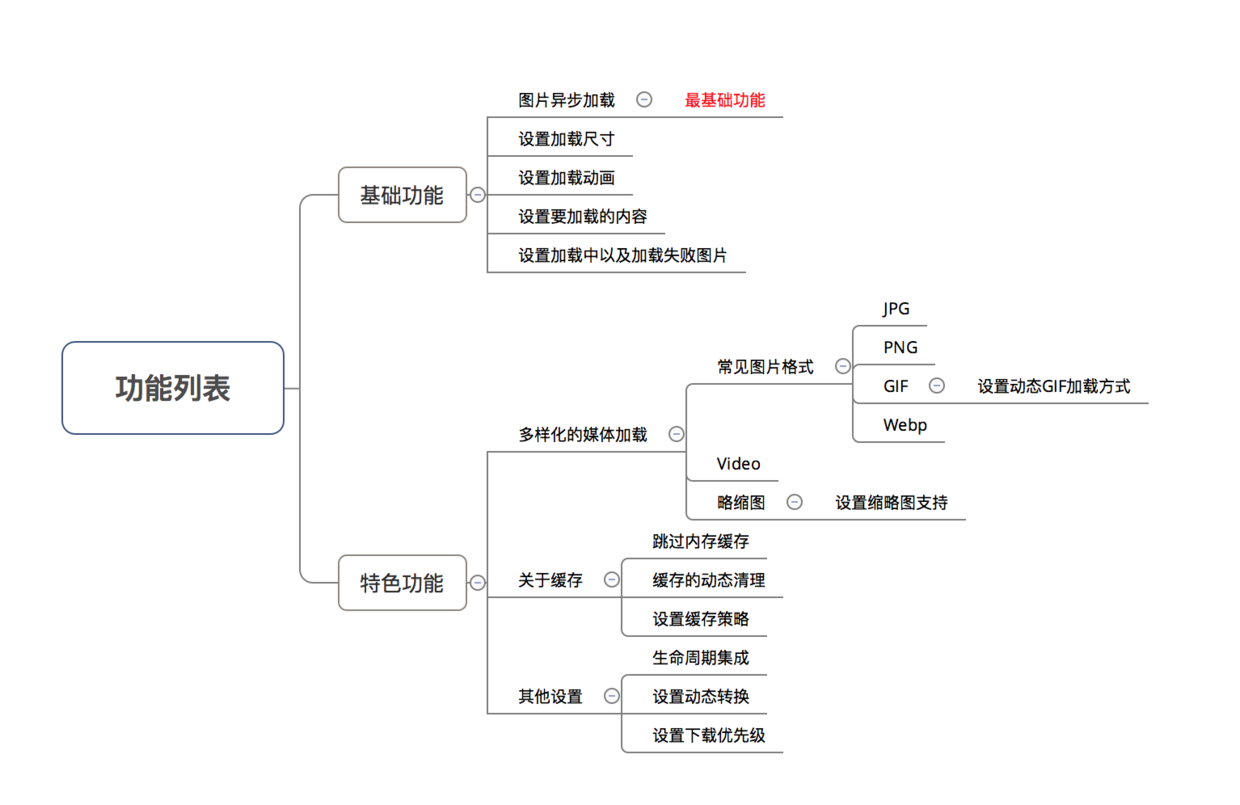
# Glide篇

1. 简介：Glide是Google开发者Sam sjudd出品的一个Android开源库
2. 作用：图片加载
3. 具体功能列表



1. 常用方式

Glide.with().load().into();

