# **[Android图片缓存之Glide进阶](http://www.cnblogs.com/whoislcj/p/5565012.html)**

#### **GlideModule使用：**

       GlideModule 是一个抽象方法，全局改变 Glide 行为的一个方式，通过全局GlideModule 配置Glide，用GlideBuilder设置选项，用Glide注册ModelLoader等。

#### **1.）自定义一个GlideModule**

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

public class MyGlideModule implements GlideModule {

@Override public void applyOptions(Context context, GlideBuilder builder) {

// Apply options to the builder here. }

@Override public void registerComponents(Context context, Glide glide) {

// register ModelLoaders here. }

}

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

#### **2.）AndroidManifest.xml注册**

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

<manifest ...>

<!-- ... permissions -->

<application ...>

<meta-data

android:name="com.mypackage.MyGlideModule"

android:value="GlideModule" />

<!-- ... activities and other components -->

</application></manifest>

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

#### **3.)添加混淆处理**

-keepnames class com.mypackage.MyGlideModule

# or more generally:

#-keep public class \* implements com.bumptech.glide.module.GlideModule

#### **4.)多个GlideModule冲突问题**

   GlideModule不能指定调用顺序，所以应该避免不同的GlideModule之间有冲突的选项设置，可以考虑将所有的设置都放到一个GlideModule里面，或者排除掉某个manifest文件的某个Module，代码如下：

<meta-data android:name=”com.mypackage.MyGlideModule” tools:node=”remove” />

### **GlideBuilder设置选项：**

#### **1.）设置Glide内存缓存大小**

int maxMemory = (int) Runtime.getRuntime().maxMemory();//获取系统分配给应用的总内存大小

int memoryCacheSize = maxMemory / 8;//设置图片内存缓存占用八分之一

//设置内存缓存大小

builder.setMemoryCache(new LruResourceCache(memoryCacheSize));

  获取默认的内存使用计算函数

MemorySizeCalculator calculator = new MemorySizeCalculator(context); int defaultMemoryCacheSize = calculator.getMemoryCacheSize(); int defaultBitmapPoolSize = calculator.getBitmapPoolSize();

#### **2.）设置Glide磁盘缓存大小**

File cacheDir = context.getExternalCacheDir();//指定的是数据的缓存地址

int diskCacheSize = 1024 \* 1024 \* 30;//最多可以缓存多少字节的数据

//设置磁盘缓存大小

builder.setDiskCache(new DiskLruCacheFactory(cacheDir.getPath(), "glide", diskCacheSize));

 也可以通过如下两种方式

//存放在data/data/xxxx/cache/ builder.setDiskCache(new InternalCacheDiskCacheFactory(context, "glide", diskCacheSize));

//存放在外置文件浏览器 builder.setDiskCache(new ExternalCacheDiskCacheFactory(context, "glide", diskCacheSize));

#### **3.)设置图片解码格式**

//设置图片解码格式

builder.setDecodeFormat(DecodeFormat.PREFER\_ARGB\_8888);

默认格式RGB\_565使用内存是ARGB\_8888的一半，但是图片质量就没那么高了，而且不支持透明度

#### **4.）设置缓存内存大小**

//设置BitmapPool缓存内存大小

builder.setBitmapPool(new LruBitmapPool(memoryCacheSize));

#### **5.)设置一个用来检索cache中没有的Resource的ExecutorService**

为了使缩略图请求正确工作，实现类必须把请求根据Priority优先级排好序。

builder.setDiskCacheService(ExecutorService service);

builder.setResizeService(ExecutorService service);

### **使用ModelLoader自定义数据源：**

例如我们使用了七牛云存储，要根据不同的要求请求不同尺寸不同质量的图片，这时我们就可以使用自定义数据源

#### **1.)定义处理URL接口**

public interface IDataModel {

String buildDataModelUrl(int width, int height);

}

#### **2.）实现处理URL接口**

JpgDataModel

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

public class JpgDataModel implements IDataModel {

private String dataModelUrl;

public JpgDataModel(String dataModelUrl) {

this.dataModelUrl = dataModelUrl;

