

MOBILE DEVELOPMENT PRACTICE EXERCISE - GALLERY APP

William Martin
Head of Product, Floored

PRACTICE EXERCISE

LEARNING OBJECTIVES

- Navigate the Xcode and Interface Builder interfaces.
- Add Views to an app via the Library.
- Run an app in the iOS Simulator.

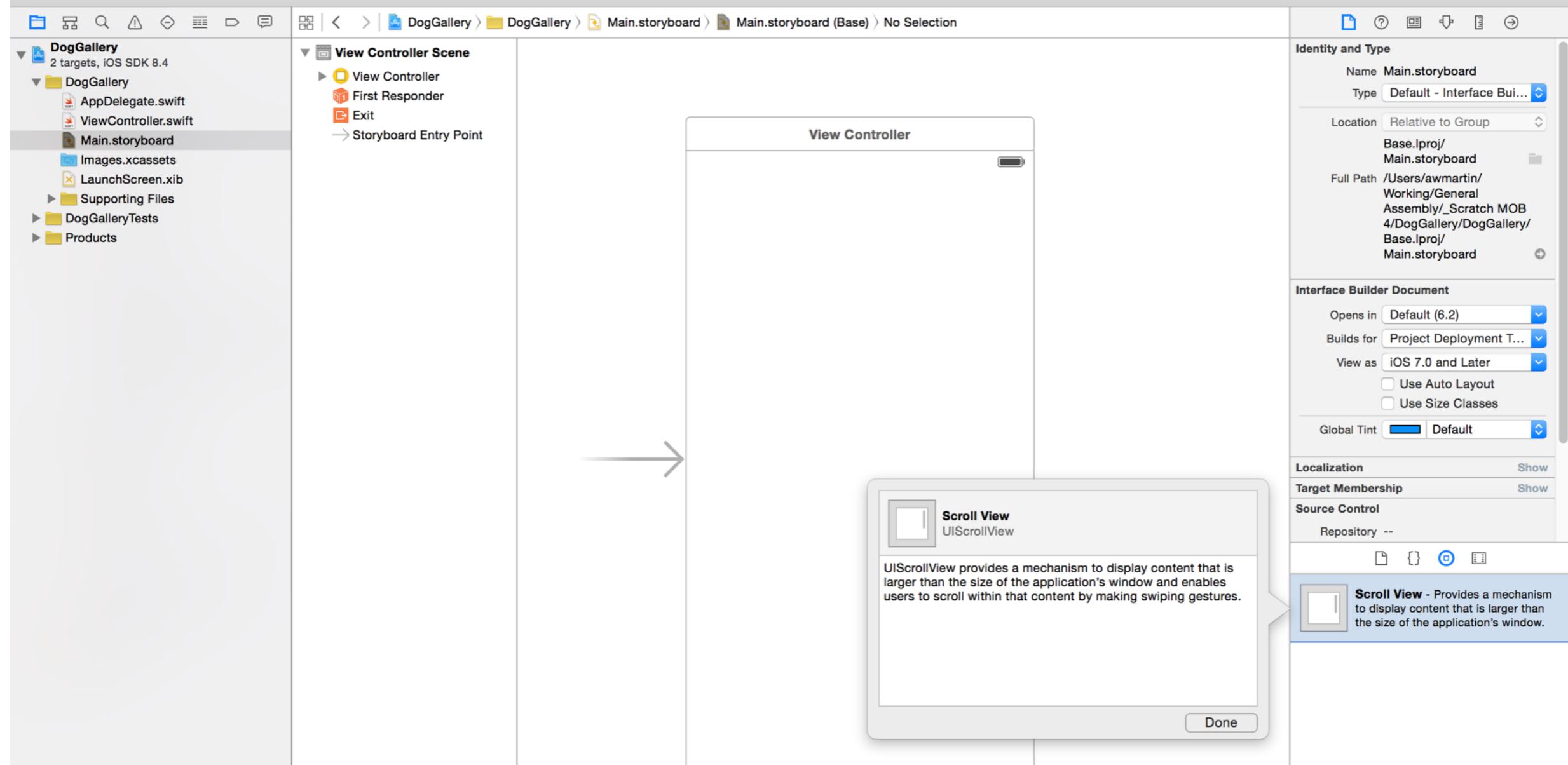
PRACTICE EXERCISE

GALLERY APP

PRACTICE EXERCISE

CREATE AN XCODE PROJECT

Follow the instructions included in Project Creation.pdf to create a new Xcode project.



