

[首页](#) > [iOS开发](#)

AutoLayout 实现固定宽度动态高度的 ScrollView

2015-09-23 10:47

编辑: suiling

分类: iOS开发

来源: CocoaChina

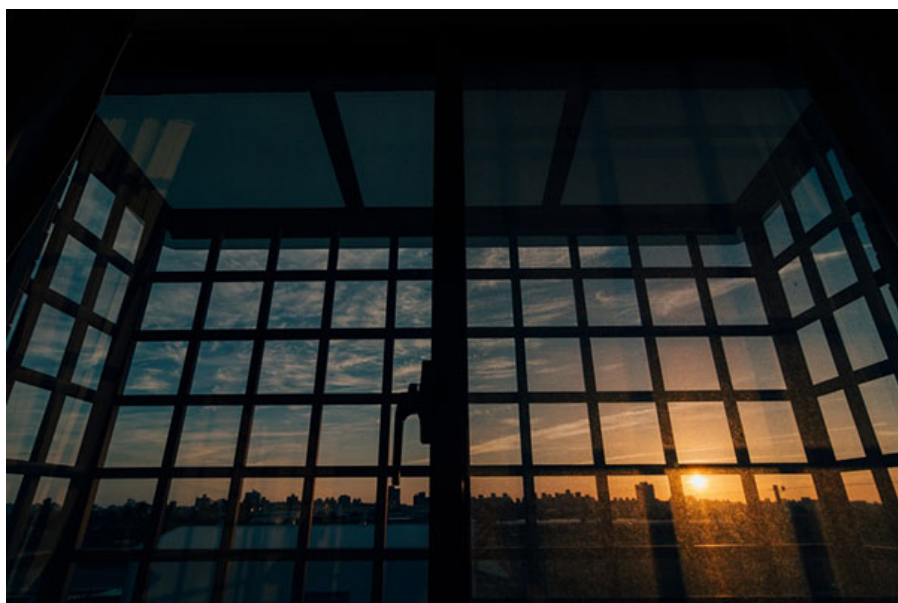
0

4348

scrollView

Autolayout

招聘信息: iOS开发工程师



- 本文由CocoaChina译者 SolarDreams 翻译
- 原文: [iOS Programming Recipe 36: A Fixed Width Dynamic Height ScrollView in AutoLayout](#)

作为一个iOS 开发者, 很多情况下会需要把一个和屏幕等宽的 contentview 添加到一个 scollview 内部中。大多数的 app 需要有响应式布局, 所以使用 Autolayout 可谓明智之选。第一次学习 scrollviews 时, 你或许会觉得碉堡了。但它们奇怪的规则也会令人有点沮丧。

准备

你对 Xcode 和 interface builder 都足够熟悉。

在 ScrollView 内的 Content View

这篇文章重点在 autolayout, 所以我要用各种方式来告诉你如何处理它。第一个方式是 interface builder, 因为这是最直观的。

设置垂直滚动视图的 interface builder

创建一个新的单视图控制器应用程序, 打开 storyboard, 添加一个 scrollView 到 storyboard 上的ViewController 上。使 scrollView 填满整个 viewcontroller 的 view。然后使用 storyborad 右下角的 pin 菜单, 给 scrollView 添加上、左、

热门资讯



现在, UICollectionViews有了

点击量 8023



iOS 9正式发布, 你发现其中隐藏的功能了吗

点击量 8010



打造安全的App! iOS 安全系列之 HTTPS 进

点击量 7173



【投稿】如何在运行时改变App的图标

点击量 6781



iOS 9系统策略更新, 使用微信SDK的开发者

点击量 6733



UICollectionView 高级进阶篇

点击量 6537



在 30 天内, 从零开始学习 Swift

点击量 6319



2015 苹果秋季新品发布会直播收看攻略

点击量 6216



全面了解 iOS 静态库开发

点击量 6203



iOS小技巧: 用runtime解决UIButton 重复点

点击量 5991

综合评论

垃圾中的垃圾

kenzo8503 评论了 奇葩双头Apple Watch出现 库克几乎是崩溃...

