

【阿峥教你实现UITableView循环利用】 |那些人追的干货



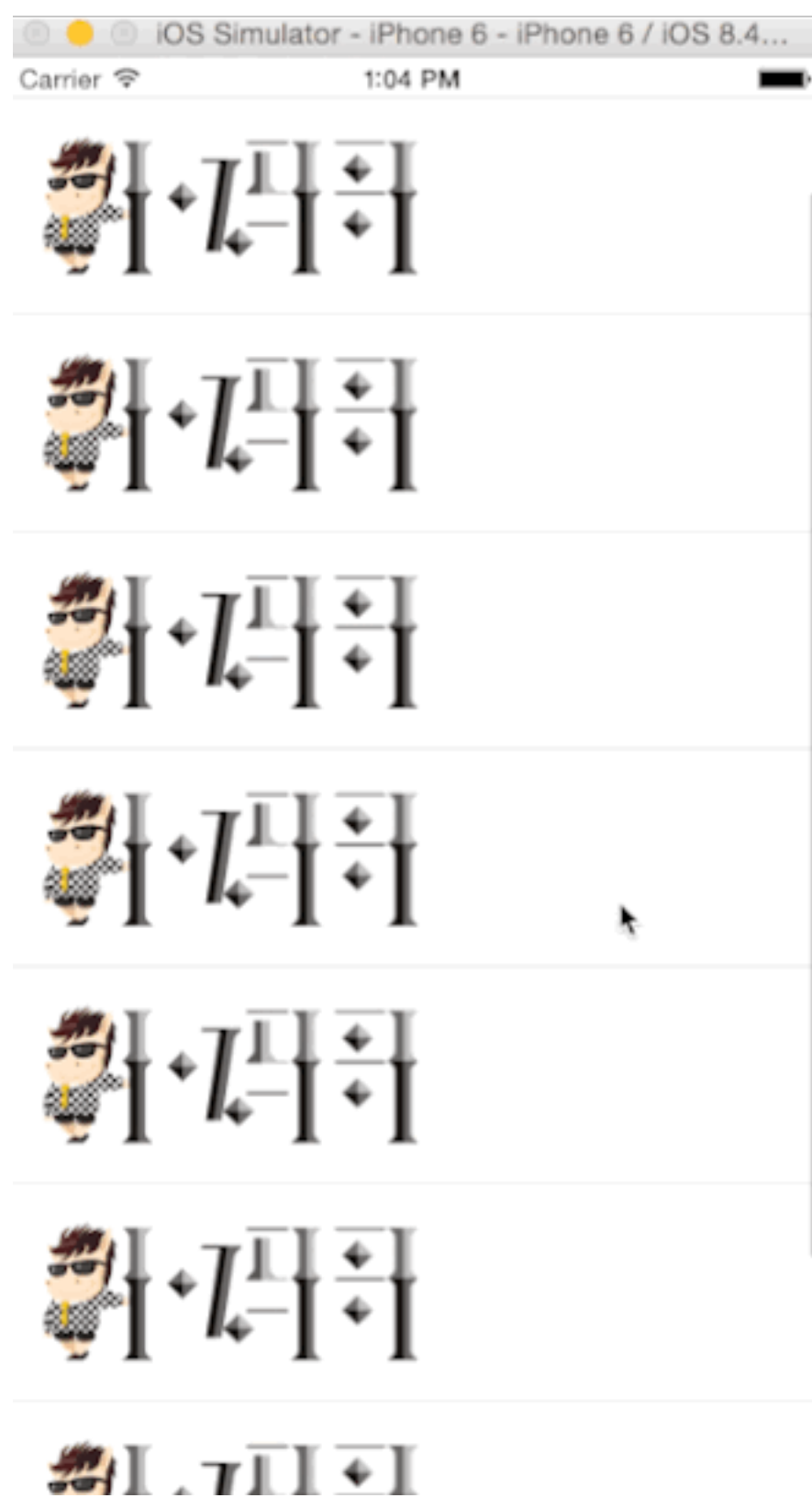
作者 袁峥Seemygo (/u/b09c3959ab3b) ✓ 已关注

2015.08.17 13:19* 字数 3424 阅读 10646 评论 39 喜欢 105
(/u/b09c3959ab3b)

前言

大家都知道UITableView，最经典在于循环利用，这里我自己模仿UITableView循环利用，写了一套自己的TableView实现方案，希望大家看了我的文章，循环利用思想有显著提升。

效果如图：



tableView效果.gif

如果喜欢我的文章，可以关注我，也可以来小码哥 (<http://www.520it.com>)，了解下我们的iOS培训课程。陆续还会有更新ing....