Open the Document Outline by clicking on the small icon in the lower-left.



The screenshot shows the Xcode interface with the project 'DogGallery' open. The left sidebar shows the project structure with 'Main.storyboard' selected. The main canvas displays a 'View Controller' scene containing a 'ScrollView' component. A callout bubble provides a detailed description of what a UIScrollView does. The right side of the screen shows the 'Identity and Type' panel for 'Main.storyboard', listing its name, type, location, and full path. Below it are sections for 'Interface Builder Document', 'Localization', 'Target Membership', and 'Source Control'. A red arrow points from the bottom search bar to the search results, which show a 'ScrollView' component.

Search for “scroll” in the Library’s search bar. Find the “Scroll View.”

DogGallery > DogGallery > Main.storyboard > Main.storyboard (Base) > No Selection

Identity and Type

Name Main.storyboard
Type Default - Interface Bui...
Location Relative to Group
Base.lproj/
Main.storyboard
Full Path /Users/awmartin/
Working/General
Assembly/_Scratch MOB
4/DogGallery/DogGallery/
Base.lproj/
Main.storyboard

Interface Builder Document

Opens in Default (6.2)
Builds for Project Deployment T...
View as iOS 7.0 and Later
 Use Auto Layout
 Use Size Classes
Global Tint Default

