iOS高可控性日历基础组件-SKCalendarView的使用和实现思路的分享

2017-04-28 09:57 编辑： sasukeo 分类： iOS开发 来源： ShevaKuilind的简书 5 24512451

iOS 日历 SKCalendarView

招聘信息：手游服务器开发工程师（C++）

简述

SKCalendarView是一个高可控性的日历基础组件，为了提高应用的自由度，默认只提供了日历部分的视图封装，但不涵盖切换月份按钮、年月分显示等非关键性控件，但请不要担心，SKCalendarView为你提供了多样性的API，你可以很轻松的拿到这些信息去展示在你自己的自定义控件中，以及对当前UI的修改：如：替换日历主题图片、节假日或特殊日期的日历背景、各种现实颜色等等。不仅如此，SKCalendarView还为你封装了公历、农历、节假日以及中国24节气的核心算法，即使你觉得默认的视图并不合胃口，也可以直接快速的利用这套算法创造出一个全新的日历控件。最后，SKCalendarView还提供了一些简单的切换动画，如果你不喜欢它，可以忽略掉，用自己的，这里完全不会受到任何限制。如果觉得还不错，点个star吧~

效果图

热门资讯


多年iOS开发经验总结(一)
点击量 12238


33 款主宰 2017 iOS 开发的开源库
点击量 11851

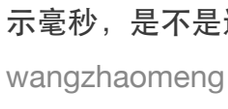

iOS:怎样创建一个好的App目录结构
点击量 6085

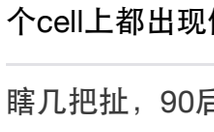
Stack Overflow 2017 开发者调查报告（程序
点击量 4481

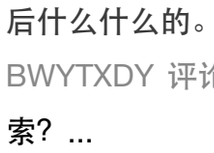
我们来谈谈iOS 11第二个开发者测试版值不值
点击量 3782

iOS面试之坎坷之路到放弃
点击量 3168


丁香园iOS电话面试问题总结
点击量 2964


对比 | Android与IOS职位需求，哪个更大?
点击量 2915


Swift 4 JSON 解析指南
点击量 2681



iOS模式详解runtime面试工作
点击量 2346

综合评论

郭明池脸都被打了多少回了，还敢出来瞎几把分析。
呜呜呜无呜呜 评论了 强势围观 郭明池对iPhone 8的十大预测...

差一点就被你坑了。。。要是倒计时显示毫秒，是不是还得每毫秒刷新一次
wangzhaomeng 评论了 iOS - 关于每个cell上都出现倒计场景的的研...

瞎几把扯，90后有这么无知？动不动90后什么什么的。90后比你知道的多得
BWYTXDY 评论了 你还在使用百度搜索？...

作者说的很良心。看一些评论，不知道有些人是不是傻，哈哈
孔思哲 评论了 你还在使用百度搜

最新更新：

- 发布了0.0.3版本
- 对日历算法部分进行优化，完全解决线程卡顿问题
- 修复了指定日期查寻在显示上的若干问题

一.如何使用

1.如何开始

1.从GitHub上Clone-->SKCalendarView, 然后查看Demo (由于使用cocoaPods管理，请打开xcworkspace工程进行查看)

2.在项目中使用SKCalendarView，直接将目录下的SKCalendarView文件夹拷贝到工程中，或在podfile文件中添加 pod 'SKCalendarView'

3.SKCalendarView的默认视图基于Masonry布局，如果需要使用, 请确保你的工程里已存在Masonry，下载地址

4.如果遇到其它问题，欢迎提交issues，我会及时回复

2.使用方法

- 头文件导入

```
1 #import "SKConstant.h"
```

索？...

装个xp虚拟机就行了，何必这么麻烦

yan127422 评论了 Mac下也能用抓包工具Fiddler...

mk

Esvn 评论了 多年iOS开发经验总结(一)...

那播放进度是怎么实现的,能否解答一下

tatatata 评论了 音频框架TheAmazingAudioEngine...

只能说作者对MVVM的理念理解不够，把网络请求，数据库操作业务处理等都

dl_genius 评论了 论MVVM伪框架结构和MVC中M的实现机制...

我去看了，里面内容很对，值得去看看

白开水6 评论了 iOS 伐码猿真爱—「偷懒 || 效率 工具类」...

mockplus做交互这么方便啊，赞、赞、赞！！！！

魂导射线 评论了 5分钟掌握8个常用交互组件...

相关帖子

求解友盟分享相关

iOS开发工程师提升与进阶

上下拉刷新

APP正在审核六天了，怎么办呢？？？

打五星宏辉怎样能赢钱|3728222|40安排小三哥哥结婚

五星宏辉有懂的人吗|3728222|40台茶叶蛋教授演讲

五星宏辉肉眼看单经验|3728222|40五星宏辉游戏机哪里有*@_@* 鬼脚七单挑游戏机

五星宏辉把命都输了|3728222|40五星宏辉草花机遥控器（⊙o⊙）真人单挑游戏机

关于UItableVIew Cell 嵌套 UItableView，第二层tableVIew 的cell点击事件

求解友盟分享相关

iOS开发工程师提升与进阶

上下拉刷新

APP正在审核六天了，怎么办呢？？？

打五星宏辉怎样能赢钱|3728222|40安排小三哥哥结婚

五星宏辉有懂的人吗|3728222|40台茶叶蛋教授演讲

五星宏辉肉眼看单经验|3728222|40五星宏辉游戏机哪里有*@_@* 鬼脚七单挑游戏机

五星宏辉把命都输了|3728222|40五星宏辉草花机遥控器（⊙o⊙）真人单挑

- 继承**SKCalendarView**

```
@property (nonatomic, strong) SKCalendarView * calendarView;
```