研究UITableView底层实现

1.系统UITabelView的简单使用，这里就不考虑分组了，默认为1组。

```
// 返回第section组有多少行
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)
section
{
    NSLog(@"%s",__func__);
    return 10;
}

// 返回每一行cell的样子
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NS
IndexPath *)indexPath
{
    NSLog(@"%s",__func__);
    static NSString *ID = @"cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:ID];

    if (cell == nil) {

        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:ID];

    }

    cell.textLabel.text = [NSString stringWithFormat:@"%ld",indexPath.row];

    return cell;
}

// 返回每行cell的高度
- (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPat
h *)indexPath
{
    NSLog(@"%s--%@",__func__,indexPath);
    return 100;
}
```

2.验证UITabelView的实现机制。

如图打印结果:

```
2015-08-08 23:07:12.210 UITableView循环利用实现原理[1385:55813] -[ViewController tableView:numberOfRowsInSection:]
2015-08-08 23:07:12.210 UITableView循环利用实现原理[1385:55813] -[ViewController
tableView:heightForRowAtIndexPath:]--<NSIndexPath 0x7f9ec0fb25f0> 2 indexes [0, 0]
2015-08-08 23:07:12.211 UITableView循环利用实现原理[1385:55813] -[ViewController
tableView:heightForRowAtIndexPath:]--<NSIndexPath 0x7f9ec0fb2dc0> 2 indexes [0, 1]
2015-08-08 23:07:12.211 UITableView循环利用实现原理[1385:55813] -[ViewController
tableView:heightForRowAtIndexPath:]--<NSIndexPath 0x7f9ec0fb25f0> 2 indexes [0, 2]
2015-08-08 23:07:12.211 UITableView循环利用实现原理[1385:55813] -[ViewController
tableView:heightForRowAtIndexPath:]--<NSIndexPath 0x7f9ec0fb2dc0> 2 indexes [0, 3]
2015-08-08 23:07:12.211 UITableView循环利用实现原理[1385:55813] -[ViewController
tableView:heightForRowAtIndexPath:]--<NSIndexPath 0x7f9ec0fb25f0> 2 indexes [0, 4]
2015-08-08 23:07:12.211 UITableView循环利用实现原理[1385:55813] -[ViewController
tableView:heightForRowAtIndexPath:]--<NSIndexPath 0x7f9ec0fb2dc0> 2 indexes [0, 5]
2015-08-08 23:07:12.220 UITableView循环利用实现原理[1385:55813] -[ViewController
tableView:heightForRowAtIndexPath:]--<NSIndexPath 0x7f9ec0c05420> 2 indexes [0, 6]
2015-08-08 23:07:12.220 UITableView循环利用实现原理[1385:55813] -[ViewController
tableView:heightForRowAtIndexPath:]--<NSIndexPath 0x7f9ec0d01740> 2 indexes [0, 7]
2015-08-08 23:07:12.220 UITableView循环利用实现原理[1385:55813] -[ViewController
tableView:heightForRowAtIndexPath:]--<NSIndexPath 0x7f9ec0d00810> 2 indexes [0, 8]
2015-08-08 23:07:12.220 UITableView循环利用实现原理[1385:55813] -[ViewController
tableView:heightForRowAtIndexPath:]--<NSIndexPath 0x7f9ec0d01740> 2 indexes [0, 9]
```

