

Custom Collection View Layouts

Hands-on Challenges

Custom Collection View Layouts Hands-on Challenges

Copyright © 2015 Razeware LLC.

All rights reserved. No part of this book or corresponding materials (such as text, images, or source code) may be reproduced or distributed by any means without prior written permission of the copyright owner.

This book and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

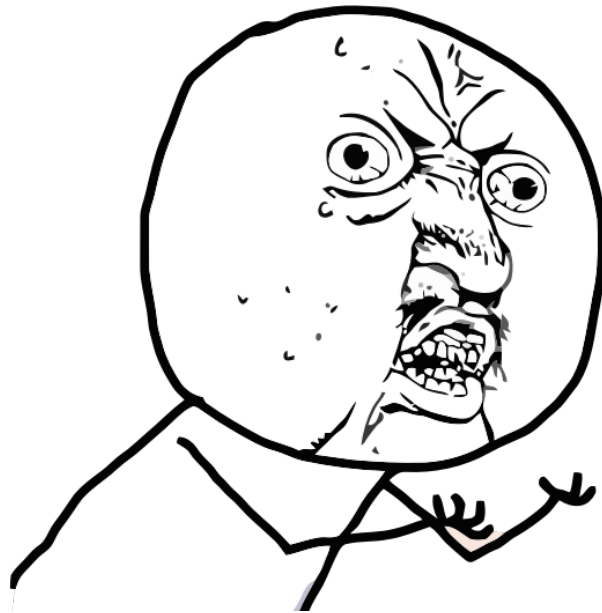
All trademarks and registered trademarks appearing in this book are the property of their respective owners.



Challenge C: Insets

Your stretchy header is looking pretty sweet at this point, but there is a flaw in its current implementation – it doesn't (yet!) take into account the `contentInset` property of the collection view.

This may not appear to be too much of an issue, but if you wanted to display some custom content above the header, such as a custom navigation bar, or wanted to move the header out from underneath the status bar, you wouldn't be able to because the layout ignores any values you set on this property.



y u no respect my content inset

So your challenge this time is to update the layout so it takes the collection view's `contentInset` property into account when calculating the frame of the header.

It's a short 'n sweet challenge, but it does a good job of highlighting the fact that you should always take any properties that can be set on the collection view which affect the position of the content into account in any custom layout you create.

Before you turn the page for our solution, be sure to give it a try for yourself first!



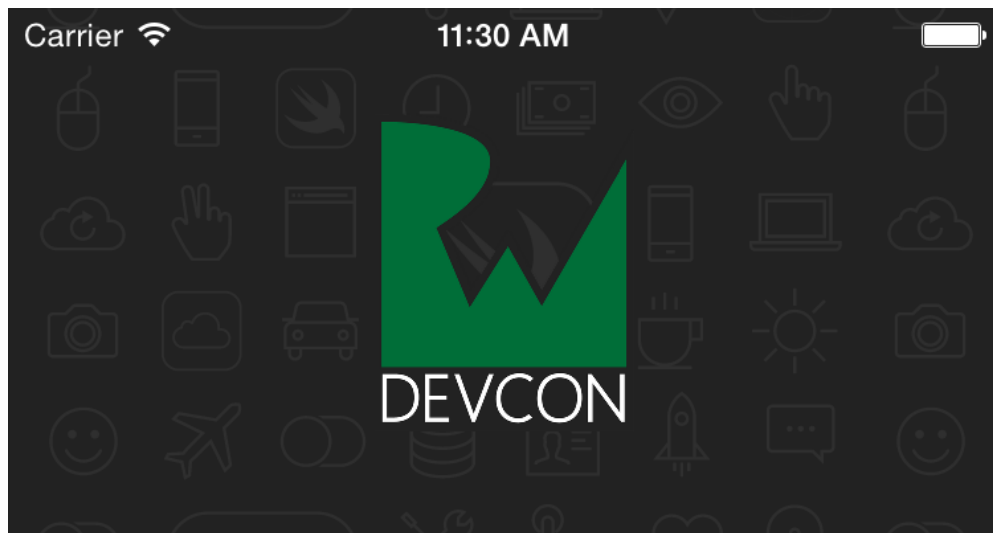
Solution

Open **ScheduleViewController.swift** and add the following line to the bottom of `viewDidLoad()`:

```
collectionView!.contentInset =  
    UIEdgeInsets(top: 20, left: 0, bottom: 0, right: 0)
```

Here you're setting the `top` value of the `contentInset` property to 20 points, which should move the header out from under the status bar.

