

[iOS]仿微博视频边下边播之封装播放器



作者 NewPan (/u/e2f2d779c022) [+ 关注](#)

2016.09.16 01:10 字数 3544 阅读 5053 评论 60 喜欢 121 阅读 5053 评论 60 喜欢 121 (/u/e2f2d779c022)

感谢简书作者@夜千寻墨 (<http://www.jianshu.com/users/527ecf8c8753>), 我框架中的 JPVideoURLAssetResourceLoader类的源码是在他的 iOS视频边下边播--缓存播放数据流 (<http://www.jianshu.com/p/990ee3db0563>) 基础上修改的。

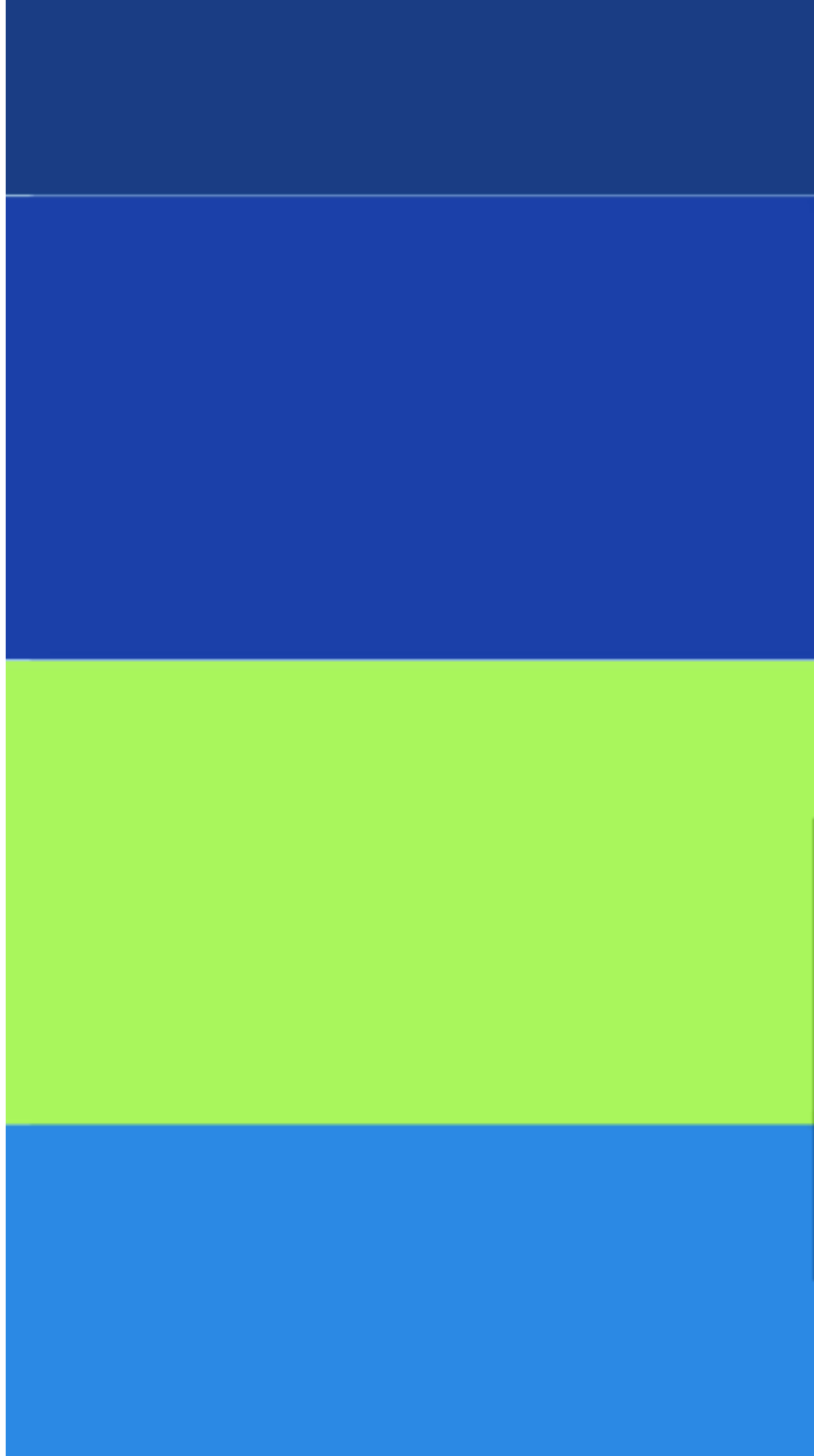


Tips: 这次的内容分为两篇文章讲述
01、[iOS]仿微博视频边下边播之封装播放器
(<http://www.jianshu.com/p/0d4588a7540f>) 讲述如何封装一个实现了边下边播并且缓存的视频播放器。
02、[iOS]仿微博视频边下边播之滑动TableView自动播放
(<http://www.jianshu.com/p/3946317760a6>) 讲述如何实现在tableView中滑动播放视频，并且是流畅，不阻塞线程，没有任何卡顿的实现滑动播放视频。同时也将讲述当tableView滚动时，以什么样的策略，来确定究竟哪一个cell应该播放视频。

