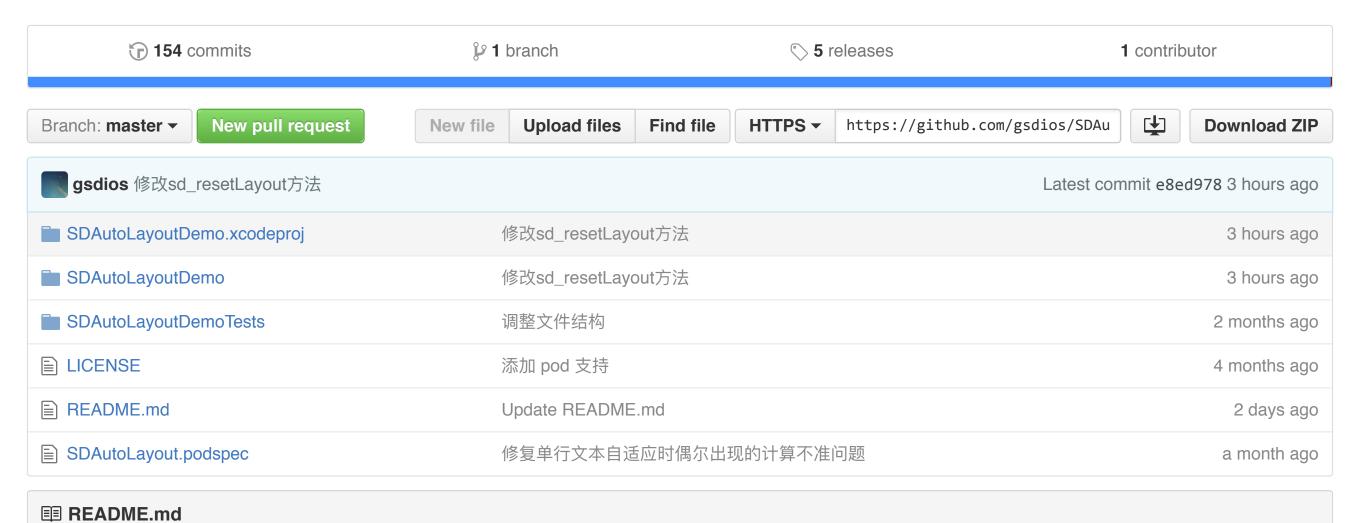


Pull requests Issues

Gist

应,致力于做最简单易用的AutoLayout库。The most easy way for autoLayout. Based Runtime.Auto Layout.



This repository Search

SDAutoLayout(一行代码搞定自动布局!)

一行代码搞定自动布局! 致力于做最简单易用的Autolayout库。The most easy way for autolayout.

支持pod: pod 'SDAutoLayout', '~> 1.31'

QQ交流群: 519489682(已满)497140713

☆☆ SDAutoLayout 基础版视频教程: http://www.letv.com/ptv/vplay/24038772.html ☆☆

☆☆ SDAutoLayout 进阶版视频教程: http://www.letv.com/ptv/vplay/24381390.html ☆☆

☆☆ SDAutoLayout 原理简介视频教程: http://www.iqiyi.com/w_19rt0tec4p.html ☆☆

部分SDAutoLayout的DEMO

完整微信Demo https://github.com/gsdios/GSD_WeiXin



更新记录:

2016.01.23 -- 增加label对attributedString的内容自适应

2016.01.21 -- 实现tableview局部刷新cell高度缓存的自动管理

2016.01.20 -- demo适配在ios7上的屏幕旋转问题

2016.01.18 -- 推出"普通简化版"tableview的cell自动高度方法(推荐使用),原来的需2步设置的普通版方法将标记过期

2016.01.13 -- 增加在不确定bottom view的情况下的cell高度自适应方法

2016.01.07 -- 1.增加 scrollview 横向内容自适应功能; 2.增加view宽高相等的功能

2016.01.03 -- 增加任何类型对象都可以实现一行代码搞定cell高度自适应;增加文档注释

2015.12.08 -- 重大升级: 1.支持scrollview内容自适应; 2.任意添加或者修改约束不冲突; 3.性能提升40%以上; 4.添加最大、

☆新增: cell高度自适应 + label文字自适应☆

>> 普通(简化)版【推荐使用】: tableview 高度自适应设置只需要2步

```
    >> 设置cell高度自适应:
        // cell布局设置好之后调用此方法就可以实现高度自适应(注意: 如果用高度自适应则不要再以cell的底边为参照去布局其子view)
        [cell setupAutoHeightWithBottomView:_view4 bottomMargin:10];
    >> 获取自动计算出的cell高度
        - (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPath *)indexPath
        {
            id model = self.modelsArray[indexPath.row];
            // 获取cell高度
            return [self.tableView cellHeightForIndexPath:indexPath model:model keyPath:@"model" cellClass:[DemoVC9Cell collabetath)
```

