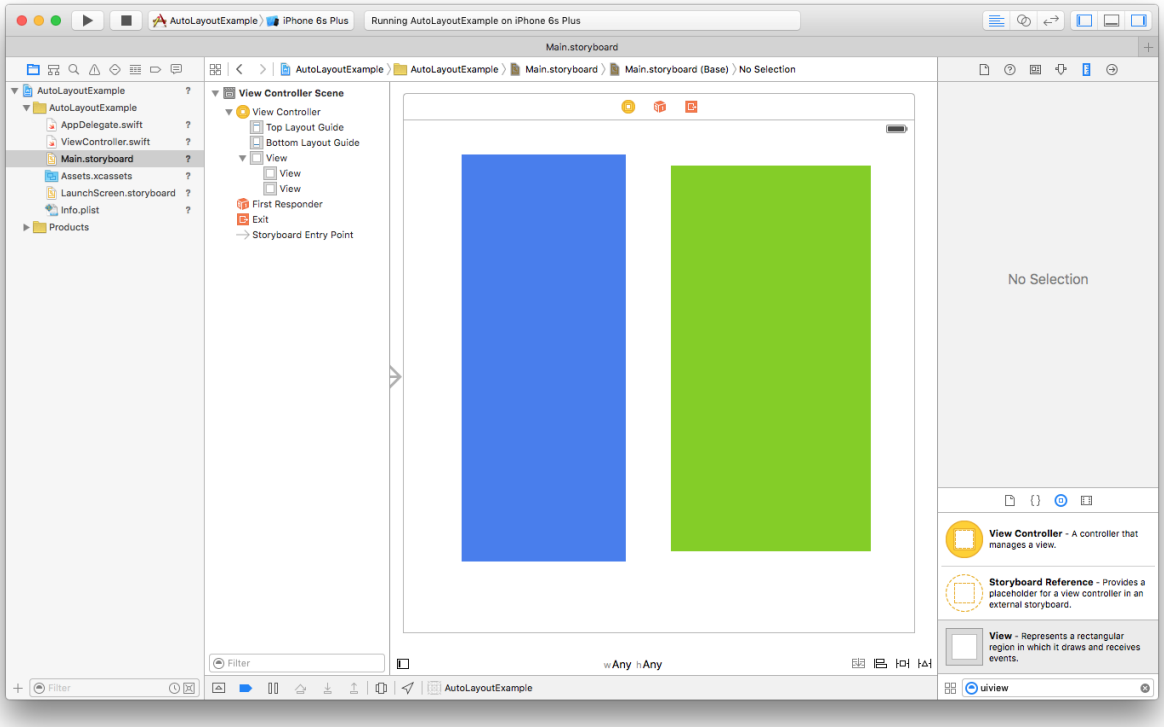
Creating Constraints in Storyboards

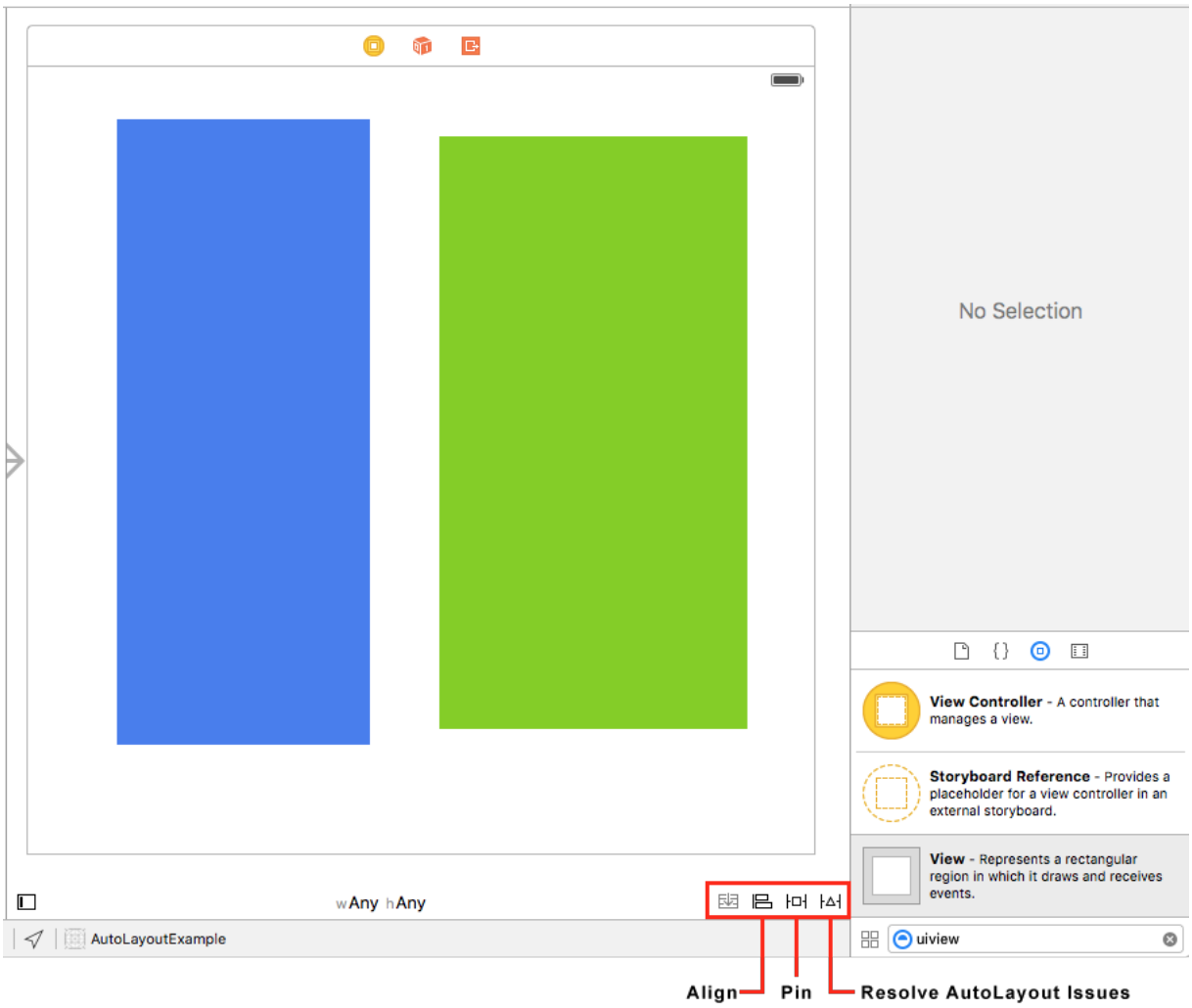
Interface Builder makes it very simple to create constraints in your storyboards. In the following steps, we will create the preceding layout using Auto Layout.

1. **Create a new Xcode Project**

2. **Create a Basic Layout**
    - Open your storyboard
    - Locate and drag two views from the Object Library to your scene