1. 源码分析

##### with()分析

为了得到一个RequestManager对象，从而将Glide加载图片周期与Activity和Fragment进行绑定，从而管理Glide加载图片周期

方法：创建Fragment，向当前的Activity中添加一个隐藏的Fragment，将RequestManager与这个隐藏的Fragment进行绑定

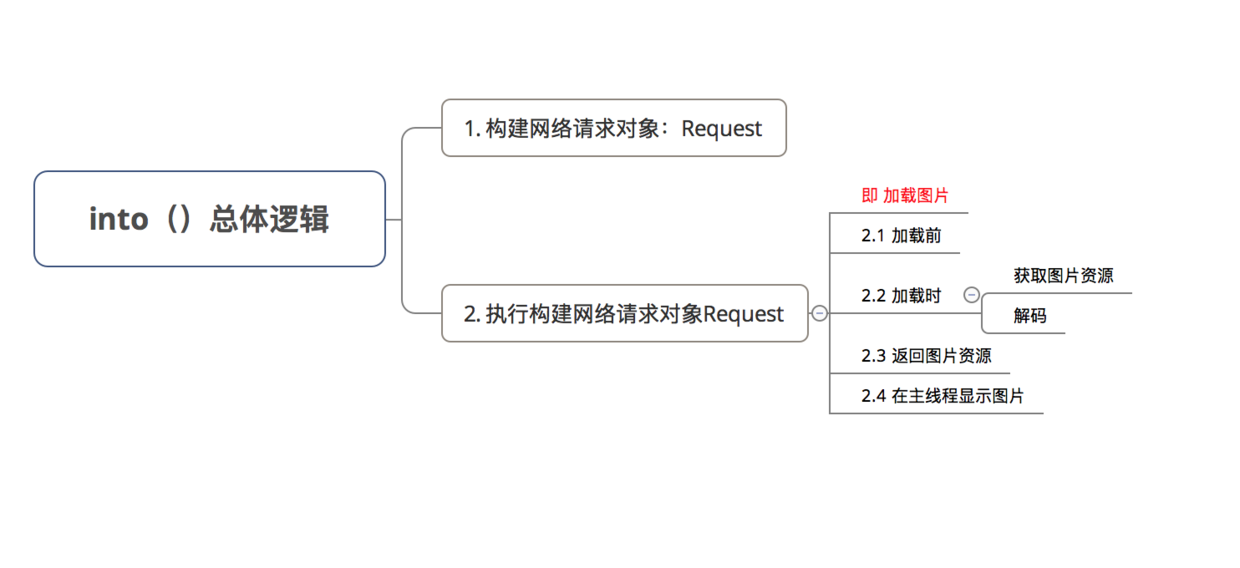
##### Load()分析

load（）中预先创建好对图片进行一系列操作（加载、编解码、转码）的对象，并全部封装到 DrawableTypeRequest对象中。

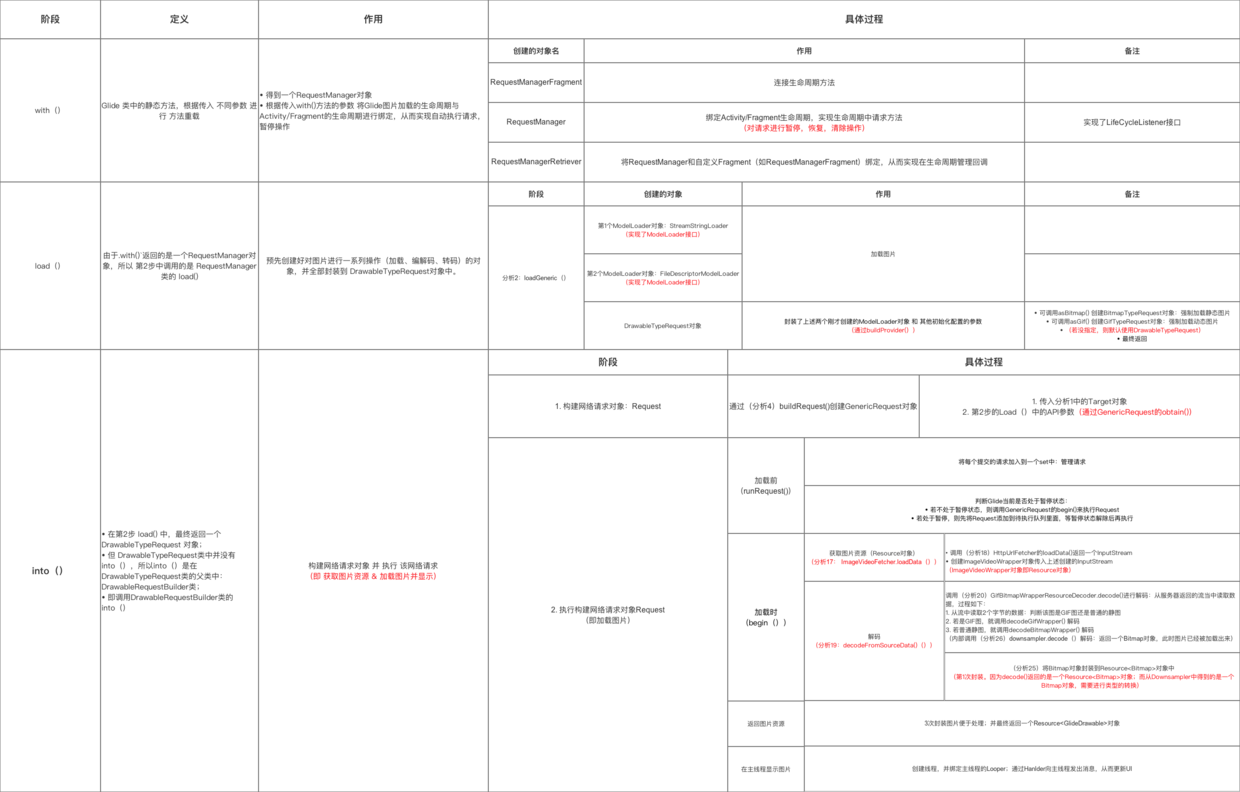
##### Into()分析

作用：构建网络请求对象并执行网络请求(即获取图片资源并加载图片显示)

总体逻辑如下：



看看Glide加载的全过程



##### 优点

* 多样化媒体加载   
  Glide 不仅是一个图片缓存，它支持 Gif、WebP、缩略图。甚至是 Video
* 生命周期集成   
  通过设置绑定生命周期，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
* 高效的缓存策略
* 支持Memory和Disk图片缓存
* Picasso 只会缓存原始尺寸的图片，而 Glide 缓存的是多种规格，也就意味着 Glide 会根据你 ImageView 的大小来缓存相应大小的图片尺寸，比如你 ImageView 大小是200\*200，原图是 400\*400 ，而使用 Glide 就会缓存 200\*200 规格的图，而 Picasso 只会缓存 400\*400 规格的。这个改进就会导致 Glide 比 Picasso 加载的速度要快，毕竟少了每次裁剪重新渲染的过程，非常灵活 & 加载速度快
