



This repository Search

Pull requests Issues Gist



You don't have any verified emails. We recommend [verifying](#) at least one email.

Email verification helps our support team verify ownership if you lose account access and allows you to receive all the notifications you ask for.

CoderMJLee / MJRefresh

Watch 317

★ Star 5,424

Fork 1,776

<> Code

Issues 98

Pull requests 7

Wiki

Pulse

Graphs

An easy way to use pull-to-refresh.

208 commits

1 branch

59 releases

12 contributors

Branch: master

New pull request

New file

Find file

HTTPS

https://github.com/CoderMJLee



Download ZIP



CoderMJLee Merge pull request #432 from WenchaoIOS/patch-1 ...

Latest commit 94ccc5d 12 days ago

MJRefresh	解决SectionHeader停留的BUG	13 days ago
MJRefreshExample.xcodeproj	Add prefix 'mj_' into category	a month ago
MJRefreshExample	gif header without images for stateidle	16 days ago
MJRefreshExampleTests	Add prefix 'mj_' into category	a month ago
.gitignore	Nothing important	11 months ago
LICENSE	Modify podspec	11 months ago
MJRefresh.podspec	解决SectionHeader停留的BUG	13 days ago
README.md	Update README.md	12 days ago

README.md



MJRefresh

- An easy way to use pull-to-refresh
- 用法简单的下拉刷新框架：一行代码搞定

Contents

- Getting Started
 - [Features](#) 【支持哪些控件的刷新】
 - [Installation](#) 【如何使用MJRefresh】
 - [Who's using](#) 【已经超过上百个App正在使用MJRefresh】
 - [Classes](#) 【MJRefresh类结构图】
- 常见API
 - [MJRefreshComponent.h](#)

- [MJRefreshHeader.h](#)
- [MJRefreshFooter.h](#)
- [MJRefreshAutoFooter.h](#)
- Examples
 - [Reference 【参考】](#)
 - [下拉刷新01-默认](#)
 - [下拉刷新02-动画图片](#)
 - [下拉刷新03-隐藏时间](#)
 - [下拉刷新04-隐藏状态和时间](#)
 - [下拉刷新05-自定义文字](#)
 - [下拉刷新06-自定义刷新控件](#)
 - [上拉刷新01-默认](#)
 - [上拉刷新02-动画图片](#)
 - [上拉刷新03-隐藏刷新状态的文字](#)
 - [上拉刷新04-全部加载完毕](#)
 - [上拉刷新05-自定义文字](#)
 - [上拉刷新06-加载后隐藏](#)
 - [上拉刷新07-自动回弹的上拉01](#)
 - [上拉刷新08-自动回弹的上拉02](#)
 - [上拉刷新09-自定义刷新控件\(自动刷新\)](#)
 - [上拉刷新10-自定义刷新控件\(自动回弹\)](#)
 - [UICollectionView01-上下拉刷新](#)
 - [UIWebView01-下拉刷新](#)
- [期待](#)

支持哪些控件的刷新

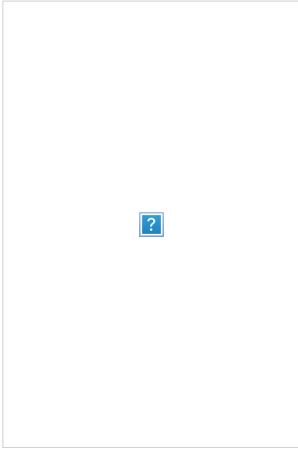
- UIScrollView 、 UITableView 、 UICollectionView 、 UIWebView

如何使用MJRefresh

- cocoapods导入: `pod 'MJRefresh'`
- 手动导入:
 - 将 MJRefresh 文件夹中的所有文件拽入项目中
 - 导入主头文件: `#import "MJRefresh.h"`