微博视频的特点：

- 秒拍团队主要致力于视频处理，微博的视频播放功能是由秒拍提供技术支持的。微博的视频一般都是不限时长的，所以它的特点是边下边播。
- 说到视频播放就不能不提微信的短视频，微信的短视频限制时长为15秒，经过微信团队处理后，一个短视频的体积能控制在2MB以内。所以微信的视频是先下载，再读取下载好的视频文件进行播放，也就是所谓的先下后播。这个功能，微信的同行已经把源码分享出来了，在这里 (<http://www.jianshu.com/p/3d5ccbde0de1>)。

我找了很多资料，没有找到完全意义上，实现了微博首页列表视频边下边播功能的资料。但是我自己项目中又有这个需求，所以只能自己动手。最后实现的效果如下：



这个列表视频边下边播包含以下主要的功能点：

- 01.必须是边下边播。
- 02.如果缓存好的视频是完整的，就要把这个视频保存起来，下次再次加载这个视频的时候，就先检查本地有没有缓存好的视频。这一点对于节省用户流量，提升用户体验很重要。要实现这一点，也就是说，我们要手动干预系统播放器加载数据的内部实现，这个细节后面再讲。
- 03.不阻塞线程，不卡顿，滑动如丝顺滑，这是保证用户体验最重要的一点。
- 04.当tableView滚动时，以什么样的策略，来确定究竟哪一个cell应该播放视频。

可能你着急赶项目，只想尽快的把这个功能集成到你的项目，那么请你直接去 Github (<https://github.com/Chris-Pan/JPVideoPlayer>) 上下载源码。需要说明的是，我上面说的功能点的第一和第二点，不用你关心，我已经帮你处理封装好了。但是，第三和第四点，需要你自己结合你自己的项目来定制，我只提供了模板和巨细无比的注释。

接下来就来看看我是怎么实现这些功能的。

第一、AVPlayer基本使用？

首先从最基本的封装播放器开始。

01、AVPlayer？

AVPlayer播放视频需要涉及以下几个类：

- AVURLAsset，是AVAsset的子类，负责网络连接，请求数据。
- AVPlayerItem，会建立媒体资源动态视角的数据模型并保存AVPlayer播放资源的状

态。说白了，就是数据管家。

- AVPlayer，播放器，将数据解码处理成为图像和声音。
- AVPlayerLayer，图像层，AVPlayer的图像要通过AVPlayerLayer呈现。

需要注意的是，AVPlayer的模式是，你不要主动调用play方法播放视频，而是等待AVPlayerItem告诉你，我已经准备好播放了，你现在可以播放了，所以我们要监听AVPlayerItem的状态，通过添加监听者的方式获取AVPlayerItem的状态：

```
// 添加监听
[_currentPlayerItem addObserver:self forKeyPath:@"status" options:NSKeyValueObservingOptionNew context:nil];
```

在监听结果中处理播放逻辑。当监听到播放器已经准备好播放的时候，就可以调用play方法。

注意点：如果视频还没准备好播放，你就把AVPlayerLayer图层添加到cell上，那么在播放器还没有准备好播放之前，负责显示的图像的图层会变成黑色，直到准备好播放，拿到数据，才会出现画面。这在列表中自动播放是应该极力避免的。所以，要等待播放器有图像输出的时候再添加显示的预览图层到cell上。