日历设置

```
_calendarView.calendarTodayTitleColor = [UIColor redColor];// 今天标题字体颜色
_calendarView.calendarTodayTitle = @"今日";// 今天下标题
_calendarView.dateColor = [UIColor orangeColor];// 今天日期数字背景颜色
_calendarView.calendarTodayColor = [UIColor whiteColor];// 今天日期字体颜色
_calendarView.dayoffInWeekColor = [UIColor redColor];
_calendarView.springColor = [UIColor colorWithRed:48 / 255.0 green:200 / 255.0 blue:104
_calendarView.summerColor = [UIColor colorWithRed:18 / 255.0 green:96 / 255.0 blue:0 al
_calendarView.autumnColor = [UIColor colorWithRed:232 / 255.0 green:195 / 255.0 blue:0
_calendarView.winterColor = [UIColor colorWithRed:77 / 255.0 green:161 / 255.0 blue:255
_calendarView.holidayColor = [UIColor redColor];//节日字体颜色
self.lastMonth = _calendarView.lastMonth;// 获取上个月的月份
self.nextMonth = _calendarView.nextMonth;// 获取下个月的月份
```

翻页动画

```
[SKCalendarAnimationManage animationWithView:self.calendarView andEffect:SK_ANIMATION_RE
```

获取农历年

```
self.chineseYearLabel.text = [NSString stringWithFormat:@"%@年", self.calendarView.chine
```

获取农历月日

```
self.chineseMonthAndDayLabel.text = [NSString stringWithFormat:@"%@%@", self.calendarVie
```

获取公历年/月

```
self.yearLabel.text = [NSString stringWithFormat:@"%@年%@月", @(self.calendarView.year),
```

获取节日/节气

```
self.holidayLabel.text = [self.calendarView getHolidayAndSolarTermsWithChineseDay:getNor
```

查询指定日期

```
[self.calendarView checkCalendarWithAppointDate:[NSDate date]];
```

日历**UI**配置

```
@property (nonatomic, strong) UIColor * weekBackgroundColor;// 周的背景颜色
@property (nonatomic, strong) UIColor * normalInWeekColor;// 周(除双休日外)字体颜色
@property (nonatomic, strong) UIColor * dayoffInWeekColor;// 双休日字体颜色
@property (nonatomic, strong) UIColor * calendarTodayColor;// 本日日期字体颜色
@property (nonatomic, strong) UIColor * dateColor;// 日期小背景颜色
@property (nonatomic, strong) UIImage * dateIcon;// 日期图片
@property (nonatomic, strong) UIColor * holidayBackgroundColor;// 节日背景颜色
@property (nonatomic, strong) UIColor * solarTeromBackgroundColor;// 节气背景颜色
@property (nonatomic, strong) UIColor * dateBackgroundColor;// 日期背景颜色(非节日&节气)
@property (nonatomic, strong) UIImage * dateBackgroundIcon;// 日期背景图片
@property (nonatomic, strong) NSString * calendarTodayTitle;// 本日日期标题
@property (nonatomic, strong) UIColor * calendarTodayTitleColor;// 本日日期标题字体颜色
@property (nonatomic, strong) UIColor * calendarTitleColor;// 日期标题字体颜色
@property (nonatomic, strong) UIColor * holidayColor;// 节日标题字体颜色
@property (nonatomic, strong) UIColor * springColor;// 春季节气颜色
@property (nonatomic, strong) UIColor * summerColor;// 夏季节气颜色
@property (nonatomic, strong) UIColor * autumnColor;// 秋季节气颜色
@property (nonatomic, strong) UIColor * winterColor;// 冬季节气颜色
@property (nonatomic, assign) BOOL enableClickEffect;// 开启点击效果
@property (nonatomic, assign) BOOL enableDateRoundCorner;// 开启日期圆角
```

获取点击到的日期

注意：这里需要先遵循代理协议

```
- (void)selectDateWithRow:(NSUInteger)row
```

二.如何实现

1.设计思路

游戏机

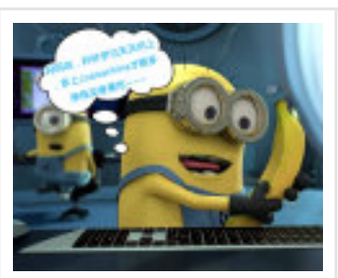
关于UItableVIew Cell 嵌套 UItableView，第二层tableVIew 的cell点击事件

微博

CocoaChina

+ 加关注

CocoaChina文章征集开始啦 CocoaChina现面向广大开发者征集技术文章（iOS开发教程、开发技巧、进阶文章、Swift、Xcode……等）。如果你是开发者或技术牛人，如果你觉得自己表达能力、文字功底都不错，没事就喜欢写写文章发发技术，而且天天看CocoaChina看CocoaChina看CocoaChina，那就快来投稿吧！