        - Tip: you can use the filter field below the Object Library to search for 'UIView'
    - Position the two views so they are side by side and not overlapping
    - You do not need to worry about setting the size and position of the views precisely
    - Style the views as you like



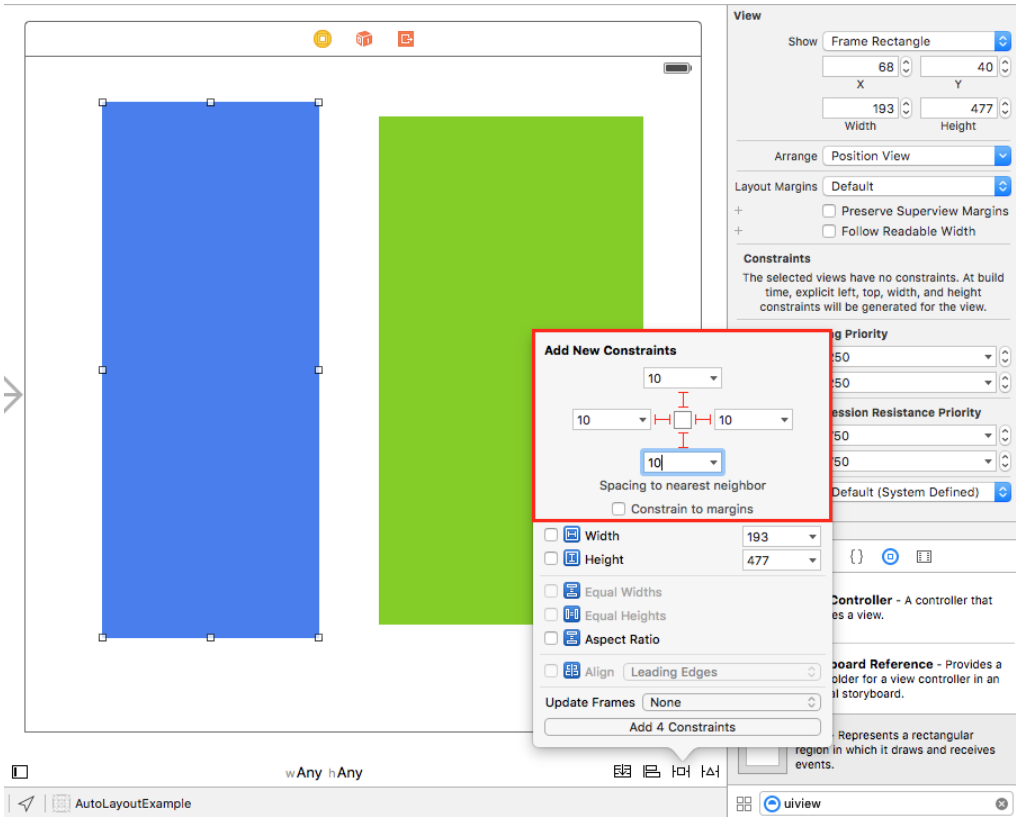
3. **The Auto Layout Menu**
    - The Auto Layout menu can be found in the bottom right corner of Interface Builder