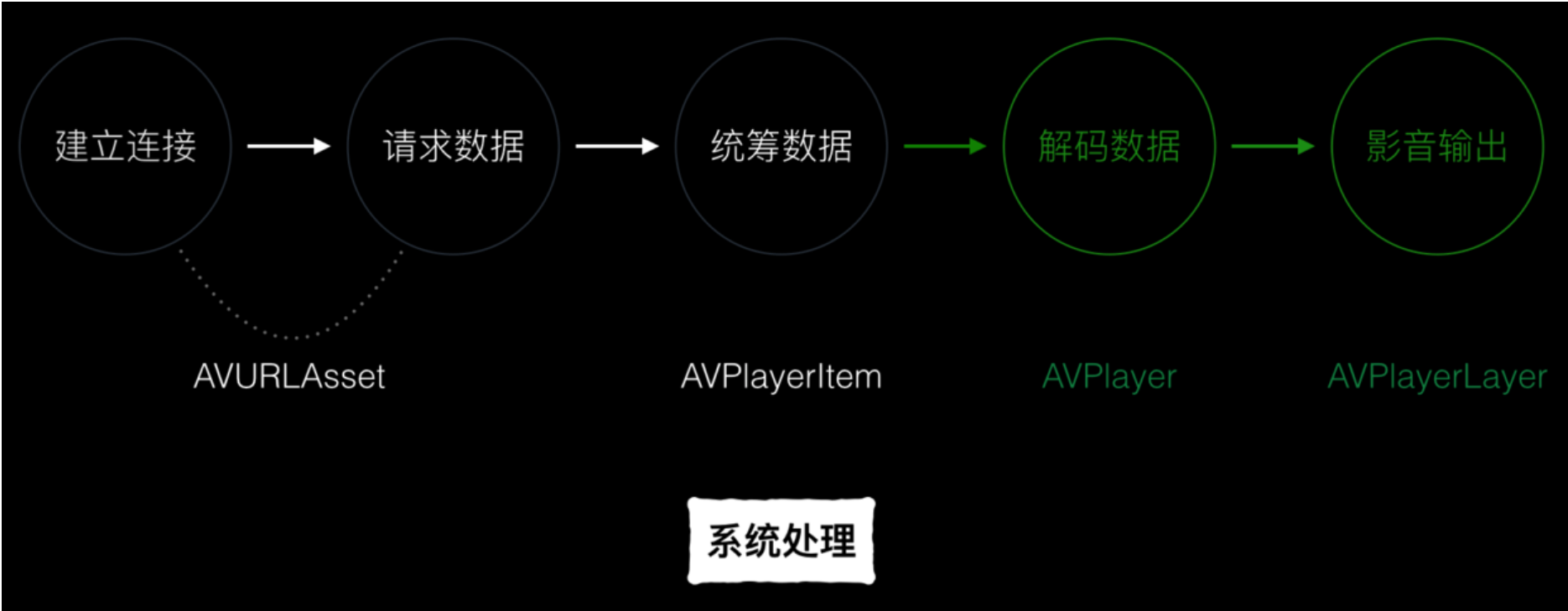
```
-(void)observeValueForKeyPath:(NSString *)keyPath ofObject:(id)object change:(NSDictionary<NSString *,id> *)change context:(void *)context{
    if ([keyPath isEqualToString:@"status"]) {
        AVPlayerItem *playerItem = (AVPlayerItem *)object;
        AVPlayerItemStatus status = playerItem.status;
        switch (status) {
            case AVPlayerItemStatusUnknown:{

            }
            break;
            case AVPlayerItemStatusReadyToPlay:{
                [self.player play];
                self.player.muted = self.mute;
                // 显示图像逻辑
                [self handleShowViewSublayers];

            }
            break;
            case AVPlayerItemStatusFailed:{

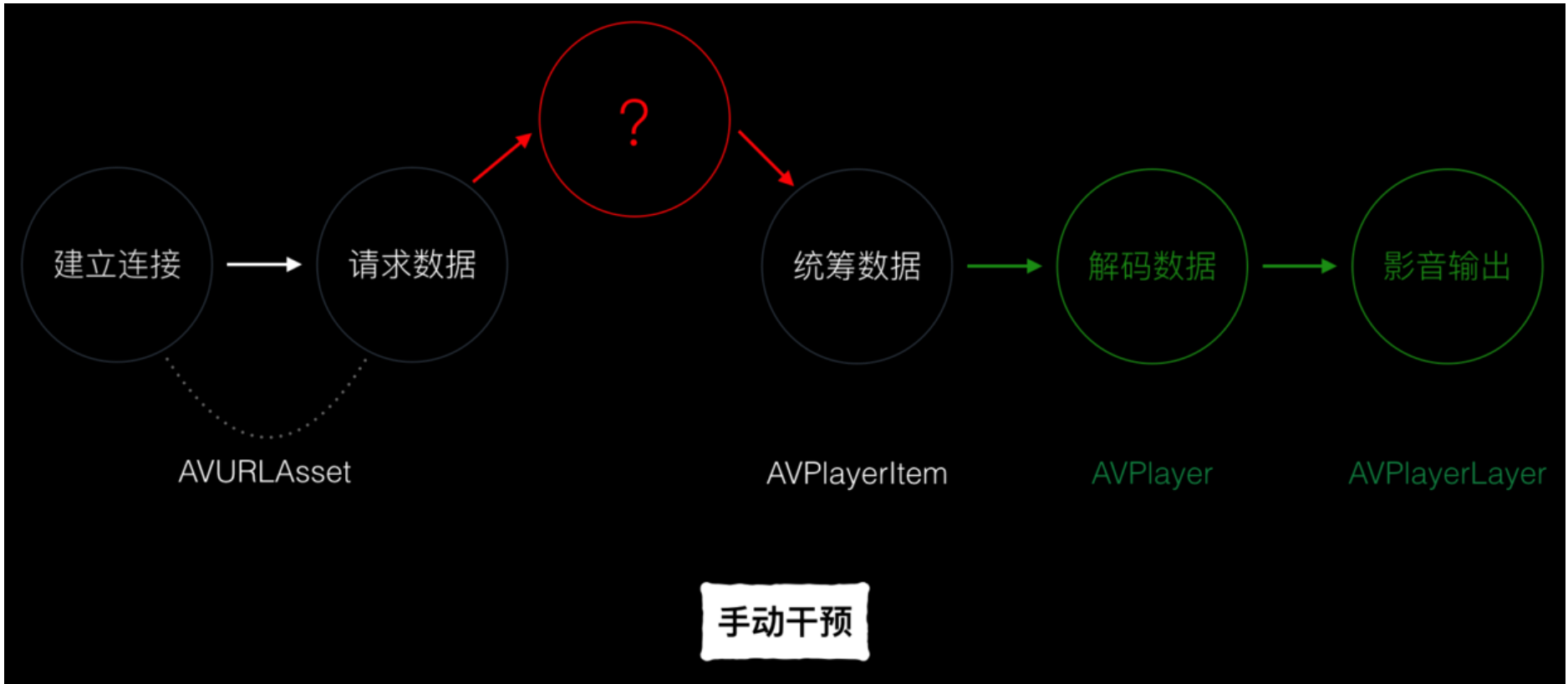
            }
            break;
            default:
                break;
        }
    }
}
```