Snip20150808_3.png

分析：底层先获取有多少cell（10个），在获取每个cell的高度，返回高度的方法一开始调用10次。

目的： 确定tableView的滚动范围，一开始计算所有cell的frame,就能计算下tableView的滚动范围。

分析：tableView:cellForRowAtIndexPath:方法什么时候调用。

打印验证，如图：

```
2015-08-08 23:10:17.508 UITableView循环利用实现原理[1443:58057] -[ViewController
tableView:cellForRowAtIndexPath:]--<NSIndexPath: 0xc000000000000016> {length = 2, path = 0 - 0}
2015-08-08 23:10:17.512 UITableView循环利用实现原理[1443:58057] -[ViewController
tableView:cellForRowAtIndexPath:]--<NSIndexPath: 0xc000000000000016> {length = 2, path = 0 - 1}
2015-08-08 23:10:17.514 UITableView循环利用实现原理[1443:58057] -[ViewController
tableView:cellForRowAtIndexPath:]--<NSIndexPath: 0xc0000000000010016> {length = 2, path = 0 - 2}
2015-08-08 23:10:17.514 UITableView循环利用实现原理[1443:58057] -[ViewController
tableView:cellForRowAtIndexPath:]--<NSIndexPath: 0xc0000000000018016> {length = 2, path = 0 - 3}
2015-08-08 23:10:17.515 UITableView循环利用实现原理[1443:58057] -[ViewController
tableView:cellForRowAtIndexPath:]--<NSIndexPath: 0xc0000000000020016> {length = 2, path = 0 - 4}
2015-08-08 23:10:17.515 UITableView循环利用实现原理[1443:58057] -[ViewController
tableView:cellForRowAtIndexPath:]--<NSIndexPath: 0xc0000000000028016> {length = 2, path = 0 - 5}
2015-08-08 23:10:17.515 UITableView循环利用实现原理[1443:58057] -[ViewController
tableView:cellForRowAtIndexPath:]--<NSIndexPath: 0xc0000000000030016> {length = 2, path = 0 - 6}
```


一开始调用了7次，因为一开始屏幕最多显示7个cell
目的：一开始只加载显示出来的cell，等有新的cell出现的时候会继续调用这个方法加载cell。

3.UITableView循环利用思想

当新的cell出现的时候，首先从缓存池中获取，如果没有获取到，就自己创建cell。
当有cell移除屏幕的时候，把cell放到缓存池中去。

二、自定义UIScrollView，模仿UITableView循环利用

1. 提供数据源和代理方法，命名和UITableView一致。

```
@class YZTableView;
@protocol YZTableViewDataSource<NSObject>

@required

// 返回有多少行cell
- (NSInteger)tableView:(YZTableView *)tableView numberOfRowsInSectionInSection:(NSInteger)section;

// 返回每行cell长什么样子
- (UITableViewCell *)tableView:(YZTableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath;

@end

@protocol YZTableViewDelegate<NSObject, UIScrollViewDelegate>

// 返回每行cell有多高
- (CGFloat)tableView:(YZTableView *)tableView heightForRowAtIndexPath:(NSIndexPath *)indexPath;

@end
```

2. 提供代理和数据源属性

```
@interface YZTableView : UIScrollView

@property (nonatomic, weak) id<YZTableViewDataSource> dataSource;

@property (nonatomic, weak) id<YZTableViewDelegate> delegate;

@end
```

警告:



解决，在YZTableView.m的实现中声明。



原因：有人会问为什么我要定义同名的delegate属性，我主要想模仿系统的tableView，系统tableView也有同名的属性。

思路：这样做，外界在使用设置我的tableView的delegate，就必须遵守的我的代理协议，而不是UIScrollView的代理协议。

3. 提供刷新方法reloadData，因为tableView通过这个刷新tableView。

```
@interface YZTableView : UIScrollView

@property (nonatomic, weak) id<YZTableViewDataSource> dataSource;

@property (nonatomic, weak) id<YZTableViewDelegate> delegate;

// 刷新tableView
- (void)reloadData;

@end
```