}

@Override

public String buildDataModelUrl(int width, int height) {

//http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg?imageView2/1/w/200/h/200/format/jpg

return String.format("%s?imageView2/1/w/%d/h/%d/format/jpg", dataModelUrl, width, height);

}

}

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

WebpDataModel

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

public class WebpDataModel implements IDataModel {

private String dataModelUrl;

public WebpDataModel(String dataModelUrl) {

this.dataModelUrl = dataModelUrl;

}

@Override

public String buildDataModelUrl(int width, int height) {

//http://78re52.com1.z0.glb.clouddn.com/resource/gogopher.jpg?imageView2/1/w/200/h/200/format/webp

return String.format("%s?imageView2/1/w/%d/h/%d/format/webp", dataModelUrl, width, height);

}

}

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

#### **3.）实现ModelLoader**

[IMG_264](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

public class MyDataLoader extends BaseGlideUrlLoader<IDataModel> {

public MyDataLoader(Context context) {

super(context);

}

public MyDataLoader(ModelLoader<GlideUrl, InputStream> urlLoader) {

super(urlLoader, null);

}

@Override

protected String getUrl(IDataModel model, int width, int height) {

return model.buildDataModelUrl(width, height);

}

/\*\*

\*/

public static class Factory implements ModelLoaderFactory<IDataModel, InputStream> {

@Override

public ModelLoader<IDataModel, InputStream> build(Context context, GenericLoaderFactory factories) {

return new MyDataLoader(factories.buildModelLoader(GlideUrl.class, InputStream.class));

}

@Override

public void teardown() {

}

}

}

[IMG_265](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

#### **4.）根据不同的要求采用不同的策略加载图片**

//加载jpg图片

Glide.with(this).using(new MyDataLoader(this)).load(new JpgDataModel(imageUrl)).into(imageView);

//加载webp图片

Glide.with(this).using(new MyDataLoader(this)).load(new WebpDataModel(imageUrl)).into(imageView);

#### **5.)如何跳过.using()**

[IMG_266](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

public class MyGlideModule implements GlideModule {

...

@Override

public void registerComponents(Context context, Glide glide) {

glide.register(IDataModel.class, InputStream.class,

new MyUrlLoader.Factory());

}

}

[IMG_267](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

上面的实现跳过using()

//加载jpg图片

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

//加载webp图片

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

### **使用signature()实现自定义cacheKey：**

    Glide 以 url、viewwidth、viewheight、屏幕的分辨率等做为联合key，官方api中没有提供删除图片缓存的函数，官方提供了signature()方法，将一个附加的数据加入到缓存key当中，多媒体存储数据，可用MediaStoreSignature类作为标识符，会将文件的修改时间、mimeType等信息作为cacheKey的一部分，通过改变key来实现图片失效相当于软删除。

#### **1.）使用StringSignature**

Glide.with(this).load(yourFileDataModel).signature(new StringSignature("1.0.0")).into(imageView);

#### **2.)使用MediaStoreSignature**

Glide.with(this) .load(mediaStoreUri).signature(new MediaStoreSignature(mimeType, dateModified, orientation)).into(view);

#### **3.)使用自定义Signature**

[IMG_268](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)

public class IntegerVersionSignature implements Key {

private int currentVersion;

public IntegerVersionSignature(int currentVersion) {

this.currentVersion = currentVersion;

}

@Override

public boolean equals(Object o) {

if (o instanceof IntegerVersionSignature) {

IntegerVersionSignature other = (IntegerVersionSignature) o;

return currentVersion = other.currentVersion;

}

return false;

}

@Override

public int hashCode() {

return currentVersion;

}

@Override

public void updateDiskCacheKey(MessageDigest md) {

messageDigest.update(ByteBuffer.allocate(Integer.SIZE)

.putInt(signature).array());

}

}

[IMG_269](http://www.cnblogs.com/whoislcj/p/javascript:void(0);)