


```
1 //options 选择方式
2
3 [self.image1 sd_setImageWithURL:imagePath1 placeholderImage:[UIImage imageNamed:@"de
```

其他就不一一介绍了，oc是自文档语言，看方法名就知道干什么的了。除了带options选项的方法，其他的方法都是综合存储，也就是内存缓存和磁盘缓存结合的方式，如果你只需要内存缓存，那么在options这里选择SDWebImageCacheMemoryOnly就可以了。

如果不想深入了解，到这里你已经可以用SDWebimage进行图片缓存了，接下来我要解释options的所有选项，以及SDWebImage内部执行流程。

一、options所有选项：

```
1 //失败后重试
2 SDWebImageRetryFailed = 1 << 0,
3
4 //UI交互期间开始下载，导致延迟下载比如UIScrollView减速。
5 SDWebImageLowPriority = 1 << 1,
6
7 //只进行内存缓存
8 SDWebImageCacheMemoryOnly = 1 << 2,
9
10 //这个标志可以渐进式下载,显示的图像是逐步在下载
11 SDWebImageProgressiveDownload = 1 << 3,
12
13 //刷新缓存
14 SDWebImageRefreshCached = 1 << 4,
15
16 //后台下载
17 SDWebImageContinueInBackground = 1 << 5,
18
19 //NSMutableURLRequest.HTTPShouldHandleCookies = YES;
20
21 SDWebImageHandleCookies = 1 << 6,
22
23 //允许使用无效的SSL证书
24 //SDWebImageAllowInvalidSSLCertificates = 1 << 7,
25
26 //优先下载
27 SDWebImageHighPriority = 1 << 8,
28
29 //延迟占位符
30 SDWebImageDelayPlaceholder = 1 << 9,
31
32 //改变动画形象
33 SDWebImageTransformAnimatedImage = 1 << 10,
```

二、SDWebImage内部实现过程

- 入口 setImageWithURL:placeholderImage:options: 会先把 placeholderImage 显示，然后SDWebImageManager 根据 URL 开始处理图片。
- 进入 SDWebImageManager-downloadWithURL:delegate:options:userInfo:，交给 SDImageCache 从缓存查找图片是否已经下载 queryDiskCacheForKey:delegate:userInfo:。
- 先从内存图片缓存查找是否有图片，如果内存中已经有图片缓存，SDImageCacheDelegate 回调 imageCache:didFindImage:forKey:userInfo: 到 SDWebImageManager。
- SDWebImageManagerDelegate 回调 webImageManager:didFinishWithImage: 到 UIImageView+WebCache 等前端展示图片。
- 如果内存缓存中没有，生成 NSInvocationOperation 添加到队列开始从硬盘查找图片是否已经缓存。
- 根据 URLKey 在硬盘缓存目录下尝试读取图片文件。这一步是在 NSOperation 进行的操作，所以回主线程进行结果回调 notifyDelegate:。
- 如果上一操作从硬盘读取到了图片，将图片添加到内存缓存中（如果空闲内存过小，会先清空内存缓存）。SDImageCacheDelegate 回调 imageCache:didFindImage:forKey:userInfo:。进而回调展示图片。
- 如果从硬盘缓存目录读取不到图片，说明所有缓存都不存在该图片，需要下载图片，回调 imageCache:didNotFindImageForKey:userInfo:。
- 共享或重新生成一个下载器 SDWebImageDownloader 开始下载图片。
- 图片下载由 NSURLConnection 来做，实现相关 delegate 来判断图片下载中、下载完成和下载失败。
- connection:didReceiveData: 中利用 ImageIO 做了按图片下载进度加载效果。
- connectionDidFinishLoading: 数据下载完成后交给 SDWebImageDecoder 做图片解码处理。

岁啦！至今没收到你的祝福！ ...

从我个人经验来讲，单单从代码角度，我觉得函数主要是拿来提高代码复用
Platycodon_ 评论了 为什么Swift中应该避免使用guard语句...

NSLog(@"happy birthday");
gang544043963 评论了 CocoaChina 八岁啦！至今没收到你的祝福！ ...

看来是时候从苹果走人了，跳到安卓市场吧！
刘世航 评论了 网友吐槽苹果：乔布斯砍掉的臃肿又回来了...

1.HZExtend:应用快速开发架构集合了MVC和MVVM的特点
IOS项目收集者 评论了 为什么Swift中应该避免使用guard语句...

1.HZExtend:应用快速开发架构集合了MVC和MVVM的特点
IOS项目收集者 评论了 iOS开发编码建议与编程经验（持续更新中） ...