6月28日 16:42 转发 | 评论

CVP平台找BUG活动[坏笑]！！！http://t.cn/RopBNYO，筒子们赶紧来找B

- 总体上SKCalendarView仍然才去模块化思路，主要分为三个部分View(视图)、Animation(动画)以及Algorithm(算法)
- View主要负责处理外部对UI的配置信息、日历核心部分的展示、UI的刷新、效果的处理和界面控件的创建和布局约束等
- Animation主要负责日历翻页时的动画效果及点击日期的动画效果的处理
- Algorithm是整个SKCalendarView最核心的部分，负责了公历、农历、节假日以及中国24节气的核心算法，以及对日期查询的处理反馈

2.功能实现

2.1布局

思路

我们先要搞清楚日历是什么。所谓日历，就是一年当中12个月份的日期展示，每个月当中的日期数量由28~31天不等，这里指的是公历, 而农历当中每个月最多30天，虽然在计算方法上是有很大差别，但好在当代日历都是以公历为展示基准，所以只需要考虑公历的每月天数。

因为要考虑到展示上的美观性，一般都是采用正方形来展示，由于一周是固定的7天，所以我们日历的横向子控件数量也必须为7。但是这样问题就来了，由于需要考虑到与日期上方的周时间相对应，并且除了2月没有哪个月是的天数的7的倍数，也就做不到整除而导致无法形成正方形布局，所以我们不能直接用和月份天数相等的子控件数量来展示我们的日历，经过思考，我决定采取填充数据的方式来达到正方形展示的目的：

- 首先规划整体子控件数量，由于横向固定是7，那么纵向就由最多的一个月31天算，31 / 7 ≈ 4.4, 既然超过了4行，那么我们就放5行吧: 5 x 7 = 35，子控件放置35个如何？但经过尝试后，发现这并不可取：因为我们这里理想状态下的31天是以这个月的第一天恰好是周日 (周日为公历一周的开始) 为前提条件的，那么显然在现实生活里并不可能每个月都恰好第一天都是周日，所以，我们就需要考虑到需要显示的这个月的第一天是周几这个问题，众所周知，一周有7天的时间，那么每个月的第一天就有7种可能。做最多的打算，假设这个月总共有31天，而第一天恰好是周六，那么在这个月的1日这一天之前就有6天是没有日期的，结合我们之前计算的数量加上周六前的6天： 35 + 6 = 41，子控件放41个又如何呢？当然是不行了，因为需要正方形的日历，所以至少要成为7的倍数，最接近这个倍数的值就是我们要的答案：42.

实现

- 在基础控件的布局上，我们采取最简便的方式：周和日期我们分别使用了weekCollectionView、calendarCollectionView这两个UICollectionView来完成
- 而月份的背景数字monthBackgroundLabel作为最上面一层采用的是UILabel，在设置了其size和weight后，效果就如同背景图一样

```
// 周
  UICollectionViewFlowLayout * layout = [[UICollectionViewFlowLayout alloc] init];
  layout.scrollDirection = UICollectionViewScrollDirectionVertical;
  self.weekCollectionView = [[UICollectionView alloc] initWithFrame:self.frame collecti
  [self addSubview:self.weekCollectionView];
  self.weekCollectionView.backgroundColor = [UIColor whiteColor];
  self.weekCollectionView.delegate = self;
  self.weekCollectionView.dataSource = self;
  [self.weekCollectionView registerClass:[SKWeekCollectionViewCell class] forCellWithRe
  [self.weekCollectionView mas_makeConstraints:^(MASConstraintMaker *make) {
      make.top.equalTo(self);
      make.left.equalTo(self);
      make.right.equalTo(self);
      make.height.mas_offset(self.frame.size.height / 7.5);
      make.height.mas_greaterThanOrEqualTo(40).priorityHigh();
  }];
  // 日期
  UICollectionViewFlowLayout * dateLayout = [[UICollectionViewFlowLayout alloc] init];
  dateLayout.scrollDirection = UICollectionViewScrollDirectionVertical;
  self.calendarCollectionView = [[UICollectionView alloc] initWithFrame:self.frame coll
  [self addSubview:self.calendarCollectionView];
  self.calendarCollectionView.backgroundColor = [UIColor whiteColor];
  self.calendarCollectionView.delegate = self;
  self.calendarCollectionView.dataSource = self;
  [self.calendarCollectionView registerClass:[SKCalendarCollectionViewCell class] forCe
```

```
    [self.calendarCollectionView mas_makeConstraints:^(MASConstraintMaker *make) {
        make.top.equalTo(self.weekCollectionView.mas_bottom);
        make.left.equalTo(self);
        make.right.equalTo(self);
        make.bottom.equalTo(self);
    }];
    // 背景月份
    self.monthBackgroundLabel = [UILabel new];
    [self addSubview:self.monthBackgroundLabel];
    self.monthBackgroundLabel.textColor = [UIColor colorWithRed:0 green:0 blue:0 alpha:20
    self.monthBackgroundLabel.font = [UIFont systemFontOfSize:150.0f weight:120.f];
    self.monthBackgroundLabel.textAlignment = NSTextAlignmentCenter;
    [self.monthBackgroundLabel mas_makeConstraints:^(MASConstraintMaker *make) {
        make.edges.equalTo(self).with.insets(UIEdgeInsetsMake(0, 0, 0, 0));
    }];
```