到这里就可以播放一个网络或者本地视频了。但是，在播放过程中：建立连接-->请求数据-->统筹数据-->数据解码-->输出图像和声音，这些过程都是AVFoundation框架下，我上面列举的那些类自动帮我们完成的。



系统处理.png

要实现边下边播，并实现缓存功能，就必须拿到播放器的数据，也就是必须手动干预数据加载的过程。我们需要在网络层和解码层中间，插入一个我们自己需要的功能块，也就是我下图中的红色模块。



手动干预.png

02、AVAssetResourceLoaderDelegate?

- 要实现在播放器请求中插入自己的模块的功能，我们需要借助于AVAssetResourceLoaderDelegate。我们用到的AVURLAsset下有一个AVAssetResourceLoader属性。

```
@property (nonatomic, readonly) AVAssetResourceLoader *resourceLoader;
```

- 这个AVAssetResourceLoader是负责数据加载的，最最重要的是我们只要遵守了AVAssetResourceLoaderDelegate，就可以成为它的代理，成为它的代理以后，数据加载可能会通过代理方法询问我们。

~~* 这个AVAssetResourceLoader是负责数据加载的，最最重要的是我们只要遵守了AVAssetResourceLoaderDelegate，就可以成为它的代理，成为它的代理以后，数据加载都会通过代理方法询问我们。这样，我们就找到切入点干预数据的加载了。~~

```
-(BOOL)resourceLoader:(AVAssetResourceLoader *)resourceLoader shouldWaitForLoadingOfRequestedResource:(AVAssetResourceLoadingRequest *)loadingRequest;
-(void)resourceLoader:(AVAssetResourceLoader *)resourceLoader didCancelLoadingRequest:(AVAssetResourceLoadingRequest *)loadingRequest;
```

- AVAssetResourceLoader通过你提供的委托对象去调节AVURLAsset所需要的加载资源。而很重要的一点是，AVAssetResourceLoader仅在AVURLAsset不知道如何去加载这个URL资源时才会被调用，就是说你提供的委托对象在AVURLAsset不知道如何加载资源时才会得到调用。所以我们又要通过一些方法来曲线解决这个问题，把我们目标视频URL地址的scheme替换为系统不能识别的scheme。
~~* 在正式进入数据干预之前，我们先看一个很重要的东西。我们知道视频数据都是容量巨大的连续媒体数据，所以请求数据的时候，我们要将请求策略置为streaming。这个策略的含义是，将容量巨大的连续媒体数据进行分段，分割为数量众多的小文件进行传递。~~


```
- (NSURL *)getSchemeVideoURL:(NSURL *)url{
    // NSURLComponents用来替代NSMutableURL，可以readwrite修改URL。这里通过更改请求策略，
    // 将容量巨大的连续媒体数据进行分段
    // AVAssetResourceLoader通过你提供的委托对象去调节AVURLAsset所需要的加载资源。
    // 而很重要的一点是，AVAssetResourceLoader仅在AVURLAsset不知道如何去加载这个URL资源
    // 时才会被调用
    // 就是说你提供的委托对象在AVURLAsset不知道如何加载资源时才会得到调用。
    // 所以我们要通过一些方法来曲线解决这个问题，把我们目标视频URL地址的scheme替换为系统不
    // 能识别的scheme
    NSURLComponents *components = [[NSURLComponents alloc] initWithURL:url resolvingAgainstBaseURL:NO];
    components.scheme = @"systemCannotRecognition";
    return [components URL];
}
```

第二、手动干预系统播放器加载数据？

01、如何使用NSURLSession来下载大文件？

在NSURLSession之前，大家都是使用NSURLConnection。如今在Xcode7中，NSURLConnection已经成为过期的类目了，我们常用的AFNNetwork也彻底抛弃了NSURLConnection，转向NSURLSession。现在看一下怎么使用NSURLSession：

```
// 替代NSMutableURL，可以动态修改scheme
NSURLComponents *actualURLComponents = [[NSURLComponents alloc] initWithURL:url resolvingAgainstBaseURL:NO];
actualURLComponents.scheme = @"http";

// 创建请求
NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:[actualURLComponents URL] cachePolicy:NSURLRequestReloadIgnoringCacheData timeoutInterval:20.0];

// 修改请求数据范围
if (offset > 0 && self.videoLength > 0) {
    [request addValue:[NSString stringWithFormat:@"bytes=%ld-%ld", (unsigned long)offset, (unsigned long)self.videoLength - 1] forHTTPHeaderField:@"Range"];
}

// 重置
[self.session invalidateAndCancel];

// 创建Session，并设置代理
self.session = [NSURLSession sessionWithConfiguration:[NSURLSessionConfiguration defaultSessionConfiguration] delegate:self delegateQueue:[NSOperationQueue mainQueue]];

// 创建会话对象
NSURLSessionDataTask *dataTask = [self.session dataTaskWithRequest:request];

// 开始下载
[dataTask resume];
```

