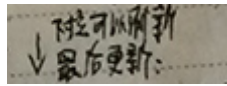


MJRefresh 用法熟悉

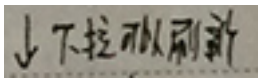
UITableView+下拉刷新

1. 默认下拉刷新:



```
__weak typeof(self) weakSelf = self;
//手动进入刷新状态
self.tableView.mj_header = [MJRefreshNormalHeader headerWithRefreshingBlock:^(
    [weakSelf.tableView reloadData];
)];
//马上进入刷新状态
[self.tableView.mj_header beginRefreshing];
```

2. 隐藏时间:



```
MJRefreshNormalHeader *header = [MJRefreshNormalHeader headerWithRefreshingTarget:self
    refreshingAction:@selector(loadNewData)];
//设置自动切换透明度 (复位完全透明)
header.automaticallyChangeAlpha = YES;
//隐藏时间
header.lastUpdatedTimeLabel.hidden = YES;
self.tableView.mj_header = header;
```

3. 隐藏状态和时间:



```
MJRefreshNormalHeader *header = [MJRefreshNormalHeader headerWithRefreshingTarget:self
    refreshingAction:@selector(loadNewData)];
//设置自动切换透明度 (复位完全透明)
header.automaticallyChangeAlpha = YES;
//隐藏时间
header.lastUpdatedTimeLabel.hidden = YES;
//隐藏状态
header.stateLabel.hidden = YES;
self.tableView.mj_header = header;
[header beginRefreshing];
```

4. 自定义文字

```
MJRefreshNormalHeader *header = [MJRefreshNormalHeader headerWithRefreshingTarget:self
    refreshingAction:@selector(loadNewData)];
//设置文字
[header setTitle:@"Pull down to refresh" forState:MJRefreshStateIdle];
[header setTitle:@"Release to refresh" forState:MJRefreshStatePulling];
[header setTitle:@"Loading..." forState:MJRefreshStateRefreshing];
self.tableView.mj_header = header;
[header beginRefreshing];
```

5. 自定义 gif 刷新动画



```
#import "MJRefreshGifHeader.h"

@interface MJChiBaoZiHeader : MJRefreshGifHeader

@end

@implementation MJChiBaoZiHeader
#pragma mark - 重写方法
#pragma mark 基本设置
- (void)prepare
{
    [super prepare];

    // 设置普通状态的动画图片
    NSMutableArray *idleImages = [NSMutableArray array];
    for (NSUInteger i = 1; i <= 60; i++) {
        UIImage *image = [UIImage imageNamed:[NSString stringWithFormat:@"dropdown_anim_000%d", i]];
        [idleImages addObject:image];
    }
    [self setIdImages: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）
    NSMutableArray *refreshingImages = [NSMutableArray array];
    for (NSUInteger i = 1; i <= 3; i++) {
        UIImage *image = [UIImage imageNamed:[NSString stringWithFormat:@"dropdown_loading_0%d", i]];
        [refreshingImages addObject:image];
    }
    [self setIdImages:refreshingImages forState:MJRefreshStatePulling];

    // 设置正在刷新状态的动画图片
    [self setIdImages:refreshingImages forState:MJRefreshStateRefreshing];
}
@end
```

```
// 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）
MJChiBaoZiHeader *header = [MJChiBaoZiHeader headerWithRefreshingTarget:self refreshingAction:@selector
    (loadNewData)];
// 隐藏时间
header.lastUpdatedTimeLabel.hidden = YES;
// 隐藏状态
header.stateLabel.hidden = YES;
// 马上进入刷新状态
[header beginRefreshing];
// 设置header
self.tableView.mj_header = header;
```