4. 实现reloadData方法，刷新表格

- 回顾系统如何刷新tableView
 - 1.先获取有多少cell,在获取每个cell的高度。因此应该是先计算出每个cell的frame.
 - 2.然后再判断当前有多少cell显示在屏幕上，就加载多少

```

// 刷新tableView
- (void)reloadData
{
    // 这里不考虑多组，假设tableView默认只有一组。

    // 先获取总共有多少cell
    NSInteger rows = [self.dataSource tableView:self numberOfRowsInSection:0];

    // 遍历所有cell的高度，计算每行cell的frame
    CGRect cellF;
    CGFloat cellX = 0;
    CGFloat cellY = 0;
    CGFloat cellW = self.bounds.size.width;
    CGFloat cellH = 0;

    CGFloat totalH = 0;

    for (int i = 0; i < rows; i++) {
        NSIndexPath *indexPath = [NSIndexPath indexPathForRow:i inSection:0];
        // 注意：这里获取的delegate，是UIScrollView中声明的属性
        if ([self.delegate respondsToSelector:@selector(tableView:heightForRowAtIndexPath:)]) {
            cellH = [self.delegate tableView:self heightForRowAtIndexPath:indexPath];
        } else {
            cellH = 44;
        }
        cellY = i * cellH;

        cellF = CGRectMake(cellX, cellY, cellW, cellH);

        // 记录每个cell的y值对应的indexPath
        self.indexPathDict[@(cellY)] = indexPath;

        // 判断有多少cell显示在屏幕上，只加载显示在屏幕上的cell
        if ([self isInScreen:cellF]) { // 当前cell的frame在屏幕上
            // 通过数据源获取cell
            UITableViewCell *cell = [self.dataSource tableView:self cellForRowAtIndexPath:indexPath];

            cell.frame = cellF;

            [self addSubview:cell];
        }

        // 添加分割线
        UIView *divideV = [[UIView alloc] initWithFrame:CGRectMake(0, cellY + cellH - 1, cellW, 1)];
        divideV.backgroundColor = [UIColor lightGrayColor];
        divideV.alpha = 0.3;
        [self addSubview:divideV];

        // 添加到cell可见数组中
        [self.visibleCells addObject:cell];

        // 计算tableView内容总高度
        totalH += cellY + cellH;
    }

    // 设置tableView的滚动范围
    self.contentSize = CGSizeMake(self.bounds.size.width, totalH);
}

```

5. 如何判断cell显示在屏幕上

- 当tableView内容往下走



当tableView内容往下走.gif

- 当tableView内容往上走



```
// 根据cell尺寸判断cell在不在屏幕上
- (BOOL)isInScreen:(CGRect)cellF
{
    // tableView能滚动，因此需要加上偏移量判断

    // 当tableView内容往下走，offsetY会一直增加，cell的最大y值 < offsetY偏移量，cell移除屏幕
    // tableView内容往上走，offsetY会一直减少，屏幕的最大Y值 < cell的y值，Cell移除屏幕
    // 屏幕最大y值 = 屏幕的高度 + offsetY

    // 这里拿屏幕来比较，其实是因为tableView的尺寸我默认等于屏幕的高度，正常应该是tableView的高度。
    // cell在屏幕上，cell的最大y值 > offsetY && cell的y值 < 屏幕的最大Y值(屏幕的高度 + offsetY)

    CGFloat offsetY = self.contentOffset.y;

    return CGRectGetMaxY(cellF) > offsetY && cellF.origin.y < self.bounds.size.height + offsetY;
}
```

6. 在滚动的时候，如果有新的cell出现在屏幕上，先从缓存池中取，没有取到，在创建新的cell.

分析：

- 需要及时监听tableView的滚动，判断下有没有新的cell出现。
- 大家都会想到scrollViewDidScroll方法，这个方法只要一滚动scrollView就会调用，但是这个方法有个弊端，就是tableView内部需要作为自身的代理，才能监听，这样不好，有时候外界也需要监听滚动，因此自身类最好不要成为自己的代理。（设计思想）

解决：

- 重写layoutSubviews，判断当前哪些cell显示在屏幕上。
- 因为只要一滚动，就会修改contentOffset,就会调用layoutSubviews，其实修改contentOffset，内部其实是修改tableView的bounds,而 layoutSubviews 刚好是 父控件尺寸一改 就会 调用 .具体需要了解scrollView底层实现 (<http://blog.jobbole.com/70143/>)。

思路：

- 判断下，当前tableView内容往上移动，还是往下移动，如何判断，取出显示在屏幕上的第一次cell，当前偏移量 > 第一个cell的y值，往下走。
- 需要搞个数组记录下，当前有多少cell显示在屏幕上，在一开始的时候记录。

```
@interface YZTableView ()

@property (nonatomic, strong) NSMutableArray *visibleCells;

@end

@implementation YZTableView

@dynamic delegate;

- (NSMutableArray *)visibleCells
{
    if (_visibleCells == nil) {
        _visibleCells = [NSMutableArray array];
    }
    return _visibleCells;
}

@end
```