我们可以在NSURLSession的代理方法中获得下载的数据，拿到下载的数据以后，我们使用NSOutputStream，将数据写入到硬盘中存放临时文件的文件夹。在请求结束的时候，我们判断是否成功下载好文件，如果下载成功，就把这个文件转移到我们的存储成功文件的文件夹。如果下载失败，就把临时数据删除。

```
// 1. 接收到服务器响应的时候
-(void)URLSession:(NSURLSession *)session dataTask:(NSURLSessionDataTask *)dataTask didReceiveResponse:(NSURLResponse *)response completionHandler:(void (^)(NSURLSessionResponseDisposition))completionHandler;

// 2. 接收到服务器返回数据的时候调用，会调用多次
-(void)URLSession:(NSURLSession *)session dataTask:(NSURLSessionDataTask *)dataTask didReceiveData:(NSData *)data;

// 3. 请求结束的时候调用（成功|失败），如果失败那么error有值
-(void)URLSession:(NSURLSession *)session task:(NSURLSessionTask *)task didCompleteWithError:(NSError *)error;
```

02、AVAssetResourceLoader的代理？

为了更好的封装性和可维护性，新建一个文件，让这个文件负责和系统播放器对接数据。上面说到，只要这个文件遵守了AVAssetResourceLoaderDelegate协议，他就有资格代理系统播放器请求数据。并且系统会通过

```
-(BOOL)resourceLoader:(AVAssetResourceLoader *)resourceLoader shouldWaitForLoadingOfRequestedResource:(AVAssetResourceLoadingRequest *)loadingRequest;
```

这个代理方法，把下载请求loadingRequest传给我们。拿到请求以后，首先把请求用一个数组保存起来。为什么要用数组保存起来？因为，当我们拿到请求去下载数据，到数据下载好，这个过程需要的时间是不确定的。

拿到请求以后，我们就需要调用上面封装的NSURLSession下载器来下载文件。

```
- (void)dealLoadingRequest:(AVAssetResourceLoadingRequest *)loadingRequest{
    NSURL *interceptedURL = [loadingRequest.request URL];
    NSRange range = NSMakeRange(loadingRequest.dataRequest.currentOffset, MAXFLOAT);

    if (self.manager) {
        if (self.manager.downLoadingOffset > 0)
            [self processPendingRequests];

        // 如果新的rang的起始位置比当前缓存的位置还大300k，则重新按照range请求数据
        if (self.manager.offset + self.manager.downLoadingOffset + 1024*300 < range.location
            // 如果往回拖也重新请求
            || self.manager.offset > range.location) {
            [self.manager setUrl:interceptedURL offset:range.location];
        }
    }
    else{
        self.manager = [JPDownloadManager new];
        self.manager.delegate = self;
        [self.manager setUrl:interceptedURL offset:0];
    }
}
```