    - The Align tool creates constraints that aligns the edges or centers of two views
    - The Pin tool creates constraints that define relationships between the edges of views and can also constrain the height and width of views
        - All required constraints can be created with this tool
    - The 'Resolve Auto Layout Issues' tool lets you update the frames of view(s) in your storyboard to match constraints that have been created or update constraint(s) based on the frames of your views in the storyboard

- We will be using the former as we want constraints that we define to update the position and size of views in our storybaord

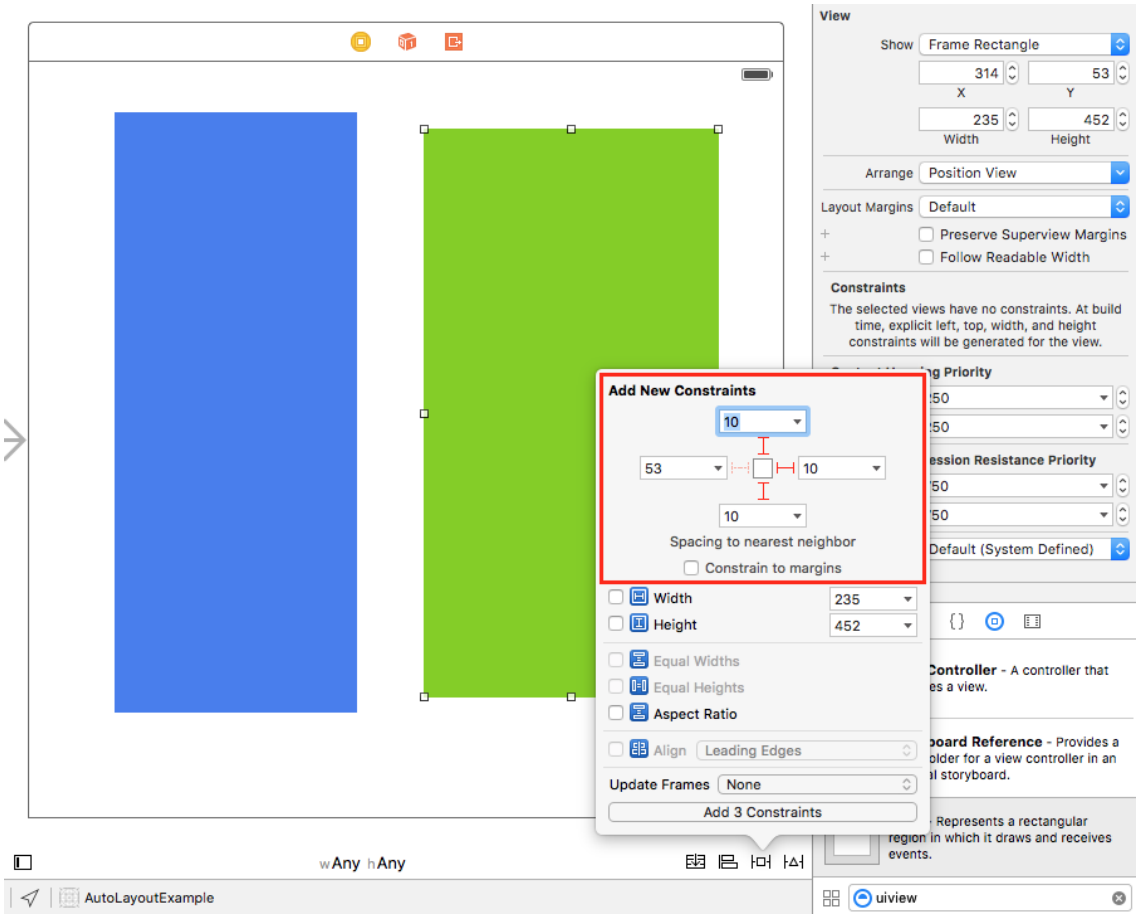
4. **Create Constraints**
    - We will begin by adding constraints to the left view:
        - Click the left view so it is selected and then click the Pin tool
        - Using the Pin tool, we want to define constraints that create 10 points of padding all around the view