6. 自定义普通刷新控件

```
@interface MYWaterView : UIView

@property (strong, nonatomic) UIColor *fillColor;//填充色
@property (assign, nonatomic) CGFloat currentLinePointY;//线高

- (void)pauseTimer;

- (void)resumeTimer;

@end
```

```
@interface MYWaterView ()

@property (strong, nonatomic) CADisplayLink *timer;//重绘计时器
@property (assign, nonatomic) CFTimeInterval startTime;//计时开始时间记录
@property (assign, nonatomic) CFTimeInterval deltaTime;//计时时间记录
@property (assign, nonatomic) CGFloat a;
@property (assign, nonatomic) CGFloat b;
@property (assign, nonatomic) BOOL jia;

@end

@implementation MYWaterView

- (void)dealloc{
    [_timer removeFromRunLoop:[NSRunLoop currentRunLoop] forMode:NSRunLoopCommonModes];
}

- (instancetype)initWithFrame:(CGRect)frame {
    if(self = [super initWithFrame:frame]){
        [self initData];
    }
    return self;
}
```

```
- (instancetype)initWithCoder:(NSCoder *)aDecoder{
    if(self = [super initWithCoder:aDecoder]){
        [self initData];
    }
    return self;
}
```

```
- (void)initData{
    _fillColor = [UIColor colorWithRed:21/255.0 green:169/255.0 blue:188/255.0 alpha:1];
    _currentLinePointY = self.frame.size.height;

    _a = 8;
    _b = 0;
    _jia = NO;

    _timer = [CADisplayLink displayLinkWithTarget:self selector:@selector(waterWaving)];
    _startTime = 0.0;
    _deltaTime = 0.0;
    [self resumeTimer];

    [_timer addToRunLoop:[NSRunLoop currentRunLoop] forMode:NSRunLoopCommonModes];
}

- (void)drawRect:(CGRect)rect {
    CGContextRef context = UIGraphicsGetCurrentContext();
    [self updateLevel:context andRect:rect];
}
```

```

- (void)updateLevel:(CGContextRef)context andRect:(CGRect)rect{
    float y = _currentLinePointY;
    //绘图
    CGContextMoveToPoint(context, 0, y);
    for(int x = 0; x <= rect.size.width; x ++){
        y = _a * sin(1.0/100*M_PI*x + 4*_b/M_PI) + _currentLinePointY;
        CGContextAddLineToPoint(context, x, y);
    }

    CGContextAddLineToPoint(context, rect.size.width, rect.size.height);
    CGContextAddLineToPoint( context, 0, rect.size.height);
    CGContextClosePath(context);

    [_fillColor set];//描色+填充色

    CGContextDrawPath(context, kCGPathFillStroke);//描线+填充路径
}

- (void)resumeTimer{
    if(!_timer.isPaused)
        [_timer setPaused:NO];
}

```

```

- (void)pauseTimer{
    [_timer setPaused:YES];
}

```

```

- (void)waterWaving:(CADisplayLink *)sender{
    if(_startTime == 0)
        _startTime = sender.timestamp;

    _deltaTime = sender.timestamp - _startTime;
    if(_deltaTime >= 0.0066){

        [self setNeedsDisplay];

        if (_jia) //曲线振幅大小变化
            _a += 0.01;
        else
            _a -= 0.01;

        if (_a <= 2)
            _jia = YES;
        else if (_a >= 8)
            _jia = NO;

        _b += 0.1;//曲线不断左移动

        _startTime = sender.timestamp;
    }
}

```

```

#import <UIKit/UIKit.h>
#import "MJRefreshHeader.h"

@interface MYRefreshHeader :MJRefreshHeader

@end

```

```

#import "MYRefreshHeader.h"
#import "MYWaterView.h"
#import "UIView+YXHView.h"

@interface MYRefreshHeader ()

@property (strong, nonatomic) CALayer *maskLayer;//遮掩图层
@property (strong, nonatomic) MYWaterView *waterView;
@property (strong, nonatomic) UIImageView *imageView;

@end

```

```

#pragma mark - 重写方法
#pragma mark 在这里做一些初始化配置（比如添加子控件）
- (void)prepare{
    [super prepare];

    UIImage *image = [UIImage imageNamed:@"shuaxin"];

    _imageView = [[UIImageView alloc] init];
    [_imageView setTranslatesAutoresizingMaskIntoConstraints:NO];
    _imageView.image = image;

    _waterView = [[MYWaterView alloc] init];
    [_waterView setTranslatesAutoresizingMaskIntoConstraints:NO];
    [_waterView setBackgroundColor:[UIColor clearColor]];

    CALayer *maskLayer = [[CALayer alloc] init];
    maskLayer.frame = CGRectMake(0, 0, 48, 30);
    maskLayer.contents = (__bridge id _Nullable)([UIImage imageNamed:@"shuaxin"].CGImage);
    [_waterView.layer setMask:maskLayer];

    [self addSubview:_imageView];
    [self addSubview:_waterView];
}

```

#pragma mark 在这里设置子控件的位置和尺寸

```

- (void)placeSubviews{
    [super placeSubviews];
    [_waterView setLayoutCenterX:self];
    [_waterView setLayoutCenterY:self];
    [_waterView setLayoutWidth:48];
    [_waterView setLayoutHeight:30];

    [_imageView setLayoutCenterX:self];
    [_imageView setLayoutCenterY:self];
    [_imageView setLayoutWidth:48];
    [_imageView setLayoutHeight:30];
}