* 内存开销小   
  默认的 Bitmap 格式是 RGB\_565 格式，而 Picasso 默认的是 ARGB\_8888 格式，这个内存开销要小一半。

Android关于图片内存计算，共有四种，分别是：

* ALPHA\_8：每个像素占用1byte内存
* ARGB\_4444:每个像素占用2byte内存
* ARGB\_8888:每个像素占用4byte内存（默认，色彩最细腻=显示质量最高=占用的内存也最大）
* RGB\_565:每个像素占用2byte内存（8bit = 1byte）

举例说明：一个32位的PNG=ARGB\_8888=1204x1024,那么占用空间是:1024x1024x(32/8) = 4,194,304kb=4M左右

在解析图片的时候，为了避免oom和节省内存，最好使用ARGB\_4444模式（节省一半的内存空间）

##### 

##### 缺点

* 使用方法复杂   
  由于Glide其功能强大，所以使用的方法非常多，其源码也相对的复杂
* 包较大

##### 

##### 相比其他图片加载库（Picasso & Fresco）

* 对比Picasso   
  Glide 是在Picasso 基础之上进行的二次开发做了不少改进，不过这也导致包比 Picasso 大不少，不过也就不到 500k（Picasso 是100多k），用法较为复杂，不过毕竟级别还是蛮小的，影响不是很大
* 对比Fresco   
  使用较Fresco简单，但性能（加载速度 & 缓存）却比不上Fresco

##### 应用场景

根据Glide的特点和与其他图片加载库的对比，可以得出其使用场景：

* 需要更多的内容表现形式(如Gif)；
* 更高的性能要求（缓存 & 加载速度）；