Localization Show

Target Membership Show

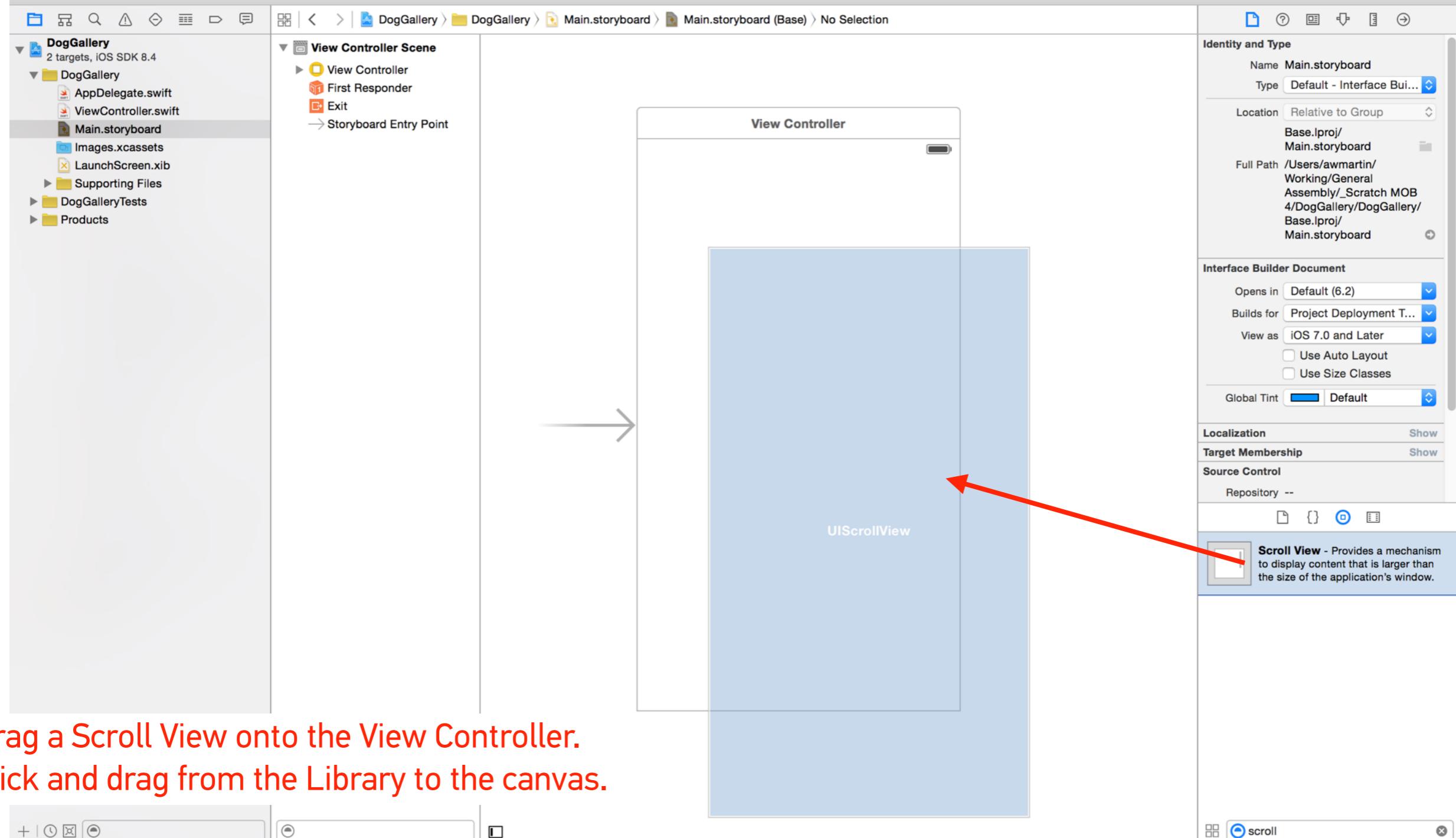
Source Control

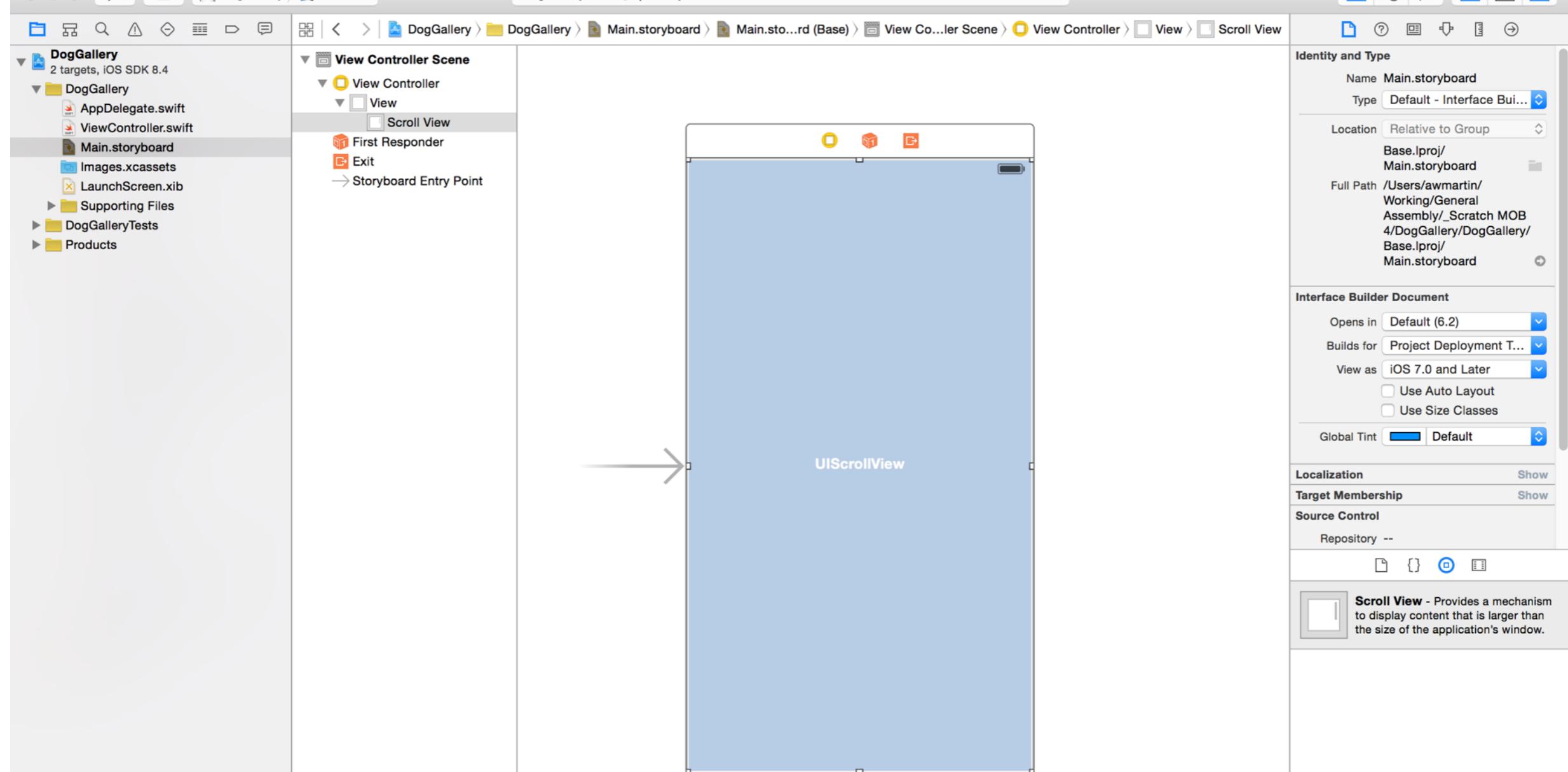
Repository --

File {} ○

ScrollView - Provides a mechanism to display content that is larger than the size of the application's window.

Drag a Scroll View onto the View Controller.
Click and drag from the Library to the canvas.





The screenshot shows the Xcode Interface Builder environment for a project named "DogGallery". The left sidebar displays the project structure with "Main.storyboard" selected. The document outline on the left shows a hierarchy: "View Controller Scene" containing "View Controller", "View", and "ScrollView". The storyboard preview in the center shows a large blue rectangular area labeled "UIScrollView". The right panel contains the "Identity and Type" inspector, showing "Name: Main.storyboard" and "Type: Default - Interface Builder Document". Below it are sections for "Interface Builder Document" (Opens in: Default (6.2), Builds for: Project Deployment Target, View as: iOS 7.0 and Later, Global Tint: Default), "Localization", "Target Membership", and "Source Control". A callout bubble on the right provides a detailed description of the "ScrollView" component.

Identity and Type

Name Main.storyboard

Type Default - Interface Builder Document

Location Relative to Group

Base.lproj/Main.storyboard

Full Path /Users/awmartin/Working/General Assembly/_Scratch MOB 4/DogGallery/DogGallery/Base.lproj/Main.storyboard

Interface Builder Document

Opens in Default (6.2)