```

#pragma mark 监听scrollView的contentOffset改变

```

- (void)scrollViewContentOffsetDidChange:(NSDictionary *)change{
    [super scrollViewContentOffsetDidChange:change];
    CGPoint newValue = [change[@"new"] CGPointValue];
    CGFloat newValueY = newValue.y;
    self.waterView.currentLinePointY = 58+newValueY;
}

```

#pragma mark 监听scrollView的contentSize改变

```

- (void)scrollViewContentSizeDidChange:(NSDictionary *)change{
    [super scrollViewContentSizeDidChange:change];
}

```



```

#pragma mark 监听scrollView的拖拽状态改变
- (void)scrollViewPanStateDidChange:(NSDictionary *)change {
    [super scrollViewPanStateDidChange:change];
}

#pragma mark 监听控件的刷新状态
- (void)setState:(MJRefreshState)state {
    MJRefreshCheckState;

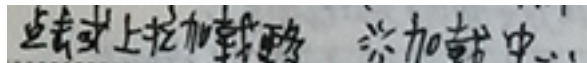
    switch (state) {
        case MJRefreshStateIdle:
            [_waterView resumeTimer];
            break;
        case MJRefreshStatePulling:
            break;
        case MJRefreshStateRefreshing:
            break;
        default:
            break;
    }
}

#pragma mark 监听拖拽比例（控件被拖出来的比例）
- (void)setPullingPercent:(CGFloat)pullingPercent {
    [super setPullingPercent:pullingPercent];
}

```

UITableView+上拉刷新

1. 默认上拉刷新：



```

self.tableView.mj_footer = [MJRefreshAutoNormalFooter footerWithRefreshingBlock:^(
    [weakSelf loadMoreData];
)];

```

2. 隐藏刷新状态文字：

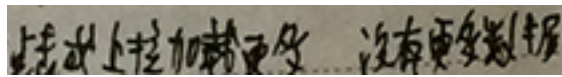


```

// 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）
MJChiBaoZiFooter *footer = [MJChiBaoZiFooter footerWithRefreshingTarget:self refreshingAction:@selector
    (loadMoreData)];
// 当上拉刷新控件出现50%时（出现一半），就会自动刷新。这个值默认是1.0（也就是上拉刷新100%出现时，才会
    自动刷新）
// footer.triggerAutomaticallyRefreshPercent = 0.5;
// 隐藏刷新状态的文字
footer.refreshingTitleHidden = YES;
// 设置footer
self.tableView.mj_footer = footer;

```


3. 全部加载完毕:



```
//判断无更多数据后
[self.tableView.mj_footer endRefreshingWithNoMoreData];
//注: 在下拉刷新时需重置无更多数据状态
[self.tableView.mj_footer resetNoMoreData];
```

4. 禁止上滑时自动加载

```
MJRefreshAutoNormalFooter *footer = [MJRefreshAutoNormalFooter footerWithRefreshingTarget:self
refreshingAction:@selector(loadMoreData)];
//禁止上滑时自动加载
footer.automaticallyRefresh = NO;
self.tableView.mj_footer = footer;
```

5. 自定义文字

```
MJRefreshAutoNormalFooter *footer = [MJRefreshAutoNormalFooter footerWithRefreshingTarget:self
refreshingAction:@selector(loadMoreData)];
//禁止上滑时自动加载
footer.automaticallyRefresh = NO;
[footer setTitle:@"" forState:MJRefreshStateIdle];
[footer setTitle:@"" forState:MJRefreshStatePulling];
[footer setTitle:@"" forState:MJRefreshStateRefreshing];
self.tableView.mj_footer = footer;
```

6. 全部加载完成后隐藏 footer

```
//判断全部加载完成后
[self.tableView.mj_footer setHidden:YES];
```

7. 自定义 gif 刷新动画

```
@interface MJChiBaoZiFooter2 : MJRefreshBackGifFooter
```

```
- (void)prepare
{
    [super prepare];

    // 设置普通状态的动画图片
    NSMutableArray *idleImages = [NSMutableArray array];
    for (NSUInteger i = 1; i <= 60; i++) {
        UIImage *image = [UIImage imageNamed:[NSString stringWithFormat:@"dropdown_anim_000%zd", i]];
        [idleImages addObject:image];
    }
    [self setImages:idleImages forState:MJRefreshStateIdle];
}
```

8. 自定义普通刷新控件：继承 MJRefreshBackFooter，参照 header 实现