        - To begin, uncheck 'Constrain to margins'
        - Next, create constraints by clicking the four spacing buttons so they become solid red
            - This constrains the spacing between the view and its neighbouring views (including its superview)
        - Next, open the dropdown for the top edge spacing and select 'View' instead of 'Top Layout Guide'
            - This lets us ignore the height of the status bar
        - Then, set the spacing for these constraints to be 10 points
        - Finally, click 'Add 4 Constraints' at the bottom of the Pin tool

- This creates the following constraints:
  - `LeftView.leading = Superview.leading + 10.0`
  - `LeftView.top = Superview.top + 10.0`
  - `Superview.bottom = LeftView.bottom + 10.0`
  - `RightView.leading = LeftView.trailing + 10.0`

- Next, we will create constraints for the right view:
  - Click the left view so it is selected and then click the Pin tool
  - Using the Pin tool, we want to define constraints that create 10 points of padding all around the view

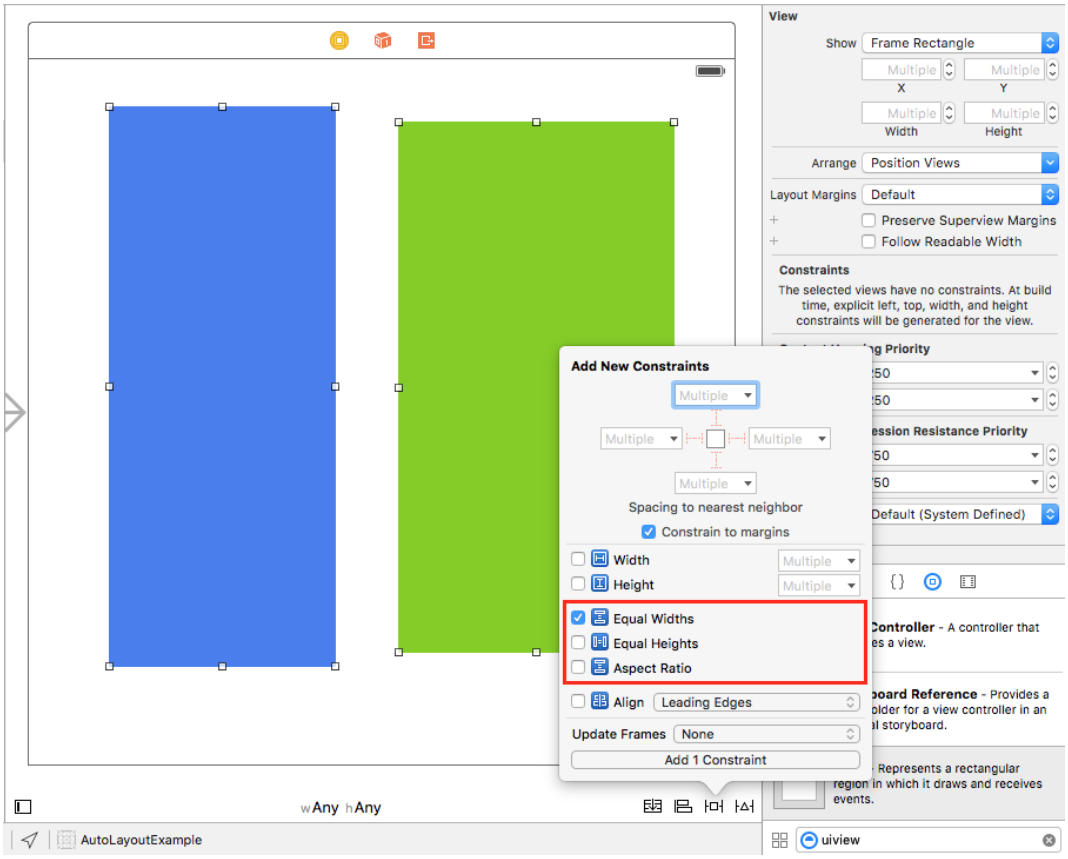


  - To begin, uncheck 'Constrain to margins'
    - Next, create constraints by clicking the the top, right, and bottom spacing buttons so they become solid red

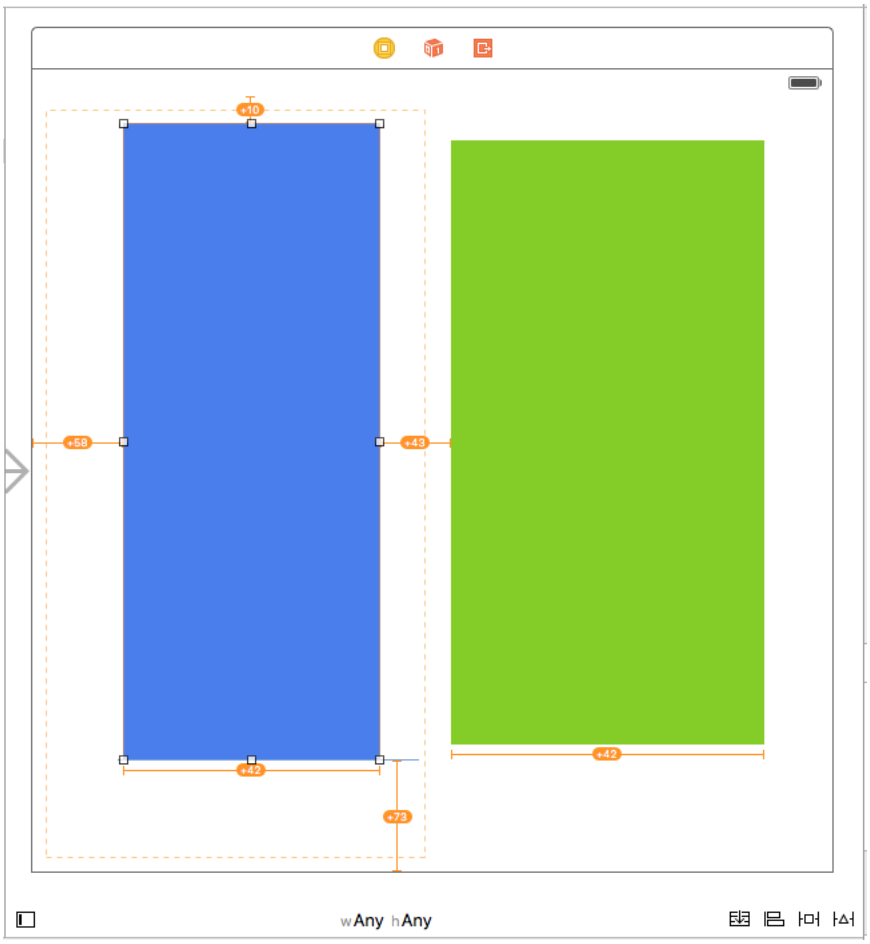
        - Note: we have already defined a constraint for the right view's leading edge in the previous step, so we should not create another - doing so would over-constrain the view
        - This constrains the spacing between the view and its neighbouring views (including its superview)
    - Then, set the spacing for these constraints to be 10 points
    - Finally, click 'Add 3 Constraints'
- This creates the following constraints:
    - `RightView.top = Superview.top + 10.0`
    - `Superview.trailing = RightView.trailing + 10.0`
    - `Superview.bottom = RightView.bottom + 10.0`

○ Finally, we need to add a constraint that sets the widths of the two views to be equal:

- Select the right and left views together, and then click the Pin tool
    - Check 'Equal Widths'
    - Finally, click 'Add 1 Constraint'