如果文件有下载好，就去检查下载好的数据长度有没有满足请求数据需要的长度，如果满足，就从硬盘的临时文件中取出对应的数据，并把这段数据填充给请求，然后把这个请求从请求列表数组中移除。播放器拿到了这段数据，就可以开始解码播放了。

```
// 判断此次请求的数据是否处理完全，和填充数据
- (BOOL)respondWithDataForRequest:(AVAssetResourceLoadingDataRequest *)dataRequest{
    // 请求起始点
    long long startOffset = dataRequest.requestedOffset;

    // 当前请求点
    if (dataRequest.currentOffset != 0)
        startOffset = dataRequest.currentOffset;

    // 播放器拖拽后大于已经缓存的数据
    if (startOffset > (self.manager.offset + self.manager.downLoadingOffset))
        return NO;

    // 播放器拖拽后小于已经缓存的数据
    if (startOffset < self.manager.offset)
        return NO;

    NSData *fileData = [NSData dataWithContentsOfFile:_videoPath options:NSDataReadingMappedIfSafe error:nil];

    NSInteger unreadBytes = self.manager.downLoadingOffset - self.manager.offset - (NSInteger)startOffset;
    NSUInteger numberOfBytesToRespondWith = MIN((NSUInteger)dataRequest.requestedLength, unreadBytes);

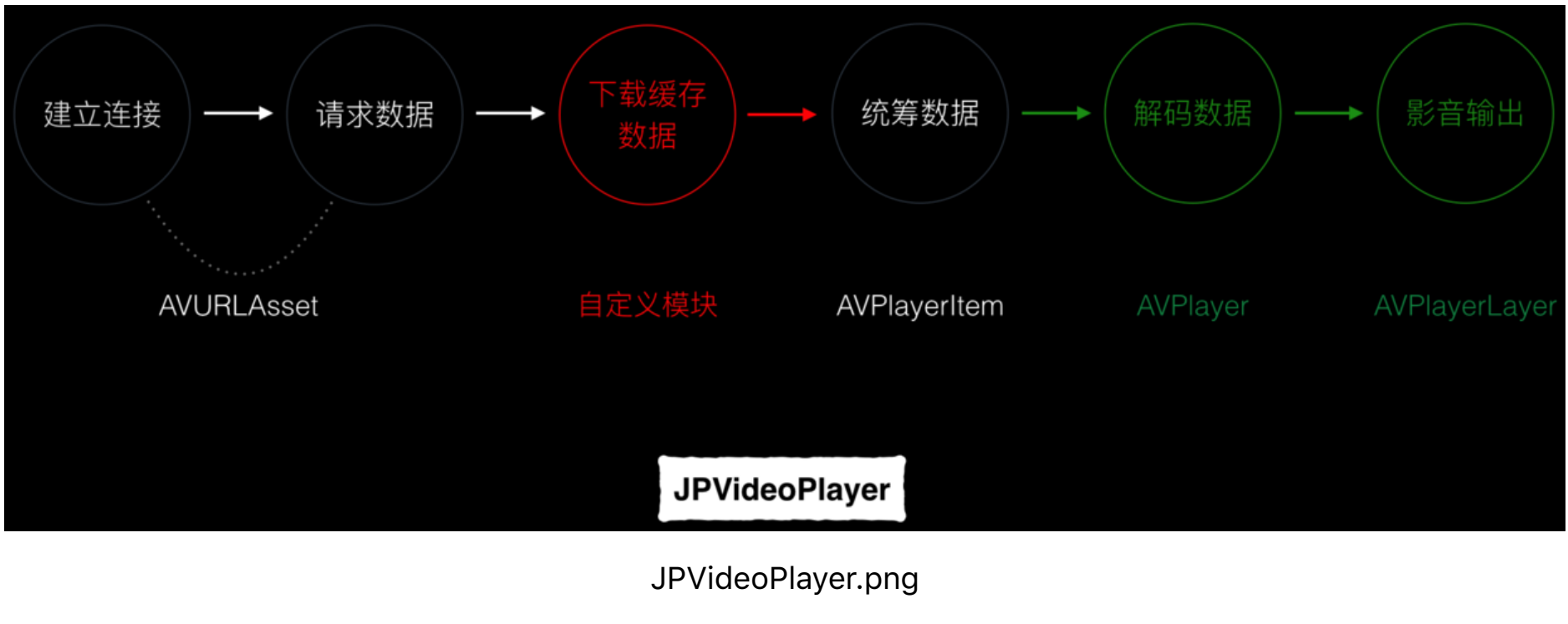
    [dataRequest respondWithData:[fileData subdataWithRange:NSMakeRange((NSInteger)startOffset- self.manager.offset, (NSUInteger)numberOfBytesToRespondWith)]];

    long long endOffset = startOffset + dataRequest.requestedOffset;

    BOOL didRespondFully = (self.manager.offset + self.manager.downLoadingOffset) >= endOffset;

    return didRespondFully;
}
```

至此，手动干预播放视频的流程就走完了。已经可以正常播放视频了。



03、加载缓存数据逻辑？

接下来要做的就是实现，当下次播放同一个视频的时候，先去检查硬盘里有没有这个文件的缓存。借助于NSFileManager，我们可以查找指定的路径有没有存在指定的文件，从而判断有没有缓存可以启用。

```
NSFileManager *manager = [NSFileManager defaultManager];
NSString *savePath = [self fileSavePath];
savePath = [savePath stringByAppendingPathComponent:self.suggestFileName];
if ([manager fileExistsAtPath:savePath]) {
    // 已经存在这个下载好的文件了
    return;
}
```

至此，播放器封装完毕。

我将在下一篇文章 [iOS]仿微博视频边下边播之滑动TableView自动播放 (<http://www.jianshu.com/p/3946317760a6>)，讲述如何实现在tableView中滑动播放视频，并且是流畅，不阻塞线程，没有任何卡顿的实现滑动播放视频。同时也将讲述当tableView滚动时，以什么样的策略，来确定究竟哪一个cell应该播放视频。