- 往下移动
 - 如果已经滚动到tableView内容最底部，就不需要判断新的cell，直接返回.
 - 需要判断之前显示在屏幕cell有没有移除屏幕
 - 只需要判断下当前可见cell数组中第一个cell有没有离开屏幕
 - 只需要判断下当前可见cell数组中最后一个cell的下一个cell显没显示在屏幕上即可。

```
// 判断有没有滚动到最底部
if (offsetY + self.bounds.size.height > self.contentSize.height) {
    return;
}

// 判断下当前可见cell数组中第一个cell有没有离开屏幕
if ([self isInScreen:firstCell.frame] == NO) { // 如果不在屏幕
    // 从可见cell数组移除
    [self.visibleCells removeObject:firstCell];

    // 删除第0个从可见的indexPath
    [self.visibleIndexPaths removeObjectAtIndex:0];

    // 添加到缓存池中
    [self.reuserCells addObject:firstCell];

    // 移除父控件
    [firstCell removeFromSuperview];
}

// 判断下当前可见cell数组中最后一个cell的下一个cell显没显示在屏幕上
// 这里需要计算下一个cell的y值，需要获取对应的cell的高度
// 而高度需要根据indexPath，从数据源获取
// 可以数组记录每个可见cell的indexPath的顺序,然后获取对应可见的indexPath的角标，就能获取下一个indexPath。

// 获取最后一个cell的indexPath
NSIndexPath *indexPath = [self.visibleIndexPaths lastObject];

// 获取下一个cell的indexPath
NSIndexPath *nextIndexPath = [NSIndexPath indexPathForRow:indexPath.row + 1 inSection:0];

// 获取cell的高度
if ([self.delegate respondsToSelector:@selector(tableView:heightForRowAtIndexPath:)]) {
    cellH = [self.delegate tableView:self heightForRowAtIndexPath:nextIndexPath];
}else{
    cellH = 44;
}

// 计算下一个cell的y值
cellY = lastCellY + cellH;

// 计算下下一个cell的frame
CGRect nextCellFrame = CGRectMake(cellX, cellY, cellW, cellH);

if ([self isInScreen:nextCellFrame]) { // 如果在屏幕上，就加载

    // 通过数据源获取cell
    UITableViewCell *cell = [self.dataSource tableView:self cellForRowAtIndexPath:indexPath.nextIndexPath];

    cell.frame = nextCellFrame;

    [self insertSubview:cell atIndex:0];

    // 添加到cell可见数组中
    [self.visibleCells addObject:cell];

    // 添加到可见的indexPaths数组
    [self.visibleIndexPaths addObject:nextIndexPath];
}
```

- 往上移动

- 如果已经滚动到tableView最顶部，就不需要判断了有没有心的cell，直接返回.
- 需要判断之前显示在屏幕cell有没有移除屏幕
- 只需要判断下当前可见cell数组中最后一个cell有没有离开屏幕
- 只需要判断下可见cell数组中第一个cell的上一个cell显没显示在屏幕上即可
- 注意点：如果可见cell数组中第一个cell的上一个cell显示到屏幕上，一定要记得是插入到可见数组第0个的位置 。

```
        // 判断有没有滚动到最顶部
        if (offsetY < 0) {
            return;
        }

        // 判断下当前可见cell数组中最后一个cell有没有离开屏幕
        if ([self isInScreen:lastCell.frame] == NO) { // 如果不在屏幕
            // 从可见cell数组移除
            [self.visibleCells removeObject:lastCell];

            // 删除最后一个可见的indexPath
            [self.visibleIndexPaths removeLastObject];

            // 添加到缓存池中
            [self.reuserCells addObject:lastCell];

            // 移除父控件
            [lastCell removeFromSuperview];
        }

        // 判断下可见cell数组中第一个cell的上一个cell显没显示在屏幕上
        // 获取第一个cell的indexPath
        NSIndexPath *indexPath = self.visibleIndexPaths[0];

        // 获取下一个cell的indexPath
        NSIndexPath *preIndexPath = [NSIndexPath indexPathForRow:indexPath.row -
1 inSection:0];

        // 获取cell的高度
        if ([self.delegate respondsToSelector:@selector(tableView:heightForRowAtIndexPath:)] ) {
            cellH = [self.delegate tableView:self heightForRowAtIndexPath:preIndexPath];
        }else{
            cellH = 44;
        }

        // 计算上一个cell的y值
        cellY = firstCellY - cellH;

        // 计算上一个cell的frame
        CGRect preCellFrame = CGRectMake(cellX, cellY, cellW, cellH);

        if ([self isInScreen:preCellFrame]) { // 如果在屏幕上，就加载

            // 通过数据源获取cell
            UITableViewCell *cell = [self.dataSource tableView:self cellForRowAtIndexPath:preIndexPath];

            cell.frame = preCellFrame;

            [self insertSubview:cell atIndex:0];

            // 添加到cell可见数组中,这里应该用插入，因为这是最上面一个cell,应该插入到数组第0个

            [self.visibleCells insertObject:cell atIndex:0];

            // 添加到可见的indexPaths数组,这里应该用插入，因为这是最上面一个cell,应该插入到数组第0个

            [self.visibleIndexPaths insertObject:preIndexPath atIndex:0];
        }

    }
```

