1. LookupReques窥探请求，将从本地缓存文件中窥探的二进制数据缓存到它的成员变量中

2.索引文件(.idx结尾),采用内存映射文件形式，将每个文件的基本信息存在32位头数据之后，基本信息采用 8(long)+4(int)形式记录，long数据缓存的是这个文件的key，int数据缓存的是不包括它本身的之前的最大缓存字节数大小（mActiveByte >= DATA\_HEADER\_SIZE，其实也是记录图片字节数组在数据块中的起始位置），之后将mActiveByte加上一个数据块头数据大小和他本身，并记录在内存字节数组中mIndexHeader中，注意以上步骤是针对内存映射文件的索引文件,以下是真正保存解码的数据(.0,.1)

3.数据块文件(.0,.1)，当前活动的文件(mActiveDataFile)指向的是mDataFile0(RandomAccessFile)文件，它首先写入一个头数据(mBlobHeader)，然后再写入真正解码的数据，其中头数据中有个当前最大活动的字节数记录的也是不包括它本身的之前的最大缓存字节数大小，然后也是有key，待缓存的数据长度

4.从窥探请求获得的数据放到ByteBuffer中供界面使用，然后不会无限开辟ByteBuffer,而是通过ByteBufferPool来循环利用ByteBuffer,这个ByteBufferPool的阀值是4，每个ByteBuffer的阀值是200K

5. mMaxBytes针对是缓存每个数据块的阀值，DATA\_HEADER\_SIZE是整个数据块文件的一个头数据，BLOB\_HEADER\_SIZE是每个数据块文件的一个头大小

6.AppHub(QAlbumModel)中有BitmapPool, BytesBufferPool，ThreadPool

7.Activity中getExternalCacheDir()用来获取系统缓存路径

8. LocalImageRequest extends ImageCacheRequest ,ImageCacheRequest是图片缓存请求的job,泛型参数是Bitmap

9: CacheManager用来管理BlobCache以及它对应的本地缓存文件的创建，删除和关闭

10: ImageCacheService用BlobCache根据文件名来进行具体的存取操作，不过数据都是用字节数组存放的二进制数据,不是通过返回值得到，而是通过参数获取

11：ThumbNailLoader extends BitmapLoader在Adapter之中用于请求图片，这边传入position是用于渲染的时候的用，防止图片显示错乱