是“self.navigationController.interacti...而不

火的三次方 评论了 8行代码教你搞定导航控制器全屏滑动返回效果

66

hanxiao 评论了 【iOS】彩色TabBar切换动画实现

不错, 受教了

phoenixne 评论了 作为程序员最应该投资的是这十件事

分析的非常棒, 赞赞赞

右、下的约束。确保你没有选中“constrain to margin”复选框。图1 显示了正确设置的 pin 菜单。这将添加 scrollview 和它的父视图之间的约束。

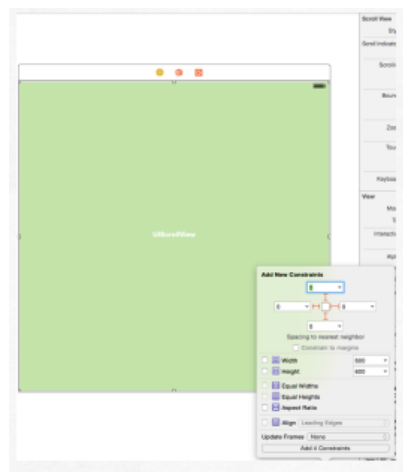


图1：给scrollview添加 上，左，右下的约束。

现在你想添加一个标准的 view 到 scrollview 上。就如你刚刚给 scrollview 添加上、左、右、下的约束那样，给这个新 view 也添加这些约束，并确保这些约束的值都是0。

当你完成后，通过选中 view 面板中的 view，查看在右侧的 Size Inspector，可以查看添加到 scrollview 和 Container view 的约束。图2 显示了内容视图，但 scroll view 应该看起来也是一样的。

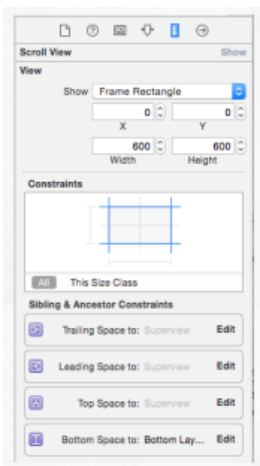


图2：Size inspector 显示在 scrollview 和 container view 上的约束。

ok，现在容器视图里已经有了一个 container view。有人会认为，因为我们给 scrollview 的边缘添加了上、左、下、右的约束，所以这个 container view 将会一直在 scrollview 的上面，并且拥有和窗口一样的宽度。但事实并非如此，因为 scrollviews 略有不同，这些约束定义了 scrollview 的 content size，但因为我们的 view 没有一个确切的 width 或 height，所以这个 content size 的 wide 和 tall 都是0。

通常我们都想要一个 scrollview 只能垂直滚动，所以并不想让 container view 比窗口宽，而高度我们通常是想要动态的，所以它会像其内部的 contents 一样高，稍后再说动态高度，现在将解决固定宽度的问题。

设置垂直滚动

我们要确保 container view 的宽度不会超过 window 的大小，为此我们将 container view 和 main view (包含

skyming 评论了 ReactiveCocoa2 源码浅析

受教了

xinyi 评论了 自动布局之 autoresizingMask使用详解...

方式“字阵”记录下来：少叠测时。这是什么方法 求教呢
Apple_MrPig 评论了 技术学习总原则

谁发的发的说法是第三代
dqw18037 评论了 集成新浪,腾讯,人人分享功能的Demo

新手表示收藏了
夜O无眠 评论了 Swift学习: 从 Objective-C到Swi...

绝世好文
章！！！！！！！！！！！！！！！！...
乔布斯2013 评论了 独立游戏人：像素风格游戏制作分享

相关帖子

谁有新浪微博点击中间加号后出来的6个图标的图片

在TabBarViewController中使用 autolayout

真机调试，界面就变得不清晰

ios开发中哪一部分比较难？

天气预报咋做呀？

UIImageView不能显示jpg图片

xcode 7 字体问题

茶漁飯後玩美情人→凱蒂貓頂級品茶會所+Line: kitty8102

茶漁飯後玩美情人→凱蒂貓頂級品茶會所+Line: kitty8102

UIScrollView 的视图)设置成等宽，在 view inspector 中，按着 ctrl 拖拽 main view 到container view 并从弹出的菜单中选中 EqualWidths。图3 显示了被连接的2个视图。