问题1:

- 判断下当前可见cell数组中最后一个cell的下一个cell显没显示在屏幕上
- 这里需要计算下一个cell的frame,frame就需要计算下一个cell的y值，需要获取对应的cell的高度 $cellY = lastCellY + cellH$
- 而高度需要根据indexPath，从数据源获取

解决:

- 可以搞个字典记录每个可见cell的indexPath,然后获取对应可见的indexPath，就能获取下一个indexPath.

```
@interface YZTableView ()

// 屏幕可见数组
@property (nonatomic, strong) NSMutableArray *visibleCells;

// 缓存池
@property (nonatomic, strong) NSMutableSet *reuserCells;

// 记录每个可见cell的indexPaths的顺序
@property (nonatomic, strong) NSMutableDictionary *visibleIndexPaths;

@end

- (NSMutableDictionary *)visibleIndexPaths
{
    if (_visibleIndexPaths == nil) {
        _visibleIndexPaths = [NSMutableDictionary dictionary];
    }

    return _visibleIndexPaths;
}
```

注意:

- 当cell从缓存池中移除，一定要记得从可见数组cell中移除，还有可见cell的indexPath也要移除.

```
        // 判断下当前可见cell数组中第一个cell有没有离开屏幕
        if ([self isInScreen:firstCell.frame] == NO) { // 如果不在屏幕
            // 从可见cell数组移除
            [self.visibleCells removeObject:firstCell];

            // 删除第0个从可见的indexPath
            [self.visibleIndexPaths removeObjectAtIndex:0];

            // 添加到缓存池中
            [self.reuserCells addObject:firstCell];
        }

        // 判断下当前可见cell数组中最后一个cell有没有离开屏幕
        if ([self isInScreen:lastCell.frame] == NO) { // 如果不在屏幕
            // 从可见cell数组移除
            [self.visibleCells removeObject:lastCell];

            // 删除最后一个可见的indexPath
            [self.visibleIndexPaths removeLastObject];

            // 添加到缓存池中
            [self.reuserCells addObject:lastCell];
        }
    }
}
```

7. 缓存池搭建，缓存池其实就是一个NSSet集合。

- 搞一个NSSet集合充当缓存池.
- cell离开屏幕，放进缓存池

- 提供从缓存池获取方法，从缓存池中获取cell,记住要从NSSet集合移除cell.

```
@interface YZTableView ()

// 屏幕可见数组
@property (nonatomic, strong) NSMutableArray *visibleCells;

// 缓存池
@property (nonatomic, strong) NSMutableSet *reuserCells;

// 记录每个cell的y值都对应一个indexPath
@property (nonatomic, strong) NSMutableDictionary *indexPathDict;

@end

@implementation YZTableView
- (NSMutableSet *)reuserCells
{
    if (_reuserCells == nil) {

        _reuserCells = [NSMutableSet set];

    }
    return _reuserCells;
}

// 从缓存池中获取cell
- (id)dequeueReusableCellWithIdentifier:(NSString *)identifier
{
    UITableViewCell *cell = [self.reuserCells anyObject];

    // 能取出cell,并且cell的标示符正确
    if (cell && [cell.reuseIdentifier isEqualToString:identifier]) {
        // 从缓存池中获取
        [self.reuserCells removeObject:cell];

        return cell;
    }

    return nil;
}

@end
```