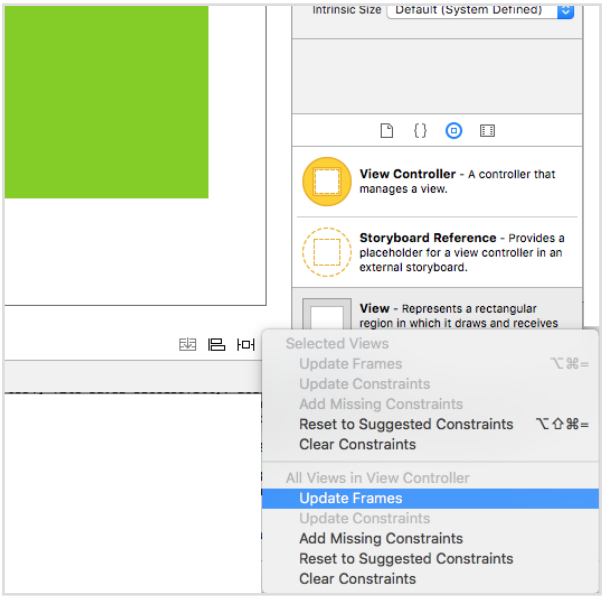


- Now that we've added all the required constraints, we want to have Auto Layout resolve the constraints and update the frames of our views
  - At this point, if you click on one of your views after you've added all the constraints, you should see something that looks like this (although the numbers may be different):

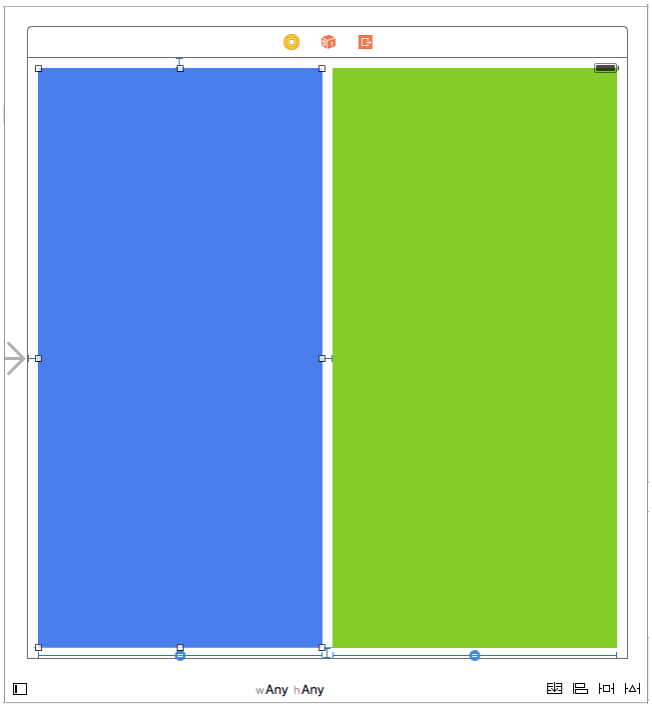


  - The orange dotted line shows where the view's frame should be based on the constraints applied to the view
  - The numbers in orange bubbles show how far off the current frames are relative to where the resolved constraints dictate they should be

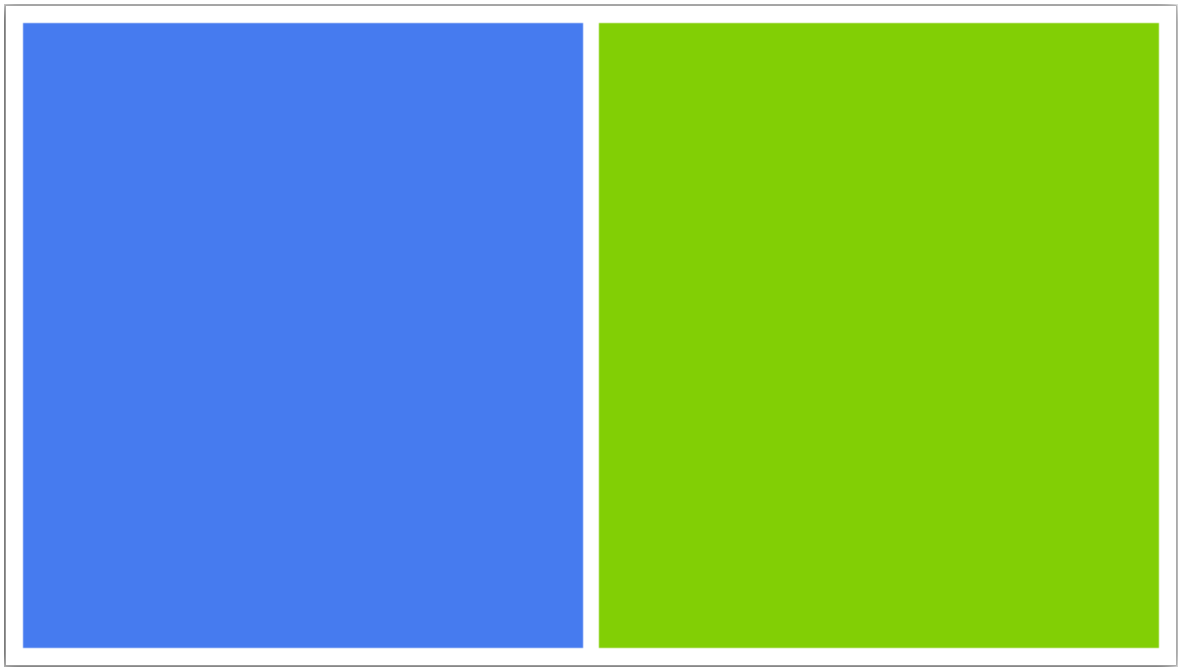
- In order to update the frames of the views in our storyboard so they represent the constraints we just made, deselect all views and then click the 'Resolve Auto Layout Issues' tool in the bottom right and select 'Update Frames' under 'All Views in View Controller'