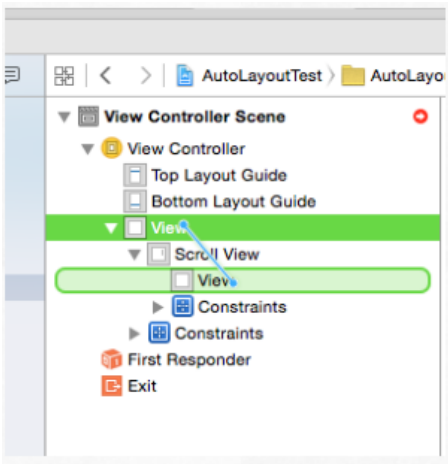


图3 按着 Ctrl 拖拽 main view 向 container view

设置动态 Container view 高

最后一步，没有那么多的步骤指导。基本上，为了这项工作，container view 上所有的子视图必须要有一个高度。一些子视图可以使用它们固有的高度，通常是由它们的宽来决定的。一般地，你需要为任何没有包含文本的视图指定宽度和高度。这些包含文本的视图至少有一个指定的宽或 margins。

第一个和最后一个视图应该分别以 container view 的顶部和底部分别固定。例如 在图4中，注意在图4中所有的视图都有左和右的约束，这将决定每个视图的宽度。labels 不具有高度的约束，因为它将取决于其内容的大小。方形的视图有确切的高度约束。还要注意所有的视图顶部和底部之间有一个显示的垂直约束。综合这些就能决定 containier view 的高，还有scrollview 的 content size。

在图4中显示如下约束。

- Top Label
 - 距 container view 顶约束:50pt
 - left: 15pt
 - right: 15pt
 - 底部约束(同为box顶约束): Standard
- Box label
 - 顶部约束(同为top label底约束): Standard
 - left:15pt
 - right: 15pt
 - 底部约束(同为bottom label顶约束): Standard
 - 高度:86pt;
- 底部label
 - 居上约束(同为box底约束): Standard
 - left:15pt
 - right:15pt
 - 距 container view 底约束: Standard

微博



CocoaChina

已关注

【开发者详解：端游及手游服务端的常用架构】手游页游和端游的服务端本质上没区别，区别的是游戏类型。卡牌跑酷类因为交互弱，玩家和玩家之间不需要实时面对面PK，打一下对方的离线数据，计算下排行榜，买卖下道具即可，所以实现往往使用简单的 HTTP 服务器：http://t.cn/RyX5sKL



今天 19:56
 转发(3) | 评论(1)

【超实用！如何准确的向工程师传达动效设计？】在实际工作中，相信大家常会有这样的疑问：怎样才能将自己精心设计的动态效果，准确的传达给工程师，避免实现的偏差呢？下面

100offer
 互联网人才猎头

一次申请让你在两天内收获
 10个优质工作机会

现在申请

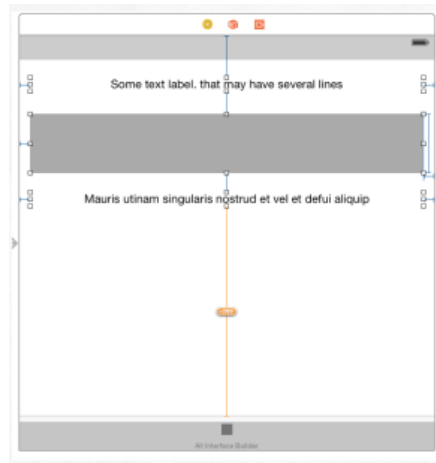


图4 添加子视图到容器视图示例

注意:如果要想 label 的 content 自动增长, 就要先选中label并在属性检查器重设置 label 的行数为"0"。这样 label 的行数就无限制了。

现在给底部的lable设置一段文字并运行这个例子, 将会触发垂直滚动的条件。图5 显示了堆叠视图。从图5中可以看出, 在容器视图顶部图4所示的单个视图, 容器视图堆放在 scrollview 上, 然后是 main view, 最后是 window。



