# Glide 4.x分析

# 流程图

Glide.pdf

亿图图示源文件 Glide.eddx

亿图图示有元素大小显示，使用破解版每次都会缩小卡死。

# 参考文档

Android开发知识（二十七）Glide4.x的源码解析（上）

https://blog.csdn.net/lc\_miao/article/details/106442804

Glide 源码分析（4.13.2）

https://blog.csdn.net/u013347784/article/details/125728996

# 概念

## 注册中心

private final ModelLoaderRegistry modelLoaderRegistry;

用于将任意复杂的数据模型转换为具体的数据类型

包含的内容太多了不想写

private final EncoderRegistry encoderRegistry;

包含能够对任意数据类型进行编码的编码器的有序列表

存储下面的值：

ByteBuffer.class, new ByteBufferEncoder()

InputStream.class, new StreamEncoder(arrayPool)

包含一个有序的资源解码器列表，能够将任意数据类型解码为从最高优先级解码器到最低优先级解码器的任意资源类型。

private final ResourceDecoderRegistry decoderRegistry;

包含能够对任意资源类型进行编码的资源编码器的有序列表

private final ResourceEncoderRegistry resourceEncoderRegistry;

储数据类到DataRewinder的映射。工厂和允许注册新类型和工厂。

DataRewinde：数据包装接口

private final DataRewinderRegistry dataRewinderRegistry;

new ByteBufferRewinder.Factory()

new InputStreamRewinder.Factory(arrayPool)

new ParcelFileDescriptorRewinder.Factory()

允许ResourceTranscoder通过它们之间转换的类进行注册和检索

private final TranscoderRegistry transcoderRegistry;

Bitmap.class, BitmapDrawable.class, new BitmapDrawableTranscoder(resources)

Bitmap.class, byte[].class, bitmapBytesTranscoder

Drawable.class,byte[].class,new DrawableBytesTranscoder(bitmapPool, bitmapBytesTranscoder, gifDrawableBytesTranscoder)

GifDrawable.class, byte[].class, gifDrawableBytesTranscoder

包含能够解析图像头的ImageHeaderParser的无序列表

private final ImageHeaderParserRegistry imageHeaderParserRegistry;

DefaultImageHeaderParser()

ExifInterfaceImageHeaderParser()

## DecodeJob.Stage

INITIALIZE：初始化阶段。

RESOURCE\_CACHE：解码的数据来源是缓存中处理过的文件。这里就不是内存缓存了。因为内存缓存获取到了，就不会走到这里来，在上一章就已经结束了加载流程。那这个缓存肯定就是磁盘缓存了。

DATA\_CACHE：解码数据来源是未处理过的源数据，这里先理解为从网络上download下来未处理的源数据缓存。

SOURCE：解码数据来源是源文件。

ENCODE：在成功下载资源并且缩放后，开始编码到本地。

FINISHED：解码这一阶段的流程结束。

# 流程

式样

微软雅黑 Light 5号

段落 固定值 23磅

# 目录1

## 目录2

### 目录3

#### 目录4

##### 目录5