Glide

Glide 初始化 流程和基本认识

第一次使用Glide.with()时初始化：

Glide.with(Context context)🡪 getRetriever(Context context)🡪Glide.get(Context context);

public static Glide get(@NonNull Context context) {  
 if (*glide* == null) {  
 GeneratedAppGlideModule annotationGeneratedModule =  
 *getAnnotationGeneratedGlideModules*(context.getApplicationContext());  
 synchronized (Glide.class) {  
 if (*glide* == null) {  
 *checkAndInitializeGlide*(context, annotationGeneratedModule);  
 }  
 }  
 }  
  
 return *glide*;  
}

get(Content content) 中使用单例设计模式，即Glide 对象在APP生命周期中是全局单例的；

Glide 对象初始化具体实现函数：

checkAndInitializeGlide()🡪initializeGlide()🡪

读取Manifest中自定义的GlideModule

initializeGlide(context, new GlideBuilder(), generatedAppGlideModule);

Glide实例化：new GlideBuilder()对象建立Glide对象实例化辅助类（build模式）：

GlideBuilder; build(@NonNull Context context)

1.线程池（包括任务执行线程池、缓存线程池、动画线程池）

2.初始缓存模块：内存缓存：大小计算、bitmap、arrary、LruResource 初始化；磁盘缓存；

3.创建RequestManager管理器RequestManagerRetriever

4.创建Engine对象

5.回调Glide构造函数

Glide构造函数中初始化Glide各项能力注册管理机制：

创建GlideContext

Into（View）

1. build 请求参数BaseRequestOptions 包括ImageView 缩放类型等；
2. build Targe

2.设置RequestListener、回调线程执行器Executor；

3. buildRequest(）递归build Request实例对象（以SingleRequest为例）, 建立track ：requestManager.track(target, request);

4. Request. begin() （ RequestTracker. runRequest（）触发）；

计算需要图片尺寸

5. Request onSizeReady（）

6. Engine. Load();

buildEngineKey 🡪 loadFromMemory()

Glide 内存机制

图片资源缓存ActiveResource（一级内存缓存）