对日历高度的控制

- 由于不同的月份的第一天所处的周时间不同，导致日历的有效日期 (有日期显示的) 行数不固定，如：当本月第一天为周日时，最多只占35个子控件位数，而我们一开始设置的子控件数量值是42，这样一来就会空出一行的空白出来，这是很不美观的。所以日历的高度对于我们来说就是一个把控的值，如何来保证可以根据每个月的天数来控制日历的高度呢，在SKCalendarView中采取了以下的办法:

```
if (self.calendarManage.isIncreaseHeight == YES) {// 根据isIncreaseHeight来判断是否需要更
          [self.calendarCollectionView mas_updateConstraints:^(MASConstraintMaker *make
              make.height.mas_offset(6 * (self.frame.size.height / 7.5));
          }];
          return 42;

      } else {
          if (self.calendarCollectionView.frame.size.height > 218) {
              [self.calendarCollectionView mas_updateConstraints:^(MASConstraintMaker *
                  make.height.mas_offset(5 * (self.frame.size.height / 7.5));
              }];
          }
          return 35;
      }
```

日期点击效果的处理

- 在SKCalendarCollectionViewCell的内部，我们将enableClickEffect(是否开启点击效果)为YES的状态设为开启效果，并调用动画管理类SKCalendarAnimationManage的方法

```
[SKCalendarAnimationManage clickEffectAnimationForView:self.baseView];
```

2.2 日历算法

这一部分算法是整个**SKCalendarView**最核心的部分

- SKCalendarManage以单例的模式封装了SKCalendarView全部的核心算法
- 主要难点在于对个别不定期节日，如复活节的日期的计算等，以及24节气和农历的计算，推荐阅读《算法:计算中国农历》
- 查看所选日期所处的月份:

```
#pragma mark - 查看所选日期所处的月份
- (void)checkThisMonthRecordFromToday:(NSDate *)today
{
    if (isEmpty(today)) {// 如果没有日期，默认今天
        today = [NSDate date];
    }
    [self calculationThisMonthDays:today];// 计算本月天数
    [self calculationThisMonthFirstDayInWeek:today];// 计算本月第一天是周几
}
```

- 计算本月天数

```
#pragma mark - 计算本月天数
- (void)calculationThisMonthDays:(NSDate *)days
{
    NSCalendar * calendar = [NSCalendar currentCalendar];
    if (isEmpty(days)) {
        days = [NSDate date];
    }
    NSRange range = [calendar rangeOfUnit:NSCalendarUnitDay inUnit:NSCalendarUnitMonth
    self.days = range.length;// 保存天数
}
```

- 计算本月第一天是周几

```objc
#pragma mark - 计算本月第一天是周几
- (void)calculationThisMonthFirstDayInWeek:(NSDate *)date;
{
    if (isEmpty(date)) {
        date = [NSDate date];
    }
    NSCalendar * calendar = [[NSCalendar alloc] initWithCalendarIdentifier:NSCalendarIc
    NSDateComponents * comps = [[NSDateComponents alloc] init];
    NSDateComponents * theComps = [[NSDateComponents alloc] init];
    NSInteger unitFlags = NSCalendarUnitDay | NSCalendarUnitWeekday | NSCalendarUnitMor
    comps = [calendar components:unitFlags fromDate:date];
    theComps = [calendar components:unitFlags fromDate:[NSDate date]];
    self.theMonth = [theComps month];// 本月的月份
    NSUInteger day = [comps day];// 是本月第几天
    self.todayInMonth = day;
    if (day > 1) {// 如果不是本月第一天
        // 将日期推算到本月第一天
        NSInteger hours = (day - 1) * -24;
        date = [NSDate dateWithTimeInterval:hours * 60 * 60 sinceDate:date];
    }
    comps = [calendar components:unitFlags fromDate:date];
    self.dayInWeek = [comps weekday];// 是周几
    self.year = [comps year];// 公历年
    self.month = [comps month];// 公里月
    [self creatcalendarArrayWithDate:date];// 创建日历数组
}
```

- 创建日历数组(公历、农历)

