int outHeight) {

return roundCrop(pool, toTransform);

}

private Bitmap roundCrop(BitmapPool pool, Bitmap source) {

if (source == null) return null;

Bitmap result = pool.get(source.getWidth(), source.getHeight(), Bitmap.Config.ARGB\_8888);

if (result == null) {

result = Bitmap.createBitmap(source.getWidth(), source.getHeight(), Bitmap.Config.ARGB\_8888);

}

Canvas canvas = new Canvas(result);

Paint paint = new Paint();

paint.setShader(new BitmapShader(source, BitmapShader.TileMode.CLAMP, BitmapShader.TileMode.CLAMP));

paint.setAntiAlias(true);

RectF rectF = new RectF(0f, 0f, source.getWidth(), source.getHeight());

canvas.drawRoundRect(rectF, radius, radius, paint);

return result;

}

@Override

public String getId() {

return getClass().getName() + Math.round(radius);

}

}

[IMG_259](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

具体使用

Glide.with(this).load(imageUrl).transform(new GlideRoundTransform(this)).into(imageView);

#### **12.）设置要加载的内容**

项目中有很多需要先下载图片然后再做一些合成的功能，比如项目中出现的图文混排，该如何实现目标下

[IMG_260](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

Glide.with(this).load(imageUrl).centerCrop().into(new SimpleTarget<GlideDrawable>() {

@Override

public void onResourceReady(GlideDrawable resource, GlideAnimation<? super GlideDrawable> glideAnimation) {

imageView.setImageDrawable(resource);

}

});

[IMG_261](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

#### **13 .）设置监听请求接口**

[IMG_262](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

Glide.with(this).load(imageUrl).listener(new RequestListener<String, GlideDrawable>() {

@Override

public boolean onException(Exception e, String model, Target<GlideDrawable> target, boolean isFirstResource) {

return false;

}

@Override

public boolean onResourceReady(GlideDrawable resource, String model, Target<GlideDrawable> target, boolean isFromMemoryCache, boolean isFirstResource) {

//imageView.setImageDrawable(resource);

return false;

}

}).into(imageView);

[IMG_263](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

设置监听的用处 可以用于监控请求发生错误来源，以及图片来源 是内存还是磁盘

#### **15.)设置动态GIF加载方式**

Glide.with(this).load(imageUrl).asBitmap().into(imageView);//显示gif静态图片

Glide.with(this).load(imageUrl).asGif().into(imageView);//显示gif动态图片

#### **16.）缓存的动态清理**

Glide.get(this).clearDiskCache();//清理磁盘缓存 需要在子线程中执行

Glide.get(this).clearMemory();//清理内存缓存 可以在UI主线程中进行