



刘坤的技术博客

- [BlueBox](#)
- [所有文章](#)

嗨，我是刘坤，一名来自中国的 IOS 开发者，现就职于杭州阿里，花名‘念纪’，沉淀技术，寻求创意

[GitHub](#) [RSS](#)

友情链接

• [Casatwy Taloyum](#)
• [gf&zjの盗梦空间](#)
• [明弈](#)

Autolayout小结(二)

在[Autolayout小结\(一\)](#)中介绍了在Autolayout学习中一些基本的注意点，本文会针对一些布局上常见的问题进行分析。

1. 如何自动适应cell的高度
2. 如何在ScrollView中使用Autolayout
3. 使用Autolayout做动画
4. Autolayout在IOS6上的坑

1. 如何自动适应cell的高度

在IOS的布局中，计算和适应cell的高度是个经典的问题，在frame时代，我们都知道用sizeWithFont：先计算出文字的高度，然后通过计算得出cell的高度，然后赋予heightForRow：。

那在Autolayout时代如何计算cell的高度呢？因为sizeWithFont:方法已经不太实用了。其实Autolayout不但更简单，还可以不用写过多的计算代码达到自适应高度。

理论上是可以通过已知的完整的Constraints和View的属性来计算高度的，我们可以通过systemLayoutSizeFittingSize:方法来获取计算出来cell的size，我们知道cell的高度需要在tableView的代理方法tableView:heightForRowAtIndexPath:中实现的，那么我们考虑从以下两点来做：

- 通过创建一个额外的cell专门用来计算其高度
- 因为计算需要布局，所以尽量让其只计算一次，计算完可以将高度保存起来

基于这两个要点，我们可以尝试如下(伪代码)：

```
1  - (CGFloat)autoAdjustedCellHeightAtIndexPath:(NSIndexPath *)indexPath inTableView:(UITableView *)tableView {
2      CGFloat cellHeight = [self cellHeightAtIndexPath:indexPath];
3      if (cellHeight > 0) {
4          return cellHeight;
5      } else {
6          //此方法创建用于自适应的cell，注意创建一个就够了
7          UITableViewCell *cell = [self cellForTableViewAutoAdjust];
8
9          //更新cell,同cellForRow中更新一致
10         [self updateCell:cell]
11
12         //让cell进行layout
13         [cell setNeedsUpdateConstraints];
14         [cell updateConstraintsIfNeeded];
15         cell = CGRectMake(0.0f, 0.0f, CGRectGetWidth(tableView.bounds), CGRectGetHeight(layoutGuideView.bounds));
16         [cell setNeedsLayout];
17         [cell layoutIfNeeded];
18
19         CGFloat height = [cell.contentView systemLayoutSizeFittingSize:UILayoutFittingCompressedSize].height;
20
21         height += 1.0f;
22         //计算完后保存，避免多次重复计算
23         [self saveCellHeight:height forIndexPath:indexPath];
24         return height;
25     }
26 }
27 }
```

经过测试，工作良好，在体会到这点好处之后就发现Autolayout的好处了，它其实是让布局变得简单了。

在完成这个自动适应高度的设计过程中要注意几点：

1. cell中的多行UILabel，如果其width不是固定的话（比如屏幕尺寸不同，width不同），要手动设置其preferredMaxLayoutWidth。因为计算UILabel的intrinsicContentSize需要预先确定其width才行。
2. 要保证垂直方向上的Constraints理论上可以计算出cell的高度，另：最好调低其中一个Constraint的优先级，避免约束发生冲突

3. tableView的dequeueReusableCellWithIdentifier取出来的cell如果未用于tableView, 那么它就leak了. 因此用于计算高度的cell不要从此方法获取。

2. 如何在scrollView中使用Autolayout

scrollView比较特殊, 因为它有个contentSize的属性。那么在遇到scrollView时, 怎么使用Autolayout呢。其实关键点就一点:

ScrollView的contentSize的大小是由其subview的constraints来决定的。

知道这一点其实就够了, 那么基于这个特性, 在应用的时候要注意哪些呢?

- 完全依赖scrollView来计算subview的坐标的Constraints设法不行了, 计算条件互相依赖的, 常见的如left+right+top+bottom不行了。
- 要保证contentSize可以通过subview的constraints能够计算出来。

举个例子:

```
1 [childView mas_makeConstraints:^(MASConstraintMaker *make) {
2     make.width.mas_equalTo(scrollView.mas_width);
3     make.height.mas_equalTo(scrollView.mas_height);
4     make.leading.mas_equalTo(scrollView.mas_leading);
5     make.top.mas_equalTo(scrollView.mas_top);
6     make.bottom.mas_equalTo(scrollView.mas_bottom);
7     make.trailing.mas_equalTo(scrollView.mas_trailing).multipliedBy(1./4.);
8 }];
```

例子中如果scrollView的size是(100x100), 那么contentSize即为(400x100)

3. Autolayout时的动画

在使用Autolayout时, 动画的使用和以前也不同了, 以前我们是修改frame, 现在我们可以通过修改Constraints, 然后在动画时layoutIfNeeded就行了。

```
1 [UIView animateWithDuration:0.2 animations:^(
2     [view layoutIfNeeded];
3 )];
```

Autolayout有时在动画时候会很方便, 因为View之间的坐标是相互影响的, 在传统frame中, 如果改变一个view的frame, 那么可能你要更改很多view的frame, 才能让页面显得和谐。在Autolayout中可能只需要修改一个Constraint就可以了, 在做动画时会很方便。

4. Autolayout在IOS6上的坑

虽然Autolayout非常强大, 但是在刚出现的版本IOS6上, 还是有一些坑的。在IOS6上你会发现在某些系统控件上如: UITableViewCell, UITableView 等view上直接添加Constraints会crash。冲突的Constraints会直接导致crash等。

在stackoverflow有回答上说明了为什么UITableViewCell等部分系统控件上添加Constraints会crash, 即UITableViewCell等部分控件的layoutSubviews里面没有执行[super layoutSubviews]。

所以想要使用Autolayout的同学们, 如果你要兼容IOS6的话, 有些问题还是要重视的, 需要充分的测试哦。

Comments

Copyright © 2016 刘坤 Design credit: [Shashank Mehta](#)