>> 升级版(适应于cell条数少于100的tableview): tableview 高度自适应设置只需要2步

```
    >> 设置cell高度自适应:
        // cell布局设置好之后调用此方法就可以实现高度自适应(注意: 如果用高度自适应则不要再以cell的底边为参照去布局其子view)
        [cell setupAutoHeightWithBottomView:_view4 bottomMargin:10];
    >> 获取自动计算出的cell高度
        - (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPath *)indexPath
        {
                  // 获取cell高度
                 return [self cellHeightForIndexPath:indexPath cellContentViewWidth:[UIScreen mainScreen].bounds.size.width];
        }
        }
        // Path cellContentViewWidth:[UIScreen mainScreen].bounds.size.width];
        // Path cellHeightForIndexPath:indexPath cellContentViewWidth:[UIScreen mainScreen].bounds.size.width];
        // Path cellHeightForIndexPath:indexPath:indexPath cellContentViewWidth:[UIScreen mainScreen].bounds.size.width];
        // Path cellHeightForIndexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:indexPath:i
```

****** 普通view的自动布局 *********

摒弃复杂累赘的约束,利用运行时Runtime在合适的时机布局视图。

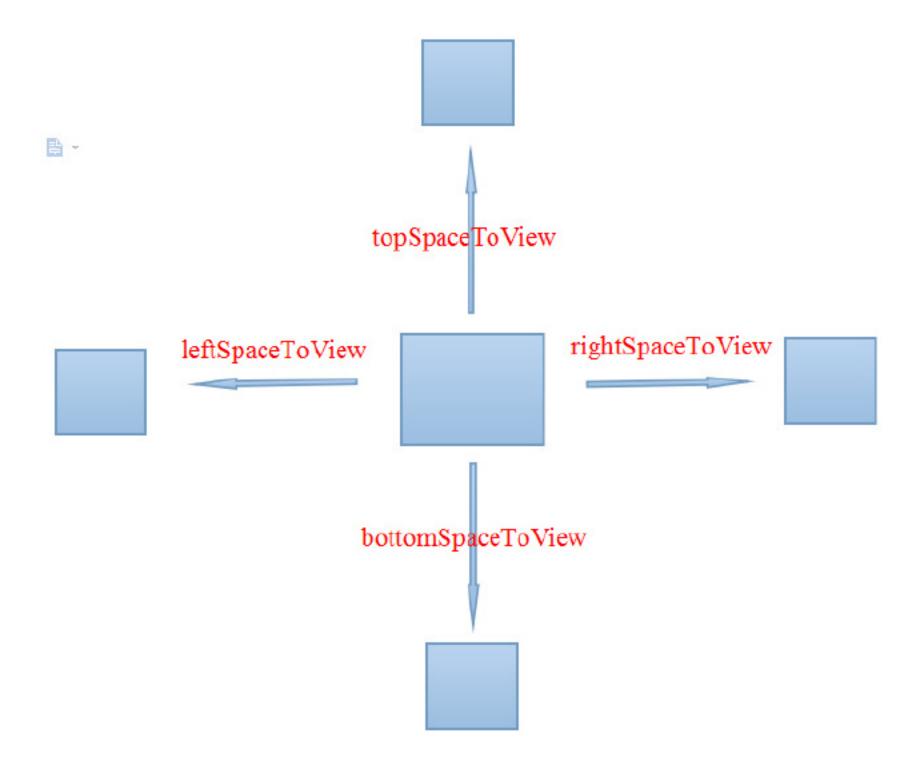
0.用法示例

```
/* 用法一 */
_view.sd_layout
.leftSpaceToView(self.view, 10)
.topSpaceToView(self.view, 80)
.heightIs(130)
.widthRatioToView(self.view, 0.4);

/* 用法二 (一行代码搞定, 其实用法一也是一行代码) */
_view.sd_layout.leftSpaceToView(self.view, 10).topSpaceToView(self.view,80).heightIs(130).widthRatioToView(self.view)
>> UILabel文字自适应:
// autoHeightRatio() 传0则根据文字自动计算高度(传大于0的值则根据此数值设置高度和宽度的比值)
```

```
_label.sd_layout.autoHeightRatio(0);
   注意:先把需要自动布局的view加入父view然后在进行自动布局,例:
   UIView *view0 = [UIView new];
   UIView *view1 = [UIView new];
   [self.view addSubview:view0];
   [self.view addSubview:view1];
   view0.sd_layout
    .leftSpaceToView(self.view, 10)
    .topSpaceToView(self.view, 80)
    .heightIs(100)
    .widthRatioToView(self.view, 0.4);
   view1.sd_layout
    .leftSpaceToView(view0, 10)
    .topEqualToView(view0)
    .heightRatioToView(view0, 1)
    .rightSpaceToView(self.view, 10);
```

1.用法简析



1.1 > leftSpaceToView(self.view, 10)

方法名中带有"SpaceToView"的方法表示到某个参照view的间距,需要传递2个参数:(UIView)参照view 和 (CGFloat)间 距数值

1.2 > widthRatioToView(self.view, 1)

方法名中带有"RatioToView"的方法表示view的宽度或者高度等属性相对于参照view的对应属性值的比例,需要传递2个参数: (UIView) 参照view 和 (CGFloat) 倍数

1.3 > topEqualToView(view)

方法名中带有"EqualToView"的方法表示view的某一属性等于参照view的对应的属性值,需要传递1个参数: (UIView)参照 view

1.4 > widthls(100)

方法名中带有"Is"的方法表示view的某一属性值等于参数数值,需要传递1个参数:(CGFloat)数值

PS

// 如果需要用"断言"调试程序请打开此宏(位于UIView+SDAutoLayout.h)

//#define SDDebugWithAssert