图5 模拟堆叠视图

设置垂直滚动视图编程

ok, 这个部分将希望巩固最后一节概念, 创建一个single-view application的项目并打开ViewController.h文件。

注意:如果你熟练使用 Xcode 你也可以在原来的 storyboard 上拖进一个新的 ViewController, 创建一个类, 设置这个 ViewController的类为你新建的那个类。你可以在右侧的Identity Inspector里设置.然后把这些类目放入一标题栏里。

在ViewController.h里, 需要创建以下属性:

- “contentView”命名的UIView
- “scrollView”命名的UIScrollView
- 2个UILabel “topLabel”和“bottomLabel”
- “boxView”命名的UIView

当你完成你的代码后, 你的ViewController.h文件, 应该和代码1一样。

代码1 设置属性

```
@interface ViewControllerTwo : UIViewController

@property (strong, nonatomic) UIView *contentView;
@property (strong, nonatomic) UIScrollView *scrollView;

@property (strong, nonatomic) UILabel *topLabel;
@property (strong, nonatomic) UILabel *bottomLabel;

@property (strong, nonatomic) UIView *boxView;

@end
```

在 ViewController.m 里, 第一步需要在 viewDidLoad 中需要设置scrollView的宽度。如代码2中所示的添加 scrollView 和 contentView。

代码2: 在主视图中添加滚动视图和内容视图

```
- (void)viewDidLoad{

    [super viewDidLoad];

    self.scrollView = [[UIScrollView alloc] init];
    self.scrollView.translatesAutoresizingMaskIntoConstraints = NO;
    self.scrollView.backgroundColor = [UIColor blueColor];
    [self.view addSubview:self.scrollView];

    self.contentView = [[UIView alloc] init];
    self.contentView.backgroundColor = [UIColor redColor];
    self.contentView.translatesAutoresizingMaskIntoConstraints = NO;
    [self.scrollView addSubview:self.contentView];

    //....
}
```

再一次我们设置 scrollView 相对于 view 的 margins 为0， contentView 相对于 content view margins 为0。最后将 content view 的 width 与 main view 的 width 设为一致。这些都是以编程方式添加的约束。将代码3中的代码添加到 viewDidLoad 方法里代码2的后面。

代码3:添加contentView和scrollView的约束

```
NSDictionary *tmpViewsDictionary = @{@"scrollView":self.scrollView,
                                     @"contentView":self.contentView};

[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"V:|-(0)-[scrollView]-(0)
[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"H:|-(0)-[scrollView]-(0)

[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"V:|-(0)-[contentView]-(0)
[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"H:|-(0)-[contentView]-(0)

[self.view addConstraint:[NSLayoutConstraint constraintWithItem:self.contentView attribute:NSLayout
```

在这份代码中，我们使用了两种不同类型的约束。第一种使用的是 Visual format language，这是一种很好的添加多个约束的字符串描述。例如：

- |意思是superview
- .意思是view之间的空间(单个 - 意味着 standard)例如.- (0)- vs. -
- (某个值value)意思是宽
- [某个view]意思是一个view

把它们连接在一起作为第一个约束 @"V:|-(0)-[scrollView]-(0)-|"

这行代码的意思是给scrollView添加距 superview 左右都为0的 margin。

更多关于visual format language 请查看文档:[Visual Format Language](#)