Builds for Project Deployment Target

View as iOS 7.0 and Later

Global Tint Default

Localization

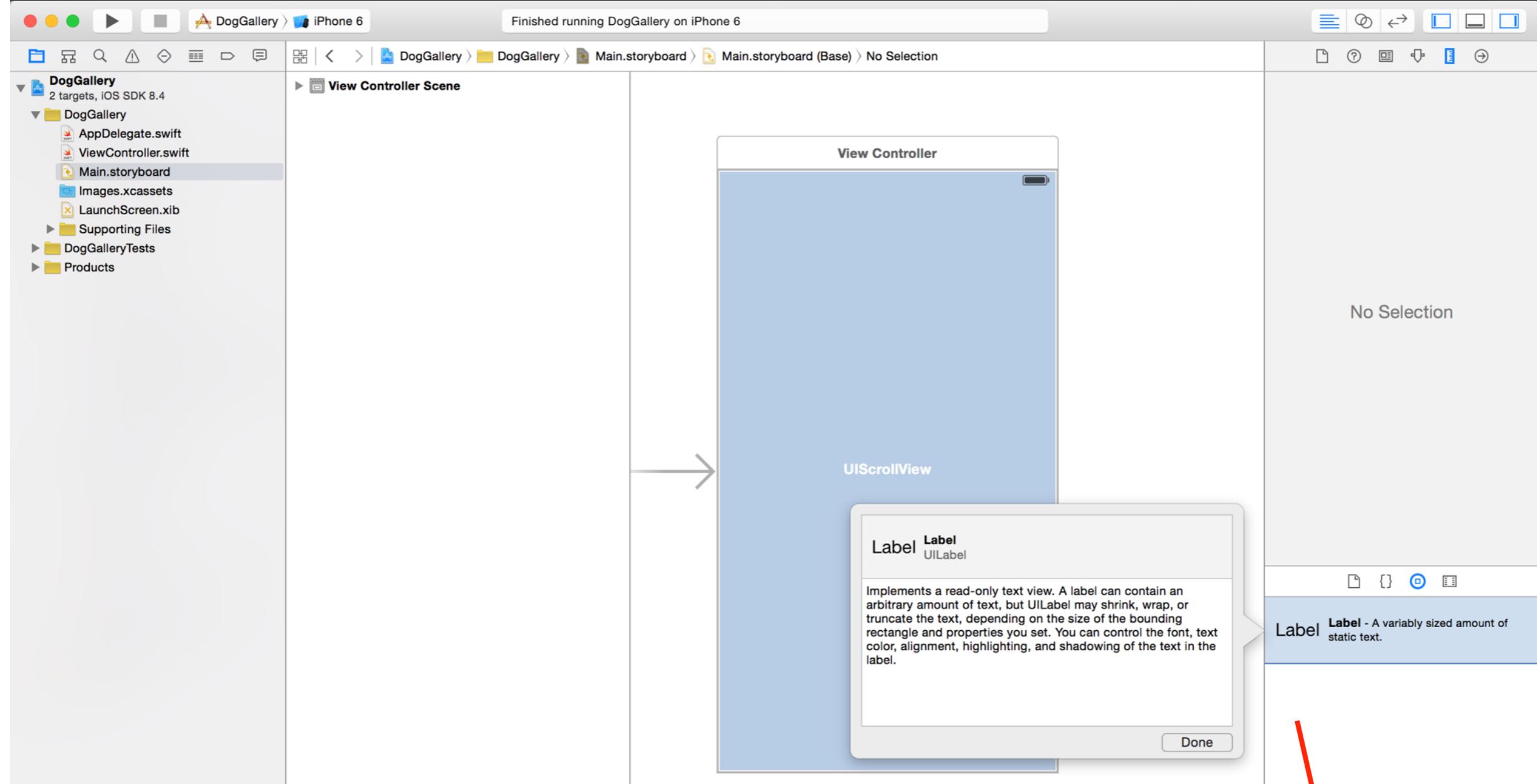
Target Membership

Source Control

Repository --

UIScrollView - Provides a mechanism to display content that is larger than the size of the application's window.

Position the Scroll View so it fills the entire View Controller view.



Search for “label” in the Library’s search bar.
Find the Label view.

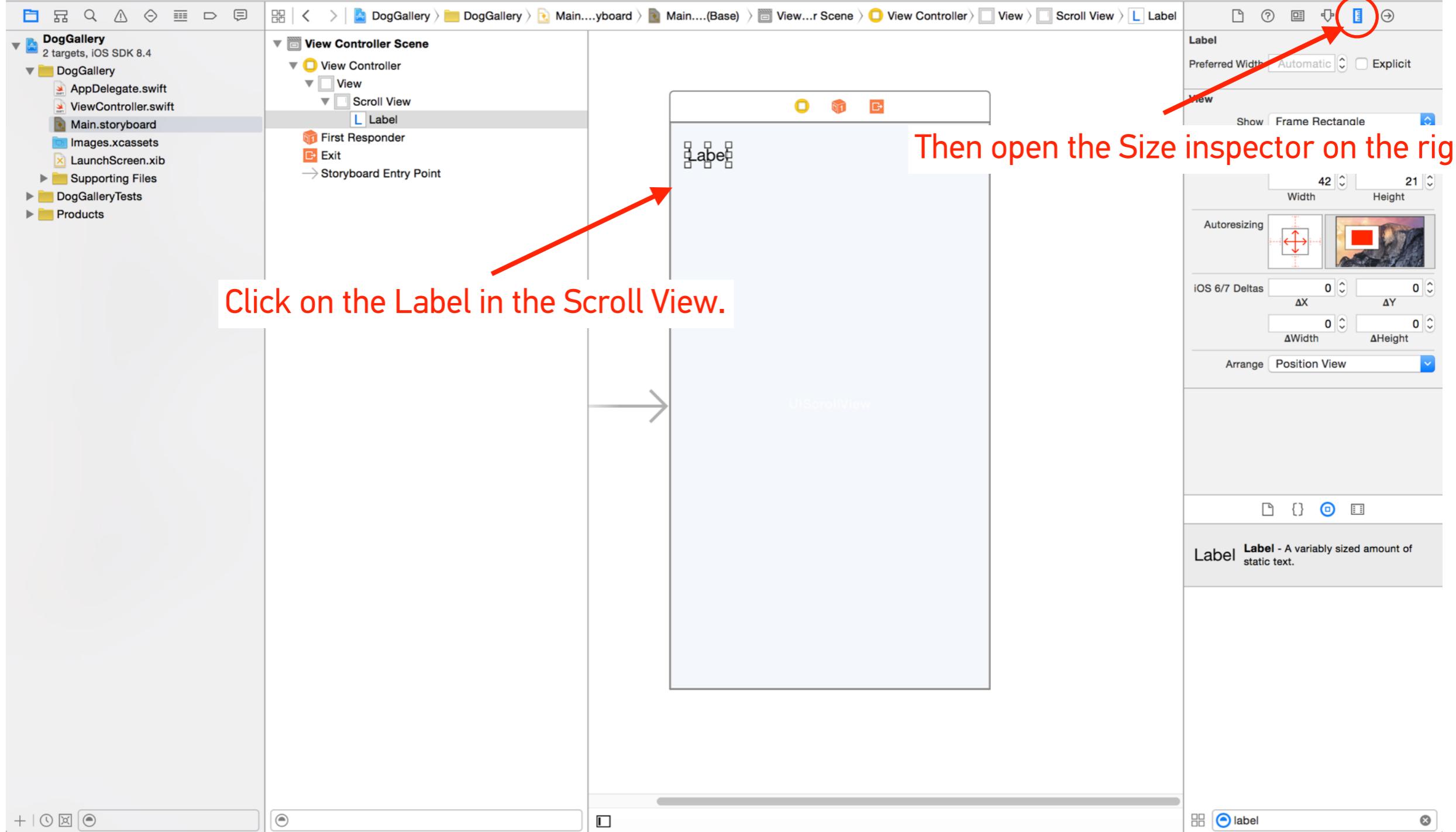
Add a Label to the app by dragging it from the Library to the UIScrollView.