这里的算法还有优化的必要，如果有朋友可以指点一二，不胜感激

```objc
#pragma mark - 创建日历数组
- (void)creatcalendarArrayWithDate:(NSDate *)date
{
    self.calendarDate = [NSMutableArray new];
    self.chineseCalendarDate = [NSMutableArray new];
    self.chineseCalendarDay = [NSMutableArray new];
    for (NSInteger j = 0; j < 42; j ++) {// 创建空占位数组
        [self.calendarDate addObject:@""];
        [self.chineseCalendarDate addObject:@""];
        [self.chineseCalendarDay addObject:@""];
    }
    // 向前推算日期到本月第一天
    NSDate * firstDay = date;
    self.todayInMonth = self.todayInMonth + self.dayInWeek - 2;// 计算在本月日历上所处的
    switch (self.dayInWeek) {// 根据本月第一天是周几，来确定之后的日期替换空占位
        case 1:// 周日
            for (NSInteger i = 1; i <= self.days; i ++) {
                [self.calendarDate replaceObjectAtIndex:i - 1 withObject:@(i)];// 替换
                for (NSInteger j = 1; j <= self.days; j ++) {// 公历日期
                    // 向后推算至本月末
                    NSInteger hours = (j - 1) * 24;
                    NSDate * date = [NSDate dateWithTimeInterval:hours * 60 * 60 since
                    NSString * chineseDay = [self calculationChinaCalendarWithDate:dat
                    [self.chineseCalendarDate replaceObjectAtIndex:j - 1 withObject:ch
                    NSString * noHoliday = [self calculationChinaCalendarWithDate:date
                    [self.chineseCalendarDay replaceObjectAtIndex:j - 1 withObject:noH
                }
            }
            self.isIncreaseHeight = NO;
            break;
        case 2:// 周一
            for (NSInteger i = 1; i = 2) {
                    [self.calendarDate replaceObjectAtIndex:i - 1 withObject:@(i - 1)]
                    for (NSInteger j = 1; j = 2) {
                        // 向后推算至本月末
                        NSInteger hours = (j - 2) * 24;
                        NSDate * date = [NSDate dateWithTimeInterval:hours * 60 *
                        NSString * chineseDay = [self calculationChinaCalendarWith
                        [self.chineseCalendarDate replaceObjectAtIndex:j - 1 withC
                        NSString * noHoliday = [self calculationChinaCalendarWithD
                        [self.chineseCalendarDay replaceObjectAtIndex:j - 1 withOb
                    }
                }
            }
            self.isIncreaseHeight = NO;
            break;
        case 3:// 周二
            for (NSInteger i = 1; i = 3) {
                    [self.calendarDate replaceObjectAtIndex:i - 1 withObject:@(i - 2)]
                    for (NSInteger j = 1; j = 3) {
                        // 向后推算至本月末
                        NSInteger hours = (j - 3) * 24;
                        NSDate * date = [NSDate dateWithTimeInterval:hours * 60 *
                        NSString * chineseDay = [self calculationChinaCalendarWith
                        [self.chineseCalendarDate replaceObjectAtIndex:j - 1 withC
```

```
                        NSString * noHoliday = [self calculationChinaCalendarWithD
                        [self.chineseCalendarDay replaceObjectAtIndex:j - 1 withOb
                    }
                }
            }
        }
        self.isIncreaseHeight = NO;
        break;
    case 4:// 周三
        for (NSInteger i = 1; i = 4) {
                [self.calendarDate replaceObjectAtIndex:i - 1 withObject:@(i - 3)]
                for (NSInteger j = 1; j = 4) {
                        // 向后推算至本月末
                        NSInteger hours = (j - 4) * 24;
                        NSDate * date = [NSDate dateWithTimeInterval:hours * 60 *
                        NSString * chineseDay = [self calculationChinaCalendarWith
                        [self.chineseCalendarDate replaceObjectAtIndex:j - 1 withC
                        NSString * noHoliday = [self calculationChinaCalendarWithD
                        [self.chineseCalendarDay replaceObjectAtIndex:j - 1 withOb
                    }
                }
            }
        }
        self.isIncreaseHeight = NO;
        break;
    case 5:// 周四
        for (NSInteger i = 1; i = 5) {
                [self.calendarDate replaceObjectAtIndex:i - 1 withObject:@(i - 4)]
                for (NSInteger j = 1; j = 5) {
                        // 向后推算至本月末
                        NSInteger hours = (j - 5) * 24;
                        NSDate * date = [NSDate dateWithTimeInterval:hours * 60 *
                        NSString * chineseDay = [self calculationChinaCalendarWith
                        [self.chineseCalendarDate replaceObjectAtIndex:j - 1 withC
                        NSString * noHoliday = [self calculationChinaCalendarWithD
                        [self.chineseCalendarDay replaceObjectAtIndex:j - 1 withOb
                    }
                }
            }
        }
        self.isIncreaseHeight = NO;
        break;
    case 6:// 周五
        for (NSInteger i = 1; i = 6) {
                [self.calendarDate replaceObjectAtIndex:i - 1 withObject:@(i - 5)]
                for (NSInteger j = 1; j = 6) {
                        // 向后推算至本月末
                        NSInteger hours = (j - 6) * 24;
                        NSDate * date = [NSDate dateWithTimeInterval:hours * 60 *
                        NSString * chineseDay = [self calculationChinaCalendarWith
                        [self.chineseCalendarDate replaceObjectAtIndex:j - 1 withC
                        NSString * noHoliday = [self calculationChinaCalendarWithD
                        [self.chineseCalendarDay replaceObjectAtIndex:j - 1 withOb
                    }
                }
            }
        }
        if (self.days == 31) {// 是否为大月
            self.isIncreaseHeight = YES;
        } else {
            self.isIncreaseHeight = NO;
        }
        break;
    case 7:// 周六
        for (NSInteger i = 1; i = 7) {
                [self.calendarDate replaceObjectAtIndex:i - 1 withObject:@(i - 6)]
                for (NSInteger j = 1; j = 7) {
                        // 向后推算至本月末
                        NSInteger hours = (j - 7) * 24;
                        NSDate * date = [NSDate dateWithTimeInterval:hours * 60 *
                        NSString * chineseDay = [self calculationChinaCalendarWith
                        [self.chineseCalendarDate replaceObjectAtIndex:j - 1 withC
                        NSString * noHoliday = [self calculationChinaCalendarWithD
                        [self.chineseCalendarDay replaceObjectAtIndex:j - 1 withOb
                    }
                }
            }
        }
        self.isIncreaseHeight = YES;
        break;
    }
}
```