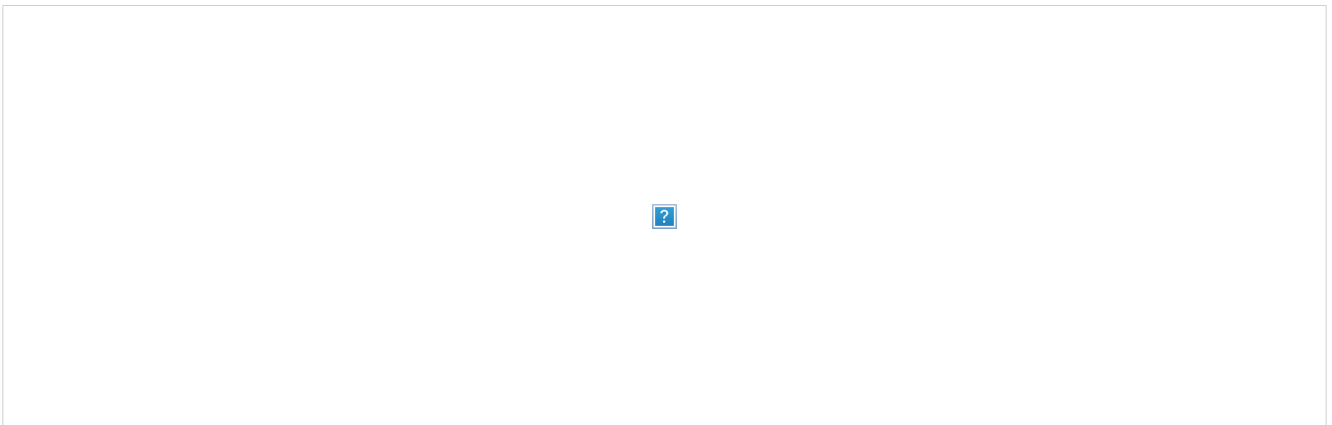
Base	Custom
MJRefresh.bundle	MJRefresh.h
MJRefreshConst.h	MJRefreshConst.m
UIScrollView+MJExtension.h	UIScrollView+MJExtension.m
UIScrollView+MJRefresh.h	UIScrollView+MJRefresh.m
UIView+MJExtension.h	UIView+MJExtension.m

已经超过上百个App正在使用MJRefresh

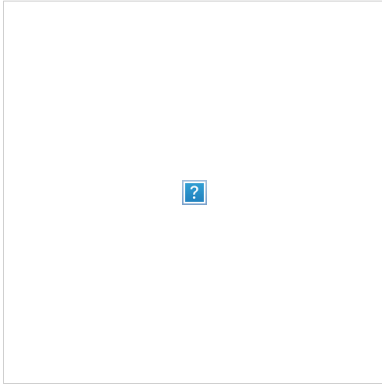


- 更多App信息可以关注：[M了个J-博客园](#)

MJRefresh类结构图



- 图中 红色文字的类：可以直接拿来用
 - 下拉刷新控件的种类
 - 默认 (Normal)：MJRefreshNormalHeader
 - 动图 (Gif)：MJRefreshGifHeader
 - 上拉刷新控件的种类
 - 自动刷新 (Auto)
 - 默认 (Normal)：MJRefreshAutoNormalFooter
 - 动图 (Gif)：MJRefreshAutoGifFooter
 - 自动回弹 (Back)
 - 默认 (Normal)：MJRefreshBackNormalFooter
 - 动图 (Gif)：MJRefreshBackGifFooter
- 图中 非红色文字的类：拿来继承，用于自定义刷新控件
- 关于如何自定义刷新控件，可以参考下图的类



MJRefreshComponent.h

```
/** 刷新控件的基类 */
@interface MJRefreshComponent : UIView
#pragma mark - 刷新状态控制
/** 进入刷新状态 */
- (void)beginRefreshing;
/** 结束刷新状态 */
- (void)endRefreshing;
/** 是否正在刷新 */
- (BOOL)isRefreshing;

#pragma mark - 其他
/** 根据拖拽比例自动切换透明度 */
@property (assign, nonatomic, getter=isAutomaticallyChangeAlpha) BOOL automaticallyChangeAlpha;
@end
```

MJRefreshHeader.h

```
@interface MJRefreshHeader : MJRefreshComponent
/** 创建header */
+ (instancetype)headerWithRefreshingBlock:(MJRefreshComponentRefreshingBlock)refreshingBlock;
/** 创建header */
+ (instancetype)headerWithRefreshingTarget:(id)target refreshingAction:(SEL)action;

/** 这个key用来存储上一次下拉刷新成功的时间 */
@property (copy, nonatomic) NSString *lastUpdateTimeKey;
/** 上一次下拉刷新成功的时间 */
@property (strong, nonatomic, readonly) NSDate *lastUpdateTime;

/** 忽略多少scrollView的contentInset的top */
@property (assign, nonatomic) CGFloat ignoredScrollViewContentInsetTop;
@end
```

MJRefreshFooter.h

```
@interface MJRefreshFooter : MJRefreshComponent
/** 创建footer */
+ (instancetype)footerWithRefreshingBlock:(MJRefreshComponentRefreshingBlock)refreshingBlock;
/** 创建footer */
+ (instancetype)footerWithRefreshingTarget:(id)target refreshingAction:(SEL)action;

/** 提示没有更多的数据 */
```

```
- (void)endRefreshingWithNoMoreData;
/** 重置没有更多的数据（消除没有更多数据的状态） */
- (void)resetNoMoreData;

/** 忽略多少scrollView的contentInset的bottom */
@property (assign, nonatomic) CGFloat ignoredScrollViewContentInsetBottom;

/** 自动根据有无数据来显示和隐藏（有数据就显示，没有数据隐藏） */
@property (assign, nonatomic) BOOL automaticallyHidden;
@end
```

