



Stefan Moonen    [stefan.moonen@freshheads.com](mailto:stefan.moonen@freshheads.com)

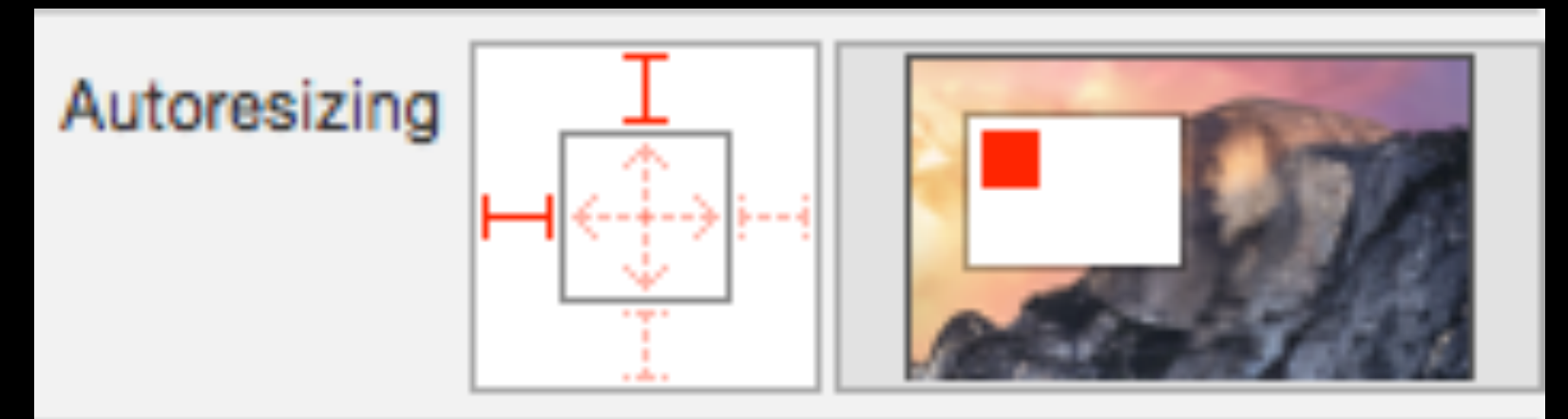
# iOS AutoLayout & Size Classes

Hoe en waarom zou je het gebruiken?

**Wat is AutoLayout?**

# Wat is AutoLayout?

- Bedoeld als vervanging voor "Springs & Struts"
- UIViews kunnen nu afhankelijk zijn van elkaar
- Werkt op basis van:
  - Constraints
  - Content hugging & compression resistance



# Constraints

- Letterlijk: de begrenzing van de layout-elementen
- Zonder constraints geen AutoLayout
- Constraints moeten compleet zijn
- Constraints mogen niet dubbelzinnig zijn

# Content hugging & compression resistance

- Hugging: niet groter willen worden
- Compression resistance: niet kleiner willen worden

Compression resistance: 750, Hugging: 250

Hugging: 249

Hugging: 251

Compres...stance: 749

Compression resistance: 751

**Hoe gebruik je  
AutoLayout?**

# AutoLayout in Xcode

- Zet AutoLayout aan in de File Inspector
- Gebruik de Assistant Editor voor Previews
- Let op de guides voor optimale positionering
- Zorg dat er geen warnings of errors overblijven