03、更新

- 2016.10.09 :
处理在切换视频的短暂时间内, 当前播放视频的cell吸收了滑动事件, 如果滑动当前播放视频的cell, 会导致tableView无法接收到滑动事件, 造成tableView假死。感谢提供bug的朋友@大墙66370 (<http://www.jianshu.com/users/6266c6477c99>) 具体见我的Github JPVideoPlayer (<https://github.com/Chris-Pan/JPVideoPlayer>)。
- 2016.11.04:
简书朋友@菜先生 (<http://www.jianshu.com/users/475fdcde8924>)提交了一个关于单例里重复添加监听的问题, 具体是播放工具单例在每次调用init方法时总会重复添加监听播放完成等的通知, 会导致通知方法重复调用, 这个问题可能带来卡顿. 最新的版本已经修复了这个问题, 具体见我的Github JPVideoPlayer (<https://github.com/Chris-Pan/JPVideoPlayer>)。
- 2016.11.08
感谢简书作者 @老孟(<http://www.jianshu.com/users/9f6960a40be6/timeline> (<http://www.jianshu.com/users/9f6960a40be6/timeline>)), 他帮我测试了多数的真机设备, 包括iPhone 5s 国行 系统9.3.5 iPhone 6plus 港行 系统10.0.2 iPhone 6s 国行 系统9.3.2 iPhone 6s plus 港行 系统10.0.0 iPhone 7plus 国行 系统10.1.1, 我之前由于手上设备有限, 只测试了 iPhone 6s 和 iPhone 6s plus, 但是 @老孟发 现在较旧设备上有卡顿的现象, 具体表现为播放本地已经缓存的视频的时候会出现2-3秒的假死, 其实是阻塞了主线程. 现在经过修改过后的版本修复了这个问题, 并且以上设备都测试通过, 没有出现卡顿情况.
- 2016.11.10
关闭播放器以后, 视频还在后台播放的bug已经修复提交, 详见JPVideoPlayer (<https://github.com/Chris-Pan/JPVideoPlayer>)。感谢简书朋友@花无缺_ (<http://www.jianshu.com/users/a16a92bff930>)提交的bug.
- 2016.11.18
 - 1.修复了可能出现, 当播放一些特别小的视频文件时, 会出现播放不了的情况.
 - 2.添加了缓存管理的工具类, 你可以调用-getSize:方法异步获取缓存大小. 也可以使用-clearVideoCacheForUrl: 或者 -clearAllVideoCache 方法清除缓存.

我的文章集合

下面这个链接是我所有文章的一个集合目录。这些文章凡是涉及实现的，每篇文章中都有GIT地址，GIT上都有源码。如果某篇文章刚好在你的实际开发中帮到你，又或者提供一种不同的实现思路，让你觉得有用，那就看看这句话“坚持每天点赞的人，99%都是帅哥美女，再也不用单身了😊”

我的文章集合索引 (<http://www.jianshu.com/p/e03cd37db0d5>)

你还可以关注我自己维护的简书专题

(http://www.jianshu.com/users/e2f2d779c022/latest_articles)iOS开发心得 (<http://www.jianshu.com/collection/72d7b853d415>)。这个专题的文章都是实打实的干货。

如果你有问题，除了在文章最后留言，还可以在微博@盼盼_HKbuy (<http://weibo.com/u/5590458451/home?wvr=5>)上给我留言，以及访问我的Github (<https://github.com/Chris-Pan>)。



NewPan (/u/e2f2d779c022)

写了 55918 字，被 866 人关注，获得了 815 个喜欢
(/u/e2f2d779c022)写了 55918 字，被 866 人关注，获得了 815 个喜欢

+ 关注

iOS开发工程师，简书非签约作者，修行中。

分享即打赏，且更有益。

赞赏支持



(/u/cec914f9c9d2)

 喜欢 (/sign_in) | 121



更多分享

(http://cwb.assets.jianshu.io/notes/images/5768863/



(/sign_in)发表评论

60条评论

只看作者

[按喜欢排序](#) [按时间正序](#) [按时间倒序](#)



天马行空得想法 (/u/418fb65c1b19)

5楼 · 2016.09.19 11:23

(/u/418fb65c1b19)

写得很好必须赞👍👍




1人赞



回复

NewPan (/u/e2f2d779c022): @天马行空得想法 (/users/418fb65c1b19) 谢谢🙏

2016.09.19 12:42  回复



添加新评论



蠢稀旺 (/u/876b6b52f389)

2楼 · 2016.09.18 17:13

(/u/876b6b52f389)

NB




赞



回复

NewPan (/u/e2f2d779c022): @蠢稀旺 (/users/876b6b52f389) NewBalance

2016.09.18 23:45  回复



添加新评论



wh5865885 (/u/3bd98e15da29)

4楼 · 2016.09.18 20:47

(/u/3bd98e15da29)

好文章!!!!




赞



回复


NewPan (/u/e2f2d779c022): @wh5865885 (/users/3bd98e15da29) 好文章,齐分享,谢谢鼓励



墓志铭333 (/u/71736d754527)
6楼 · 2016.10.24 11:52

(/u/71736d754527)
好文章刚好用到,已收藏

赞 回复

默數塵封滴曖戀 (/u/7ed0845e887d)
7楼 · 2016.10.31 14:07


(/u/7ed0845e887d)
楼主，你这个视频播放出来没有声音，只有画面

赞 回复

NewPan (/u/e2f2d779c022): @默數塵封滴曖戀 (/users/7ed0845e887d) 我把声音关闭了的，因为在列表中播放不需要声音，你如果在其他页面打开，默认是有声音的，请看一下我的API，谢谢。
2016.10.31 14:59 回复

默數塵封滴曖戀 (/u/7ed0845e887d): @NewPan (/users/e2f2d779c022) 哦哦哦，谢谢哈，我以为是bug，还有个问题就是我在5s上面测试的时候第二个cell总是不显示视频
2016.10.31 15:01 回复

添加新评论

科比布莱恩特 (/u/e1c2420217b8)
8楼 · 2016.11.01 09:12

(/u/e1c2420217b8)
昨天下午我在你的 github 上提了这个播放器的相关issue，早上上班的时候查看了一下发现你把这个issue给删除了，我也是醉了，呵呵