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure for "DogGallery". The "Main.storyboard" file is selected.
- Document Outline:** Shows the "View Controller Scene".
- Editor:** Displays a "View Controller" scene with a blue background. Inside is a "UIScrollView" containing a single "Label". A red arrow points from the word "Label" in the text below to the "Label" component in the storyboard.
- Library:** At the bottom right, the "Label" component is selected in the library, with its description "Label - A variably sized amount of static text." visible.
- Text at the bottom:** The text "Add a Label to the app by dragging it from the Library to the UIScrollView." is displayed in red.

The Document Outline should reflect the fact that the Label is contained by the Scroll View.

The screenshot shows the Xcode interface with a storyboard file open. In the Document Outline (left sidebar), under the 'View Controller Scene' section, there is a 'Label' node. A red arrow points from the text above to this 'Label' node. In the main canvas, a large gray rectangle labeled 'UIScrollView' contains a smaller white rectangle labeled 'Label'. The storyboard entry point is also visible. On the right side, the Utilities panel displays settings for the selected 'Label' node, including its frame (X: 17, Y: 21, Width: 42, Height: 21), autoresizing mask, and a preview of the label's appearance.



DogGallery > iPhone 6 Finished running DogGallery on iPhone 6

DogGallery
2 targets, iOS SDK 8.4
DogGallery
AppDelegate.swift
ViewController.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
DogGalleryTests
Products