- 计算农历日期

由于农历、节假日都是在同一个位置展示，就放到了一个函数里

1.复活节采用了Meeus/Jones/Butcher算法

2.二十四节气采用了积日日计算公式F = 365.242 (y – 1900) + 6.2 + 15.22 x - 1.9 sin(0.262 x)

探讨：

这个函数当中24节气的算法在执行当中由于需要对积日进行计算，就需要处理1900-1-0这个基准日的日期转换，由于stringFromDate方法过于耗时，会导致一定的线程卡顿，目前我是将这24个节气根据月份分开来执行，然后使用单例NSDateFormatter来解决这个问题

```
- (NSDateFormatter *)dateFormatter
{
    if (!_dateFormatter) {
        _dateFormatter = [[NSDateFormatter alloc] init];
        [_dateFormatter setDateFormat:@"yyyy-MM-dd"];
    }
    return _dateFormatter;
}
- (NSDateFormatter *)strDateFormatter
{
    if (!_strDateFormatter) {
        _strDateFormatter = [[NSDateFormatter alloc] init];
        [_strDateFormatter setDateFormat:@"MM-dd"];
    }
    return _strDateFormatter;
}
- (NSDate *)baseDate
{
    if (!_baseDate) {
        _baseDate = [self.dateFormatter dateFromString:@"1900-1-1"];
    }
    return _baseDate;
}
```

```
#pragma mark - 计算农历日期
- (NSString *)calculationChinaCalendarWithDate:(NSDate *)date dispalyHoliday:(BOOL)dis
{
    if (isEmpty(date)) {
        return nil;
    }
    NSArray * chineseYears = @[@"甲子", @"乙丑", @"丙寅", @"丁卯", @"戊辰", @"己巳", @"庚
    NSArray * chineseMonths = @[@"正月", @"二月", @"三月", @"四月", @"五月", @"六月", @"
                                @"九月", @"十月", @"冬月", @"腊月"];
    NSArray * chineseDays = @[@"初一", @"初二", @"初三", @"初四", @"初五", @"初六", @"初
    NSCalendar * localeCalendar = [[NSCalendar alloc] initWithCalendarIdentifier:NSCal
    unsigned unitFlags = NSCalendarUnitYear | NSCalendarUnitMonth |  NSCalendarUnitDay
    NSDateComponents * localeComp = [localeCalendar components:unitFlags fromDate:date
    self.chineseYear = [chineseYears objectAtIndex:localeComp.year - 1];
    NSString * m_str = [chineseMonths objectAtIndex:localeComp.month - 1];
    self.chineseMonth = m_str;
    NSString * d_str = [chineseDays objectAtIndex:localeComp.day - 1];
    NSString * chineseCal_str = d_str;
    // 农历节日
    if([chineseMonths containsObject:m_str] && [d_str isEqualToString:@"初一"]) {
        chineseCal_str = m_str;
        if ([m_str isEqualToString:@"正月"] && [d_str isEqualToString:@"初一"]) {
            chineseCal_str = @"春节";
        } else{
            chineseCal_str = @"初一";
        }
    } else if ([m_str isEqualToString:@"正月"] && [d_str isEqualToString:@"十五"]) {
        chineseCal_str = @"元宵节";
    } else if ([m_str isEqualToString:@"五月"] && [d_str isEqualToString:@"初五"]) {
        chineseCal_str = @"端午节";
    } else if ([m_str isEqualToString:@"七月"] && [d_str isEqualToString:@"初七"]) {
        chineseCal_str = @"七夕";
    } else if ([m_str isEqualToString:@"七月"] && [d_str isEqualToString:@"十五"]) {
        chineseCal_str = @"中元节";
    } else if ([m_str isEqualToString:@"八月"] && [d_str isEqualToString:@"十五"]) {
        chineseCal_str = @"中秋节";
    } else if ([m_str isEqualToString:@"九月"] && [d_str isEqualToString:@"初九"]) {
        chineseCal_str = @"重阳节";
    } else if ([m_str isEqualToString:@"腊月"] && [d_str isEqualToString:@"初八"]) {
        chineseCal_str = @"腊八节";
    } else if ([m_str isEqualToString:@"腊月"] && [d_str isEqualToString:@"廿三"]) {
        chineseCal_str = @"小年";
    } else if ([m_str isEqualToString:@"腊月"] && [d_str isEqualToString:@"三十"]) {
        chineseCal_str = @"除夕";
    }
    // 公历节日
    NSDictionary * Holidays = @{@"01-01":@"元旦",
                                @"02-14":@"情人节",
                                @"03-08":@"妇女节",
                                @"03-12":@"植树节",
                                @"04-01":@"愚人节",
                                @"05-01":@"劳动节",
                                @"05-04":@"青年节",
                                @"06-01":@"儿童节",
                                @"07-01":@"建党节",
                                @"08-01":@"建军节",
```

```
                                    @"09-10":@"教师节",
                                    @"10-01":@"国庆节",
                                    @"12-24":@"平安夜",
                                    @"12-25":@"圣诞节"};
//    NSDateFormatter * dateFormatt= [[NSDateFormatter alloc] init];
//    [dateFormatt setDateFormat:@"MM-dd"];
    NSString * nowStr = [self.strDateFormatter stringFromDate:date];
    // 复活节, Meeus/Jones/Butcher算法
    NSUInteger a = self.year % 19;
    NSUInteger b = self.year / 100;
    NSUInteger c = self.year % 100;
    NSUInteger d = b / 4;
    NSUInteger e = b % 4;
    NSUInteger f = (b + 8) / 25;
    NSUInteger g = (b - f + 1) / 3;
    NSUInteger h = (19 * a + b - d - g + 15) % 30;
    NSUInteger i = c / 4;
    NSUInteger k = c % 4;
    NSUInteger l = (32 + (2 * e) + (2 * i) - h - k) % 7;
    NSUInteger m = (a + (11 * h) + (22 * l)) / 451;
    NSUInteger theMonth = (h + l - (7 * m) + 114) / 31;
    NSUInteger day = ((h + l - (7 * m) + 114) % 31)+ 1;
    NSString * easter = [NSString stringWithFormat:@"0%@-%@", @(theMonth), @(day)];
    if ([easter isEqualToString:nowStr]) {
        chineseCal_str = @"复活节";
    }
    NSArray * array = [Holidays allKeys];
    if([array containsObject:nowStr]) {
        chineseCal_str = [Holidays objectForKey:nowStr];
    }
    // 公历礼拜节日
    NSCalendar * calendar = [[NSCalendar alloc] initWithCalendarIdentifier:NSCalendarI
    NSDateComponents * comps = [[NSDateComponents alloc] init];
    NSInteger unit = NSCalendarUnitDay | NSCalendarUnitWeekday | NSCalendarUnitMonth |
    comps = [calendar components:unit fromDate:date];
    NSUInteger month = [comps month];
    NSUInteger dayInMonth = [comps day];
    switch (month) {
        case 5:
            if (dayInMonth == 14) {
                chineseCal_str = @"母亲节";
            }
            break;
        case 6:
            if (dayInMonth == 21) {
                chineseCal_str = @"父亲节";
            }
            break;