Build and run. Unfortunately, you'll see nothing has changed.



You're going to fix that now.

Open **DIYLayout.swift**, find `layoutAttributesForElementsInRect(_:)` and add the following just below the call to `super`:

```
let insets = collectionView!.contentInset
```

Here you're getting a reference to the insets that are set on the collection view.

Next, add the following just above the `if` statement:

```
let minY = -insets.top
```

And now update the condition of the `if` statement to the following:

```
if (offset.y < minY) {
```

You first negate the `top` value of the insets, and then compare it with the `y` value of the offset, which is already negative. You do this because you don't want to start stretching the header until the `y` offset is less than the negated top inset.



Now update the line that calculates the `deltaY` value so it takes `minY` into account:

```
let deltaY = fabs(offset.y - minY)
```

You need to do this because the collection view's `contentOffset` property doesn't take it's `contentInset` property into account, and a `contentOffset` of 0,0 is always the upper-left corner of the collection view. So you need to subtract the inset from the offset to get the true delta.

Finally, update the line that calculates the height of the header's frame so it also takes `minY` into account:

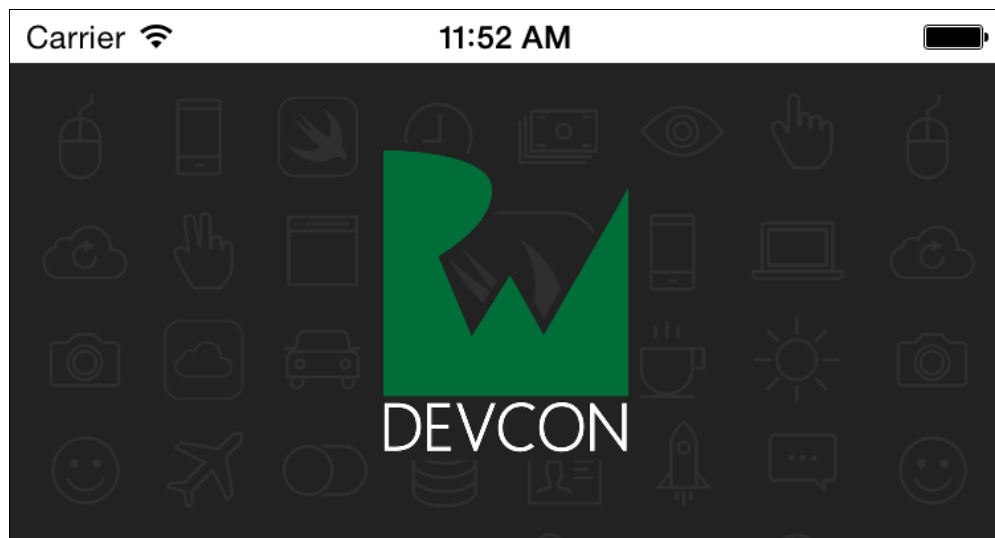
```
frame.size.height = max(minY, headerReferenceSize.height + deltaY)
```

Finally, reopen **ScheduleViewController.swift** and update the implementation of `preferredStatusBarStyle()` to:

```
return UIStatusBarStyle.Default
```

You do this so the status bar uses black text rather than white, so it can be seen on the white background below the collection view.

Build and run. You'll now see a 20 point gap between the top of the collection view and the top of the header, which is exactly what you wanted.



As this particular implementation doesn't require the gap, reset `preferredStatusBarStyle()` back to:

```
return UIStatusBarStyle.LightContent
```

And remove the line you added to `viewDidLoad()` that sets the collection view's `contentInset` property.



Congratulations! You have successfully updated the stretchy header layout so it takes the `contentInset` property of the collection view into account.

You may be thinking “*ok, but what about the left and right insets*” but don’t worry, your custom layout only manipulates the height and vertical origin of the header, so the rest are taken care of automatically by `UICollectionViewFlowLayout` :]