8. tableView细节处理

原因：

刷新方法经常要调用

解决：

每次刷新的时候，先把之前记录的全部清空

```
// 刷新tableView
- (void)reloadData
{

    // 刷新方法经常要调用
    // 每次刷新的时候，先把之前记录的全部清空
    // 清空indexPath字典
    [self.indexPathDict removeAllObjects];
    // 清空屏幕可见数组
    [self.visibleCells removeAllObjects];
    ...
}
```

联系方式

如果你喜欢这篇文章，可以继续关注我，微博:吖了个峥
(<http://weibo.com/2034818060/profile?rightmod=1&wvr=6&mod=personinfo>),欢迎交流。
点击这下载源代码 (<https://github.com/iThinkerYZ/YZTableView>)。

(PS:另外咱们公司小码哥，诚邀IT届有事业心，有能力，有拼劲，有干劲各路英豪加盟一起创业，详情可以点击小码哥 (http://www.520it.com),小码哥官方微博 (http://weibo.com/u/5596623481?topnav=1&wvr=6&topsug=1)，或者微博私聊我)

如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

赞赏支持

♡ 喜欢

|

105

💬

🐼

🖼️

更多分享

(http://cwb.assets.jianshu.io/notes/images/1738040/)



写下你的评论...

39条评论

只看作者

按喜欢排序 按时间正序 按时间倒序



庸者的救赎 (/u/0726f4d689a3)

13楼 · 2015.08.23 18:17

(/u/0726f4d689a3)

以前杰哥的学生？现在小码哥当老师？



2人赞



回复



肥朝 (/u/f7daa458b874)

15楼 · 2015.10.13 00:47

(/u/f7daa458b874)

碉堡



1人赞



回复



这冬天不会冷 (/u/3e858b85c36f)

2楼 · 2015.08.17 13:53

(/u/3e858b85c36f)

把9改成50 滑动太快崩了



赞



回复

袁崢Seemygo (/u/b09c3959ab3b): @这冬天不会冷 (/users/3e858b85c36f) 我试试哈 报错原因最好复制给我下

2015.08.17 13:57 回复

这冬天不会冷 (/u/3e858b85c36f): @啊崢 (/users/b09c3959ab3b) 错误原因: Terminating app due to uncaught exception 'NSRangeException', reason: '*** -[__NSArrayM objectAtIndex:]: index 0 beyond bounds for empty array'

2015.08.17 14:14 回复

袁崢Seemygo (/u/b09c3959ab3b): @这冬天不会冷 (/users/3e858b85c36f) 嗯嗯 等我下课回去调试下

2015.08.17 14:17 回复



添加新评论

| 还有3条评论, 展开查看



这冬天不会冷 (/u/3e858b85c36f)

3楼 · 2015.08.17 14:12

(/u/3e858b85c36f)

UITableViewCell *firstCell = self.visibleCells[0]; 打全局断点 这里崩了



赞



回复

袁峥Seemygo (/u/b09c3959ab3b): @这冬天不会冷 (/users/3e858b85c36f) 好的 谢谢

2015.08.17 14:12 回复

袁峥Seemygo (/u/b09c3959ab3b): @这冬天不会冷 (/users/3e858b85c36f) bug以解决 晚上回去上传

2015.08.17 18:35 回复



添加新评论



lan_He (/u/ddaa8dae50b0)

4楼 · 2015.08.17 18:21

(/u/ddaa8dae50b0)

享元模式, 经典设计模式中有很多都能在OC和一些framework里面找到实现.