        case 11:
            if (dayInMonth == 26) {
                chineseCal_str = @"感恩节";
            }
            break;
        default:
            break;
    }
    // 二十四节气，将节气按月份拆开计算，否则由于计算积日所需日期转换stringFromDate方法过于耗
    NSString * solarTerms = @"";
    switch (self.month) {// 过滤月份
        case 1:
            for (NSInteger i = 0; i < 2; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 0:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"小寒";
                        }
                        break;
                    case 1:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"大寒";
                        }
                        break;
                }
            }
            break;
        case 2:
            for (NSInteger i = 2; i < 4; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 2:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"立春";
                        }
                        break;
                    case 3:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"雨水";
                        }
                        break;
                }
            }
            break;
```

```
        case 3:
            for (NSInteger i = 4; i < 6; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 4:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"惊蛰";
                        }
                        break;
                    case 5:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"春分";
                        }
                        break;
                }
            }
            break;
        case 4:
            for (NSInteger i = 6; i < 8; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 6:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"清明";
                        }
                        break;
                    case 7:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"谷雨";
                        }
                        break;
                }
            }
            break;
        case 5:
            for (NSInteger i = 8; i < 10; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 8:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"立夏";
                        }
                        break;
                    case 9:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"小满";
                        }
                        break;
                }
            }
            break;
        case 6:
            for (NSInteger i = 10; i < 12; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 10:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"芒种";
                        }
                        break;
                    case 11:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"夏至";
                        }
                }
            }
            break;
        case 7:
            for (NSInteger i = 12; i < 14; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 12:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"小暑";
                        }
                        break;
                    case 13:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"大暑";
                        }
                        break;                }
            }
            break;
        case 8:
            for (NSInteger i = 14; i < 16; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 14:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"立秋";
                        }
                        break;
                    case 15:
```

```
                    if ([solarTerms isEqualToString:nowStr]) {
                        chineseCal_str = @"处暑";
                    }
                    break;
                }
            }
            break;
        case 9:
            for (NSInteger i = 16; i < 18; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 16:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"白露";
                        }
                        break;
                    case 17:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"秋分";
                        }
                        break;
                }
            }
            break;
        case 10:
            for (NSInteger i = 18; i < 20; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 18:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"寒露";
                        }
                        break;
                    case 19:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"霜降";
                        }
                        break;
                }
            }
            break;
        case 11:
            for (NSInteger i = 20; i < 22; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 20:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"立冬";
                        }
                        break;
                    case 21:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"小雪";
                        }
                        break;
                }
            }
            break;
        case 12:
            for (NSInteger i = 22; i < 24; i ++) {
                solarTerms = [self calculationSolarTermsWithYear:self.year solarTermsI
                switch (i) {
                    case 22:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"大雪";
                        }
                        break;
                    case 23:
                        if ([solarTerms isEqualToString:nowStr]) {
                            chineseCal_str = @"冬至";
                        }
                        break;
                }
            }
            break;
    }
    if (display == YES) {// 需要显示假期&节日
        return chineseCal_str;
    }
    return d_str;
}
```