MJRefreshAutoFooter.h

```
@interface MJRefreshAutoFooter : MJRefreshFooter
/** 是否自动刷新（默认为YES） */
@property (assign, nonatomic, getter=isAutomaticallyRefresh) BOOL automaticallyRefresh;

/** 当底部控件出现多少时就自动刷新（默认为1.0，也就是底部控件完全出现时，才会自动刷新） */
@property (assign, nonatomic) CGFloat triggerAutomaticallyRefreshPercent;
@end
```

参考

- * 由于这个框架的功能较多，就不写具体文字描述其用法
- * 大家可以直接参考示例中的MJTableViewController、MJCollectionViewController、MJWebViewController，更为直观快速



下拉刷新01-默认

```
self.tableView.header = [MJRefreshNormalHeader headerWithRefreshingBlock:^(
    // 进入刷新状态后会自动调用这个block
)];
```

或

// 设置回调（一旦进入刷新状态，就调用target的action，也就是调用self的loadNewData方法）

```
self.tableView.header = [MJRefreshNormalHeader headerWithRefreshingTarget:self refreshingAction:@selector(loadNewData)];
```

// 马上进入刷新状态

```
[self.tableView.header beginRefreshing];
```

(下拉刷新01-普通)



下拉刷新02-动画图片

// 设置回调（一旦进入刷新状态，就调用target的action，也就是调用self的loadNewData方法）

```
MJRefreshGifHeader *header = [MJRefreshGifHeader headerWithRefreshingTarget:self refreshingAction:@selector(loadNewData)];
```

// 设置普通状态的动画图片

```
[header setImages:idleImages forState:MJRefreshStateIdle];
```

// 设置即将刷新状态的动画图片（一松开就会刷新的状态）

```
[header setImages:pullingImages forState:MJRefreshStatePulling];
```

// 设置正在刷新状态的动画图片

```
[header setImages:refreshingImages forState:MJRefreshStateRefreshing];
```

// 设置header

```
self.tableView.mj_header = header;
```

(下拉刷新02-动画图片)



下拉刷新03-隐藏时间

```
// 隐藏时间
header.lastUpdateTimeLabel.hidden = YES;
```

(下拉刷新03-隐藏时间)



下拉刷新04-隐藏状态和时间

```
// 隐藏时间
header.lastUpdateTimeLabel.hidden = YES;

// 隐藏状态
header.stateLabel.hidden = YES;
```


(下拉刷新04-隐藏状态和时间0)



下拉刷新05-自定义文字

```
// 设置文字
[header setTitle:@"Pull down to refresh" forState:MJRefreshStateIdle];
[header setTitle:@"Release to refresh" forState:MJRefreshStatePulling];
[header setTitle:@"Loading ..." forState:MJRefreshStateRefreshing];

// 设置字体
header.stateLabel.font = [UIFont systemFontOfSize:15];
header.lastUpdatedTimeLabel.font = [UIFont systemFontOfSize:14];

// 设置颜色
header.stateLabel.textColor = [UIColor redColor];
header.lastUpdatedTimeLabel.textColor = [UIColor blueColor];
```

(下拉刷新05-自定义文字)



下拉刷新06-自定义刷新控件

```
self.tableView.mj_header = [MJDIYHeader headerWithRefreshingTarget:self refreshingAction:@selector(loadMoreData)]
// 具体实现参考MJDIYHeader.h和MJDIYHeader.m
```

(下拉刷新06-自定义刷新控件)



上拉刷新01-默认

```
self.tableView.mj_footer = [MJRefreshAutoNormalFooter footerWithRefreshingBlock:^(
    // 进入刷新状态后会自动调用这个block
)];
或
// 设置回调（一旦进入刷新状态，就调用target的action，也就是调用self的loadMoreData方法）
self.tableView.mj_footer = [MJRefreshAutoNormalFooter footerWithRefreshingTarget:self refreshingAction:(
```

(上拉刷新01-默认)



上拉刷新02-动画图片

```
// 设置回调（一旦进入刷新状态，就调用target的action，也就是调用self的loadMoreData方法）
MJRefreshAutoGifFooter *footer = [MJRefreshAutoGifFooter footerWithRefreshingTarget:self refreshingAction:self];

// 设置刷新图片
[footer setImages:refreshingImages forState:MJRefreshStateRefreshing];

// 设置尾部
self.tableView.mj_footer = footer;
```