宽度约束是将约束添加到视图的标准方式，你可以看得更清楚，但是有点繁琐。

最后，调用一个新的方法，把剩余的视图添加到content view中，我们把它叫做 addContentSubviews。

代码4 展示了这个复杂的 viewDidLoad 方法。

代码4 完整的 viewDidLoad 方法

```
- (void)viewDidLoad{

    [super viewDidLoad];

    self.scrollView = [[UIScrollView alloc] init];
    self.scrollView.translatesAutoresizingMaskIntoConstraints = NO;
    self.scrollView.backgroundColor = [UIColor blueColor];
    [self.view addSubview:self.scrollView];

    self.contentView = [UIView alloc] init];
    self.contentView.backgroundColor = [UIColor redColor];
```

```

self.contentView.translatesAutoresizingMaskIntoConstraints = NO;
[self.scrollView addSubview:self.contentView];

//Auto Layout Constraints for scrolling content view

NSDictionary *tmpViewsDictionary = @{@"scrollView":self.scrollView,
                                      @"contentView":self.contentView};

[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"V:|-(0)-[scrollView
[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"H:|-(0)-[scrollView

[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"V:|-(0)-[contentVie
[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"H:|-(0)-[contentVie

[self.view addConstraint:[NSLayoutConstraint constraintWithItem:self.contentView attribute:NSLa

[self addContentSubViews];
}

```

addContentSubViews方法十分简单，仅仅创建了几个label和一个box view.label的numberOfLines = 0，意味着lable的content会满足多行的需求。label 居中且自动换行。

代码5 展示了应该添加下面的viewDidLoad方法完整的方法。

代码5: addContentSubViews 实现

```

- (void)addContentSubViews{

    self.topLabel = [[UILabel alloc] init];
    self.topLabel.translatesAutoresizingMaskIntoConstraints = NO;
    self.topLabel.numberOfLines = 0;
    self.topLabel.textAlignment = NSTextAlignmentCenter;
    self.topLabel.lineBreakMode = NSLineBreakByWordWrapping;
    self.topLabel.text = @"Some text label. that may have several lines";
    self.topLabel.textColor = [UIColor blackColor];
    [self.contentView addSubview:self.topLabel];

    self.boxView = [[UIView alloc] init];
    self.boxView.translatesAutoresizingMaskIntoConstraints = NO;
    self.boxView.backgroundColor = [UIColor lightGrayColor];
    [self.contentView addSubview:self.boxView];

    self.bottomLabel = [[UILabel alloc] init];
    self.bottomLabel.numberOfLines = 0;
    self.bottomLabel.textAlignment = NSTextAlignmentCenter;
    self.bottomLabel.lineBreakMode = NSLineBreakByWordWrapping;

```

```

        self.bottomLabel.translatesAutoresizingMaskIntoConstraints = NO;

        self.bottomLabel.text = [self bottomLabelText];

        self.bottomLabel.textColor = [UIColor blackColor];

        [self.contentView addSubview:self.bottomLabel];

        [self addContentSubviewConstraints];

    }

```

在代码5 中 我们实现了两个新方法。一个简单地返回一个大字符串。代码6展示 bottomLabelText 方法的实现。

代码6: bottomLabelText 实现

```

- (NSString *)bottomLabelText{

    return @"Put in a massive string of your own here to see the scrolling in action";

}

```

最后一种方法将添加所有的约束来定义内容视图的内容高度。代码7是 addContentSubviewConstraints 的最终实现，它添加了和IB中所展示的完全相同的约束。

这是相当多的。代码7展示了完整的viewController.m文件。

代码7: viewcontoller.m完整视图

```

#import "ViewController.h"

@implementation ViewController

- (void)viewDidLoad{

    [super viewDidLoad];

    self.scrollView = [[UIScrollView alloc] init];
    self.scrollView.translatesAutoresizingMaskIntoConstraints = NO;
    self.scrollView.backgroundColor = [UIColor blueColor];
    [self.view addSubview:self.scrollView];

    self.contentView = [[UIView alloc] init];
    self.contentView.backgroundColor = [UIColor redColor];
    self.contentView.translatesAutoresizingMaskIntoConstraints = NO;
    [self.scrollView addSubview:self.contentView];

    //Auto Layout Constraints for scrolling content view

    NSDictionary *tmpViewsDictionary = @{@"scrollView":self.scrollView,
                                           @"contentView":self.contentView};