赞 回复

NewPan (/u/e2f2d779c022): @科比布莱恩特 (/users/e1c2420217b8) 我已经回复你了，回复完成我把问题关闭了，请你自己去close里查看，谢谢。
2016.11.01 12:49 回复

添加新评论

raymondchans (/u/d6fdd37d999a)
9楼 · 2016.11.10 09:30

(/u/d6fdd37d999a)
下载了你的demo，十分强大。但发现有些小视频是无法缓存。。。有的缓存成功，播放失败。能帮忙看一下问题吗？http://ocgk7i2aj.bkt.clouddn.com/16cd48c6-5cd2-4540-bb76-81795cc174f2 (http://ocgk7i2aj.bkt.clouddn.com/16cd48c6-5cd2-4540-bb76-81795cc174f2)

赞 回复

flower喜欢你 (/u/b49838cf2eaa): @raymondchans (/users/d6fdd37d999a) 我也是，解决了吗？
2016.11.10 14:54 回复

落叶有殇 (/u/97341aaa1917): @raymondchans (/users/d6fdd37d999a) 我也是怎么解决？
2016.11.10 18:08 回复

NewPan (/u/e2f2d779c022): 晚上回去看下问题出现在哪里，这段时间项目要上线，比较赶，见谅。
2016.11.10 18:11 回复

添加新评论 | 还有12条评论， 展开查看



zmj27404 (/u/60a07299244f)

10楼 · 2016.11.13 16:32

(/u/60a07299244f)

正好能用上，学习了！



赞



回复

NewPan (/u/e2f2d779c022): @zmj27404 (/users/60a07299244f) 好用就帮忙去github上star一下，谢谢

2016.11.13 18:41 回复



添加新评论



哪里有会生气的龙 (/u/371e7dfb9a55)

11楼 · 2016.11.16 08:54

(/u/371e7dfb9a55)

Mark



赞



回复



1a416da6092e (/u/1a416da6092e)

12楼 · 2016.11.18 10:53

(/u/1a416da6092e)

http://xxt1005-

10072067.video.myqcloud.com/2016/tTch/Chat/402962%261479354112007.071045.

mov (http://xxt1005-

10072067.video.myqcloud.com/2016/tTch/Chat/402962%261479354112007.071045.

mov) 这样的视频缓存下来了，播放不了，楼主帮忙看一下是什么情况，项目很急！跪谢



赞



回复

NewPan (/u/e2f2d779c022): @1a416da6092e (/users/1a416da6092e) 类似小视频无法播放的问题已经解决, 请上我的git上查看. 或者直接使用pod 'JPVideoPlayer', '~> 1.3.1'.

2016.11.18 23:27 回复



添加新评论



1a416da6092e (/u/1a416da6092e)

13楼 · 2016.11.18 10:55

(/u/1a416da6092e)

http://xxt1005-

10072067.video.myqcloud.com/iOS/2016/tTch/Chat/402962%261479364102901.707

031.mov (http://xxt1005-

10072067.video.myqcloud.com/iOS/2016/tTch/Chat/402962%261479364102901.707

031.mov) 一样的视频格式，这个就可以，上一个就不行



赞



回复



1a416da6092e (/u/1a416da6092e)

14楼 · 2016.11.18 11:10

(/u/1a416da6092e)

博主，我给你star了，帮我看一下这个问题吧，真的很急，非常非常感谢



赞



回复

NewPan (/u/e2f2d779c022): @1a416da6092e (/users/1a416da6092e) 我正在找有些特别小的视频会出现播放不了的问题, 周末应该会有结果, 你保持关注把.

2016.11.18 12:34 回复

1a416da6092e (/u/1a416da6092e): @NewPan (/users/e2f2d779c022) 谢谢博主，十分感谢

2016.11.18 13:57 回复

NewPan (/u/e2f2d779c022): @1a416da6092e (/users/1a416da6092e) 类似小视频无法播放的问题已经解决, 请上我的git上查看. 或者直接使用pod 'JPVideoPlayer', '~> 1.3.1'.

添加新评论

还有1条评论， 展开查看



低头式高调 (/u/3aef6558f4b7)

15楼 · 2016.11.21 16:11

(/u/3aef6558f4b7)

博主，你好,如果不使用单利,当做独立控件来做怎么用呢,就像微信那样

赞

回复



晓折 (/u/cec914f9c9d2)

16楼 · 2016.12.02 09:08

(/u/cec914f9c9d2)

先看一遍，晚上回去研究。感谢楼主分享。

赞

回复



独自去偷欢_ (/u/721e472431a4)

17楼 · 2016.12.12 22:23

(/u/721e472431a4)

你好，请问有类似进度条滑动快进后退播放视频的功能吗

赞

回复

被以下专题收入，发现更多相似内容

iOS开发
心得

关注

iOS进阶
相关

iOS综合
相关

iOS图形
处理相关

牛叉的
demo

技术交流

iOS开发

iOS

iOS 开发

IOS

iOS进阶

iOS学习

iOS
Dev...

音视频开
发

加载更多...