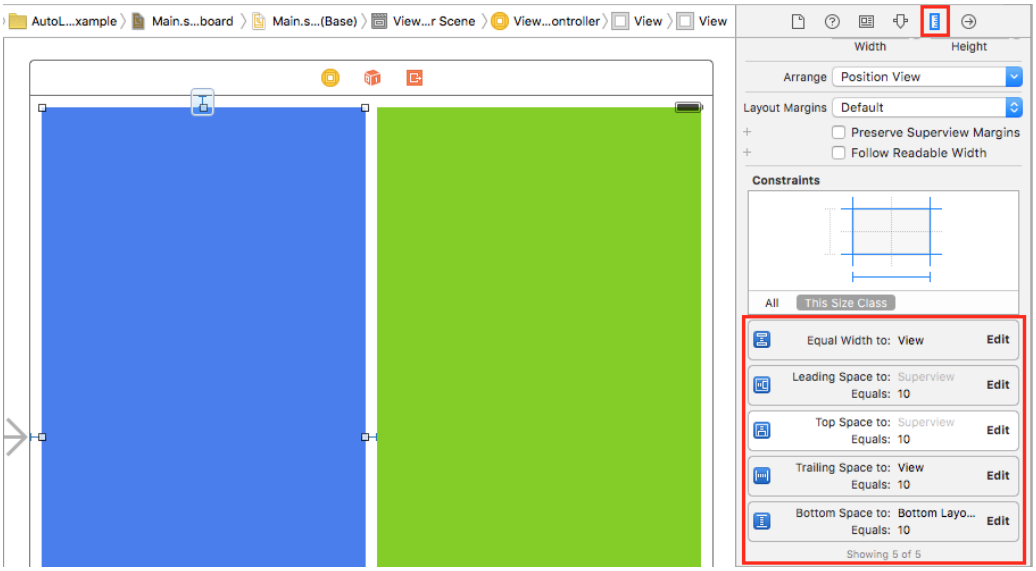
- Auto Layout should now update the frames of the views in your storyboards so they show the result of the constraints you just added
  - If you click on one of the views, there should no longer be orange dotted lines or orange numbers, since the layout should now match the constraints



5. **Test Layout**
    - Test your layout by running the app in a simulator
        - Change orientation by selecting Hardware -> Rotate Left/Right in the Simulator's Menu Bar
        - Your layout should adapt naturally based on the constraints you set



6. **Troubleshooting Auto Layout**
    - When troubleshooting Auto Layout issues, you can start by viewing the constraints that have been added to a view as follows:
        - Select any view, and then click on the Size inspector in the top right of the Utilities area
        - This will show a list of all constraints involving the selected view



        - Since constraints often involve more than one view, the same constraint can show up in this list when selecting different views
            - e.g. the 'equal widths' constraint will show up in the list for both the right and left views
        - If a constraint looks incorrect, double click the constraint and then hit the 'Delete' key on your keyboard
        - If you delete any incorrect constraints, recreate them correctly using the Pin or Align tools