```



```

[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"V:|-()-[scrollView
[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"H:|-()-[scrollView

[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"V:|-()-[contentVie
[self.view addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"H:|-()-[contentVie

[self.view addConstraint:[NSLayoutConstraint constraintWithItem:self.contentView attribute:NSLa

[self addContentSubviews];
}

- (void)addContentSubviews{

    self.topLabel = [[UILabel alloc] init];
    self.topLabel.translatesAutoresizingMaskIntoConstraints = NO;
    self.topLabel.numberOfLines = 0;
    self.topLabel.textAlignment = NSTextAlignmentCenter;
    self.topLabel.lineBreakMode = NSLineBreakByWordWrapping;
    self.topLabel.text = @"Some text label. that may have several lines";
    self.topLabel.textColor = [UIColor blackColor];
    [self.contentView addSubview:self.topLabel];

    self.boxView = [[UIView alloc] init];
    self.boxView.translatesAutoresizingMaskIntoConstraints = NO;
    self.boxView.backgroundColor = [UIColor lightGrayColor];
    [self.contentView addSubview:self.boxView];

    self.bottomLabel = [[UILabel alloc] init];
    self.bottomLabel.numberOfLines = 0;
    self.bottomLabel.textAlignment = NSTextAlignmentCenter;
    self.bottomLabel.lineBreakMode = NSLineBreakByWordWrapping;
    self.bottomLabel.translatesAutoresizingMaskIntoConstraints = NO;
    self.bottomLabel.text = [self bottomLabelText];
    self.bottomLabel.textColor = [UIColor blackColor];
    [self.contentView addSubview:self.bottomLabel];

    [self addContentSubviewConstraints];
}

- (NSString *)bottomLabelText{

    return @"Mauris utinam singularis nostrud et vel et defui aliquip dui. Regula suscipere vel ra
}

```

```

- (void)addContentSubviewConstraints{

    NSDictionary *tmpViewsDictionary = @{@"topLabel":self.topLabel,
                                          @"boxView":self.boxView,
                                          @"bottomLabel":self.bottomLabel};

    [self.contentView addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"V:|-(60)-[to

    [self.contentView addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"H:|-[topLabe
    [self.contentView addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"H:|-[bottomL
    [self.contentView addConstraints:[NSLayoutConstraint constraintsWithVisualFormat:@"H:|-[boxView

    [self.contentView addConstraint:[NSLayoutConstraint constraintWithItem:self.contentView attribu

}

@end

```

噢，别忘了 swift，代码8 是完整的 swift 版实现。

代码8：完整的swift 控制器

GeSHi Error: GeSHi could not find the language swift (using path /home1/jhoffman/nscookbook/wp-cont

注意：使用 swift 时我不得不为 content view 定义一个高，这如没有任何意义一般，因为几乎是逐字的将代码转换为 swift。目前看来像是一个 swift bug，有任何消息我会通知你，我猜在 swift 2 时会表现的更好。好了，这期就到这里了，希望对你有所帮助。



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：

相关资讯

源码推荐(9.22): 利用UIScrollView实现几个页面的切换,
【投稿】友好的 ContainerView & AutoLayout 资料推荐
 源码推荐(8.21): Pageview和Scrollview的结合, 使用最简
 源码推荐(8.12): 带header footer 滑动菜单的
 使用UIScrollView 结合 UImageview 实现图片循环滚动

iOS 小 Tip: 优化侧滑返回与 ScrollView 的兼容性
 源码推荐(8.27): 自定义宽高比的相册框, view隐藏时也
 源码推荐(8.13): 滑动文本标签, 一个类解决广告轮播
 解决点击状态栏时ScrollView自动滚动到初始位置失效办
 源码推荐(7.15): 重写UIFont方法 设置字体格式以及大



文章评论 (0)

对这篇文章有什么感想, 写一下吧.....

发表评论

友情链接

Testin | APICloud | OneAPM | 极光推送 | Cocos商店 | 人人都是产品经理 | Cocos Link | Cocos引擎中文官网 | 推酷网 | 游戏陀螺 |
 9秒社团 | SegmentFault | 手游那点事 | 苹果开发者中心 | cocos2d-x | 游戏邦 | Nooidea.com | 9RIA天地会 | 爱应用



关注微信 每日推荐



扫一扫 浏览移动版

关于我们 广告合作 联系我们

北京触控科技有限公司版权所有

©2015 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