- 计算24节气的具体日期

这里的计算是整个线程里最耗时的地方，昨天用instruments查看这里的执行，竟然有8000x，我想最可能到这这个的原因就是dateFromString这里了，在我做了一些优化调整后，虽然已经不卡顿了，但不知道有什么更好的解决方案吗?

```
#pragma mark - 计算二十四节气的具体日期
```

```objc
/**
 * @param year 年份
 * @param index 节气索引，0代表小寒，1代表大寒，其它节气按照顺序类推
 */
- (NSString *)calculationSolarTermsWithYear:(NSUInteger)year solarTermsIndex:(NSUIntege
{
    NSString * solarTerms = @"";
    CGFloat base = 365.242 * (year - 1900) + 6.2 + (15.22 * index) - (1.9 * sinf(0.262
    NSInteger hours = (base - 1) * 24;// 由于基准日为1900年1月0日，所以这里需要-1
    NSDate * date = [NSDate dateWithTimeInterval:hours * 60 * 60 sinceDate:self.baseDat
    solarTerms = [self.strDateFormatter stringFromDate:date];
    return solarTerms;
}
```

2.3 动画

- 动画方面主要就是两个方面，翻页动画和点击效果
- 翻页动画

```objc
+ (void)animationWithView:(UIView *)view andEffect:(SK_ANIMATION)effect isNext:(BOOL)ne
{
    CATransition * transition = [CATransition animation];
    if (next == YES) {// 向下翻页
        switch (effect) {
            case SK_ANIMATION_REVEAL:
                transition.type = @"pageUnCurl";
                transition.subtype = kCATransitionFromLeft;
                break;
            case SK_ANIMATION_RIPPLE:
                transition.type = @"rippleEffect";
                transition.subtype = kCATransitionFromLeft;
                break;
            case SK_ANIMATION_SUCK:
                transition.type = @"suckEffect";
                transition.subtype = kCATransitionFromLeft;
                break;
        }
    } else {
        switch (effect) {
            case SK_ANIMATION_REVEAL:
                transition.type = @"pageCurl";
                transition.subtype = kCATransitionFromLeft;
                break;
            case SK_ANIMATION_RIPPLE:
                transition.type = @"rippleEffect";
                transition.subtype = kCATransitionFromRight;
                break;
            case SK_ANIMATION_SUCK:
                transition.type = @"suckEffect";
                transition.subtype = kCATransitionFromRight;
                break;
        }
    }
    transition.duration = 0.5;
    [view.layer addAnimation:transition forKey:nil];
}
```

- 点击效果

```objc
+ (void)clickEffectAnimationForView:(UIView *)view
{
    CABasicAnimation * scaleAnimation = [CABasicAnimation animationWithKeyPath:@"transfc
    scaleAnimation.fromValue = [NSNumber numberWithFloat:1.3];
    scaleAnimation.toValue = [NSNumber numberWithFloat:0.7];
    scaleAnimation.duration = 0.1;
    scaleAnimation.timingFunction = [CAMediaTimingFunction functionWithName:kCAMediaTimi
    [view.layer addAnimation:scaleAnimation forKey:nil];
}
```

好了，以上就是本次内容的分享，如果能帮到你，我很开心，欢迎在文章下面留言，在文中提到的关于算法上的优化，希望能够得到大神的指点

感谢你花时间阅读以上内容, 如果这个项目能够帮助到你，记得告诉我

Email: shevakuilin@gmail.com

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：

1313

上一篇：把握AFNet网络请求完成的正确时机

下一篇：改变iOS app的icon（iOS10.3）

相关资讯

- iOS 时区获取问题
- Spring MVC注解、标签库、国际化
- Swift 图片自动无线轮播用这个就够了
- iOS 性能监控方案（上篇）
- iOS开发，如何将照片保存到相册
- iOS APP 崩溃日志分析
- Runtime的运用和减少应用崩溃
- iOS 性能监控方案（下篇）
- 玩转swift -- UIKit 之 UIView（1）
- 教你开发省电的 iOS app（WWDC17 观后）

我来说两句

你怎么看？快来评论一下吧！

您还没有登录！请 登录 或 注册

发表评论

所有评论（5）

强_风云 2017-05-05 07:54:25

有个bug，点击写个月再点击上个月，显示今日的日期对不上。

0 0 回复

IvanChen 2017-05-06 06:00:44

回复：强_风云 有调整过来吗

0 0 回复

XXcc_学无止境 2017-05-02 15:14:02

需要小码哥扩展班视频的 加我QQ 245570636

0 0 回复

俊彩飞扬

2017-05-02 01:21:07

正好需要写个日历，32个赞

1 0 回复

pota500

2017-05-01 15:14:27

葡萄 交友+赖：pota500

0 0 回复

强_风云

2017-05-05 07:54:25

有个bug，点击写个月再点击上个月，显示今日的日期对不上。

0 0 回复

IvanChen

2017-05-06 06:00:44

回复：强_风云 有调整过来吗

0 0 回复

XXcc_学无止境

2017-05-02 15:14:02

需要小码哥扩展班视频的 加我QQ 245570636

0 0 回复

俊彩飞扬

2017-05-02 01:21:07

正好需要写个日历，32个赞

1 0 回复

pota500

2017-05-01 15:14:27

葡萄 交友+赖：pota500

0 0 回复

关于我们 商务合作 联系我们 合作伙伴

北京触控科技有限公司版权所有

©2016 Chukong Technologies,Inc.

京ICP备 11006519号 京ICP证 100954号 京公网安备11010502020289 京网文[2012]0426-138号