(上拉刷新02-动画图片)



上拉刷新03-隐藏刷新状态的文字

```
// 隐藏刷新状态的文字
footer.refreshingTitleHidden = YES;
// 如果没有上面的方法，就用footer.stateLabel.hidden = YES;
```

(上拉刷新03-隐藏刷新状态的文字)



上拉刷新04-全部加载完毕

```
// 变为没有更多数据的状态  
[footer endRefreshingWithNoMoreData];
```

(上拉刷新04-全部加载完毕)



上拉刷新05-自定义文字

```
// 设置文字
[footer setTitle:@"Click or drag up to refresh" forState:MJRefreshStateIdle];
[footer setTitle:@"Loading more ..." forState:MJRefreshStateRefreshing];
[footer setTitle:@"No more data" forState:MJRefreshStateNoMoreData];

// 设置字体
footer.stateLabel.font = [UIFont systemFontOfSize:17];

// 设置颜色
footer.stateLabel.textColor = [UIColor blueColor];
```

(上拉刷新05-自定义文字)



上拉刷新06-加载后隐藏

```
// 隐藏当前的上拉刷新控件  
self.tableView.mj_footer.hidden = YES;
```


(上拉刷新06-加载后隐藏)



上拉刷新07-自动回弹的上拉01

```
self.tableView.mj_footer = [MJRefreshBackNormalFooter footerWithRefreshingTarget:self refreshingAction:(
```



(上拉刷新07-自动回弹的上拉01)



上拉刷新08-自动回弹的上拉02

```
MJRefreshBackGifFooter *footer = [MJRefreshBackGifFooter footerWithRefreshingTarget:self refreshingActio

// 设置普通状态的动画图片
[footer setImages:idleImages forState:MJRefreshStateIdle];
// 设置即将刷新状态的动画图片（一松开就会刷新的状态）
[footer setImages:pullingImages forState:MJRefreshStatePulling];
// 设置正在刷新状态的动画图片
[footer setImages:refreshingImages forState:MJRefreshStateRefreshing];

// 设置尾部
self.tableView.mj_footer = footer;
```

(上拉刷新07-自动回弹的上拉02)



上拉刷新09-自定义刷新控件(自动刷新)

```
self.tableView.mj_footer = [MJDIYAutoFooter footerWithRefreshingTarget:self refreshingAction:@selector(1  
// 具体实现参考MJDIYAutoFooter.h和MJDIYAutoFooter.m
```

(上拉刷新09-自定义刷新控件(自动刷新))



上拉刷新10-自定义刷新控件(自动回弹)

```
self.tableView.mj_footer = [MJDIYBackFooter footerWithRefreshingTarget:self refreshingAction:@selector(1  
// 具体实现参考MJDIYBackFooter.h和MJDIYBackFooter.m
```

(上拉刷新10-自定义刷新控件(自动回弹))



UICollectionView01-上下拉刷新

```
// 下拉刷新
self.collectionView.mj_header = [MJRefreshNormalHeader headerWithRefreshingBlock:^(
    // 进入刷新状态后会自动调用这个block
)];

// 上拉刷新
self.collectionView.mj_footer = [MJRefreshAutoNormalFooter footerWithRefreshingBlock:^(
    // 进入刷新状态后会自动调用这个block
)];
```

(UICollectionView01-上下拉刷新)



UIWebView01-下拉刷新

```
// 添加下拉刷新控件
self.webView.scrollView.mj_header = [MJRefreshNormalHeader headerWithRefreshingBlock:^(
    // 进入刷新状态后会自动调用这个block
)];
```

(UICollectionView01-上下拉刷新)



提醒

- 本框架纯ARC，兼容的系统>=iOS6.0、iPhone\iPad横竖屏

期待

- 如果在使用过程中遇到BUG，希望你能Issues我，谢谢（或者尝试下载最新的框架代码看看BUG修复没有）
- 如果在使用过程中发现功能不够用，希望你能Issues我，我非常想为这个框架增加更多好用的功能，谢谢
- 如果你想为MJRefresh输出代码，请拼命Pull Requests我
- 如果你开发的应用中用到了MJRefresh，希望你能到[CocoaControls](#)添加你应用的iTunes路径，我将会安装使用你的应用，并且根据众多应用的使用情况，对MJRefresh进行一个更好的设计和完善，提供更多好用的功能，谢谢
 - 步骤01（微信是举个例子，百度“你的应用名称 itunes”）

(step01)



◦ 步骤02

(step02)



◦ 步骤03

(step03)



◦ 步骤04

(step04)