赞



回复



楚雨荨 (/u/cb7dba925a05)

5楼 · 2015.08.17 19:04

(/u/cb7dba925a05)

峥哥 你的动态图片是怎么用什么做的



赞



回复

袁峥Seemygo (/u/b09c3959ab3b): @楚雨荨 (/users/cb7dba925a05) gifbrewery

2015.08.17 23:56 回复



添加新评论



叶舞清风 (/u/34322aa36a0a)

6楼 · 2015.08.17 22:55

(/u/34322aa36a0a)

有没有一些关于自定义cell的讲解啊，我对于自定义cell和布局不太懂，总是数据解析出来了，呈现出来的时候不美观



赞



回复



CoderChou (/u/d5099a89892d)

7楼 · 2015.08.18 14:36

(/u/d5099a89892d)

@啊峥 (/users/b09c3959ab3b), 峥哥，我在自定义cell的时候，在.m里面这么写的

```
+(instancetype)cellWithTableViewCell:(UITableView *)tableView
```

```
{
```

```
static NSString * ID=@"nickCell";
```

```
VPNickNameCell * cell=[tableView dequeueReusableCellWithIdentifier:ID];
```

```
if (cell==nil) {
```

```
cell=[[NSBundle mainBundle]loadNibNamed:@"VPNickNameCell" owner:nil
```

```
options:nil]lastObject];
```

```
cell.accessoryType=UITableViewCellAccessoryDisclosureIndicator;
```

```
}
```

```
return cell;
```

```
}
```

然后再cellforindex就直接用了，老大说没有在cell的xib里面Identifier写上“nickCell”，没有cell的复用，我这是仿着明杰哥的团购视频写的，难道有错？求解释



赞



回复

袁峥Seemygo (/u/b09c3959ab3b): @为理想而行 (/users/d5099a89892d) 需要给cell绑定标识的

2015.08.18 16:32 回复

CoderChou (/u/d5099a89892d): @啊崢 (/users/b09c3959ab3b) 是通过注册register这个方法绑定吗


2015.08.18 18:34 回复

撸码是一种情怀 (/u/29f6fcce0289): @为理想而行 (/users/d5099a89892d) xib中绑定，你通过xib加载的cell.

2015.08.18 21:18 回复

添加新评论

还有3条评论， 展开查看

 勤劳的小男生 (/u/58a20988101a)


8楼 · 2015.08.18 17:23

(/u/58a20988101a)

李明杰的学生，还是小码哥的老师？

赞

回复

 这冬天不会冷 (/u/3e858b85c36f)


9楼 · 2015.08.18 18:21

(/u/3e858b85c36f)

是李明杰的学生，还是小码哥的老师

赞

回复

 娘哩个脚 (/u/0aaf1df5b2a9)

10楼 · 2015.08.18 20:49

(/u/0aaf1df5b2a9)

源码在哪下载呢？

赞

回复

袁峥Seemygo (/u/b09c3959ab3b): @娘哩个脚 (/users/0aaf1df5b2a9) 点击最后源码

2015.08.18 21:34 回复

添加新评论

 永战 (/u/3b6f730ba9b9)

11楼 · 2015.08.19 12:49

(/u/3b6f730ba9b9)

二、自定义UIScrollView， 模仿UITableView循环利用-----scrollView写错了，原谅我放荡不羁的眼睛


赞

回复

袁峥Seemygo (/u/b09c3959ab3b): @永战 (/users/3b6f730ba9b9) 😊

2015.08.19 12:51 回复

添加新评论

 8866b97b0554 (/u/8866b97b0554)

12楼 · 2015.08.19 15:01

(/u/8866b97b0554)

reloadData里写cellY = i * cellH;如果heightForRowAtIndexPath返回固定值这样没错，如果返回不固定值就有问题了，建议修改成cellY = totalH

赞

回复



抹茶不加糖 (/u/5993d22b90e8)

14楼 · 2015.09.09 13:15

(/u/5993d22b90e8)

老师，cell被循环利用后，它自身存储的属性比如文字，图片，还存在么？



赞



回复

袁峥Seemygo (/u/b09c3959ab3b): @bigbigworld (/users/5993d22b90e8) 在

2015.09.09 13:30 回复

抹茶不加糖 (/u/5993d22b90e8): @啊崢 (/users/b09c3959ab3b) 嗯嗯 ^_^

2015.09.09 20:19 回复



添加新评论



SilenceZhou (/u/065d6ccf06dc)

16楼 · 2015.10.18 00:15

(/u/065d6ccf06dc)

想请教下 那个你点击的时候出现圆点是什么软件呈现的效果！



赞



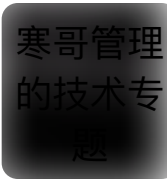
回复

1

2

下一页

被以下专题收入，发现更多相似内容



加载更多...