# Valkuilen

- Intrinsic Content Size
- Teveel of te weinig vaste afmetingen gebruiken
- Grote wijzigingen in je layout
- iOS 7 en lager vereisen soms PreferredMaxLayoutWidth

```
[Register ("MaxWidthLabel")]
public class MaxWidthLabel : UILabel
{
    public MaxWidthLabel (IntPtr handle) : base (handle) {}

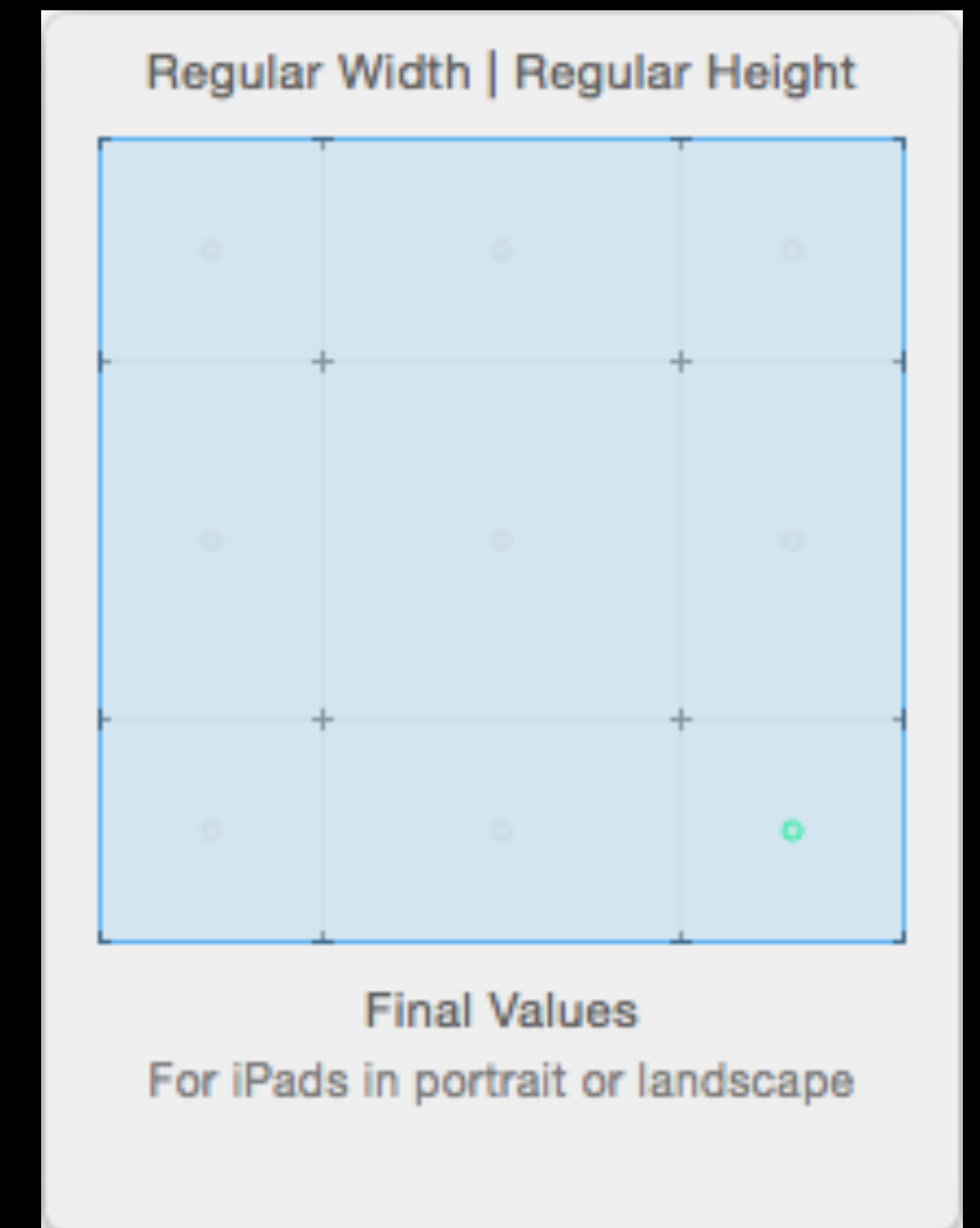
    public override void LayoutSubviews ()
    {
        base.LayoutSubviews ();

        PreferredMaxLayoutWidth = Bounds.Width;
    }
}
```

**Wat zijn Size  
Classes?**

# Wat zijn Size Classes?

- Ondersteuning voor meerdere schermformaten
- Geen noodzakelijke scheiding tussen phones en tablets (en andere apparaten)
- Backward compatible



# Segues & Traits

- Nieuwe relationship segues
- Bepaalde view types niet langer Idiom afhankelijk
- Bepaal zelf hoe je Size Classes indeelt met Traits
- Traits definiëren in de parent ViewController

**Hoe gebruik je  
Size Classes?**

**Vragen?**