相关帖子

关于Storyboard Id 的问题
魔能碰撞 - 手势施法+装备词缀+伙伴养成的多元RPG手游

姜老师的网游学习记录（65）——延时处理表格中的数据

shader效果pc上可以 真机上无效 有人知道怎么解决么

每天App收入在20-30Rmb,是什么水平，，感觉养活不了自己的呀

图片上传

我以为我是发帖最多的.....没想到.....没想到...

iOS屏幕适配问题

Xcode中混合C++代码
关于Storyboard Id 的问题
魔能碰撞 - 手势施法+装备词缀+伙伴养成的多元RPG手游

姜老师的网游学习记录（65）——延时处理表格中的数据

shader效果pc上可以 真机上无效 有人知道怎么解决么

每天App收入在20-30Rmb,是什么水平，，感觉养活不了自己的呀

图片上传

我以为我是发帖最多的.....没想到.....没想到...

iOS屏幕适配问题

Xcode中混合C++代码

13. 图片解码处理在一个 NSOperationQueue 完成，不会拖慢主线程 UI。如果有需要对下载的图片进行二次处理，最好也在这里完成，效率会好很多。
14. 在主线程 notifyDelegateOnMainThreadWithInfo: 宣告解码完成，imageDecoder:didFinishDecodingImage:userInfo: 回调给 SDWebImageDownloader。
15. imageDownloader:didFinishWithImage: 回调给 SDWebImageManager 告知图片下载完成。
16. 通知所有的 downloadDelegates 下载完成，回调给需要的地方展示图片。
17. 将图片保存到 SDImageCache 中，内存缓存和硬盘缓存同时保存。写文件到硬盘也在以单独 NSInvocationOperation 完成，避免拖慢主线程。
18. SDImageCache 在初始化的时候会注册一些消息通知，在内存警告或退到后台的时候清理内存图片缓存，应用结束的时候清理过期图片。
19. SDWI 也提供了 UIButton+WebCache 和 MKAnnotationView+WebCache，方便使用。
20. SDWebImagePrefetcher 可以预先下载图片，方便后续使用。

从上面流程可以看出，当你调用setImageWithURL:方法的时候，他会自动去给你干这么多事，当你需要在某一具体时刻做事情的时候，你可以覆盖这些方法。比如在下载某个图片的过程中要响应一个事件，就覆盖这个方法：

```
1 //覆盖方法，指哪打哪，这个方法是下载imagePath2的时候响应
2 SDWebImageManager *manager = [SDWebImageManager sharedManager];
3
4 [manager downloadImageWithURL:imagePath2 options:SDWebImageRetryFailed progress:^(NSTimeInterval elapsed) {
5     NSLog(@"显示当前进度");
6 }
7 }
8 } completed:^(UIImage *image, NSError *error, SDImageCacheType cacheType, BOOL finished) {
9     NSLog(@"下载完成");
10 }
11 }];
```

对于初级来说，用sd_setImageWithURL:的若干个方法就可以实现很好的图片缓存。

demo[下载点\[这里\]\(#\)](#)



微信扫一扫

订阅每日移动开发及APP推广热点资讯

公众号：CocoaChina

我要投稿

收藏文章

分享到：





201201

上一篇：Xcode概览：测试应用程序

下一篇：Xcode概览：保存或者恢复更改

相关资讯


- SDWebImage源码解析之SDWebImageManager的注解


■ iOS图片缓存库基准对比

■ 自定义ImageView for Cocoa 比SDWebImage更简单易用
- Chun 阅读笔记 - 如何做一个图片缓存库

■ Path图片缓存实现方法：Fast Image Cache

■ 【分享】对SDWebImage的扩展


 微博



CocoaChina

+ 加关注

「iOS开发编码建议与编程经验」在开发过程中，我们不仅要去别人的代码，也要让别人看我们的代码。那么，有一个良好的编码习惯将会非常重要。下面将会罗列使用Objective-C来开发iOS的编码建议。<http://t.cn/RqvWx2>



今天 14:16

转发(15)

|

评论(9)

「iOS runtime实战应用：成员变量和属性」有筒子在面试的时候，遇到这样一个问题：“你知道成员变量的本质是什么吗？”，筒子立马懵逼了，成员变量的本质？成员变量就是成员变量啊，平时口答用，还有什么更深入的



精准推送
就用极光推送

集成简单 高并发 毫秒送达

极光推送



发表评论



2016-02-27 10:49:38

 0 0 回复



2015-12-21 01:40:04

👍 0 💬 0 回复



2015-12-21 01:37:40

👍 0 💬 0 回复



2015-10-27 08:39:41

👍 0 💬 0 回复



2016-02-27 10:49:38

0 0 回复



2015-12-21 01:40:04

👍 0 💬 0 回复



2015-12-21 01:37:40

👍 0 💬 0 回复



2015-10-27 08:39:41

👍 0 💬 0 回复