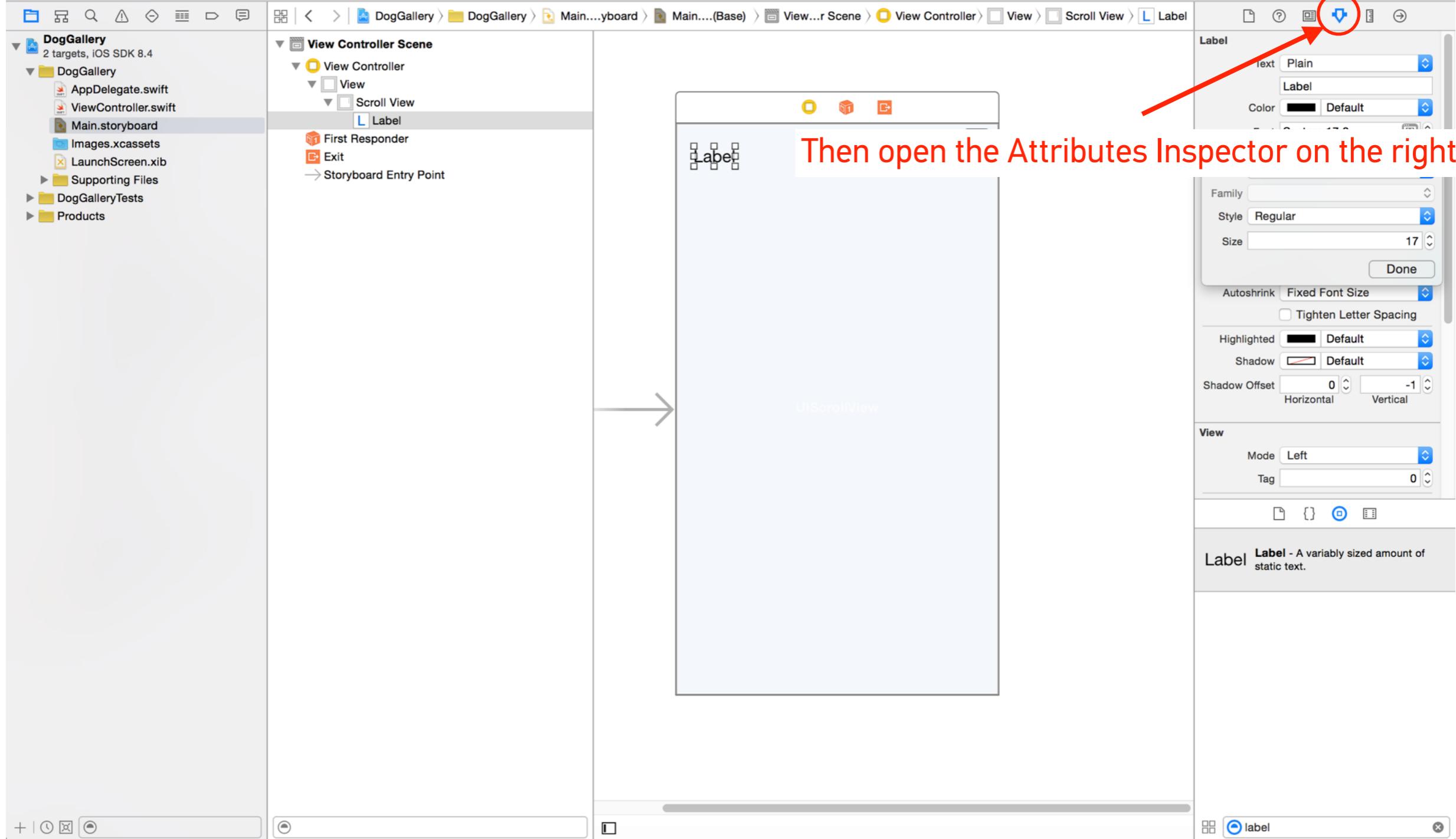
View Controller Scene
View Controller
View
ScrollView
Label
First Responder
Exit
Storyboard Entry Point

Label
Preferred Width Automatic ▾ Explicit

View
Show Frame Rectangle ▾
X 17 ▾ Y 22 ▾
Width 42 ▾ Height 21 ▾
Autoresizing
iOS 6/7 Deltas
ΔX 0 ΔY 0
ΔWidth 0 ΔHeight 0

Click on the Autoresizing panel to turn off all the “struts” and enable the “springs,” i.e. the double arrows in the middle.

For now, we’re going to make all of our Views scale proportionally to the main view.



DogGallery > iPhone 6 Finished running DogGallery on iPhone 6

DogGallery View Controller Scene Label

Main.storyboard View Controller Text Plain Color Default

View Font System 17.0 Font - System

ScrollView Label Family Regular

Label Style Size 17

First Responder Letter Spacing

Exit Shadow

Storyboard Entry Point Shadow Offset

Label - A variably sized amount of static text.

Change the font of the Label by first clicking on the “T” icon in the font selection.

DogGallery > iPhone 6

Finished running DogGallery on iPhone 6

Label

Text Plain ▾ Label

Color Black Default ▾ Custom

System System System Italic

Font Family

Style

Size

Autofit

Highlighted Default

Shadow Default

Shadow Offset 0 -1 Horizontal Vertical

View

Mode Left

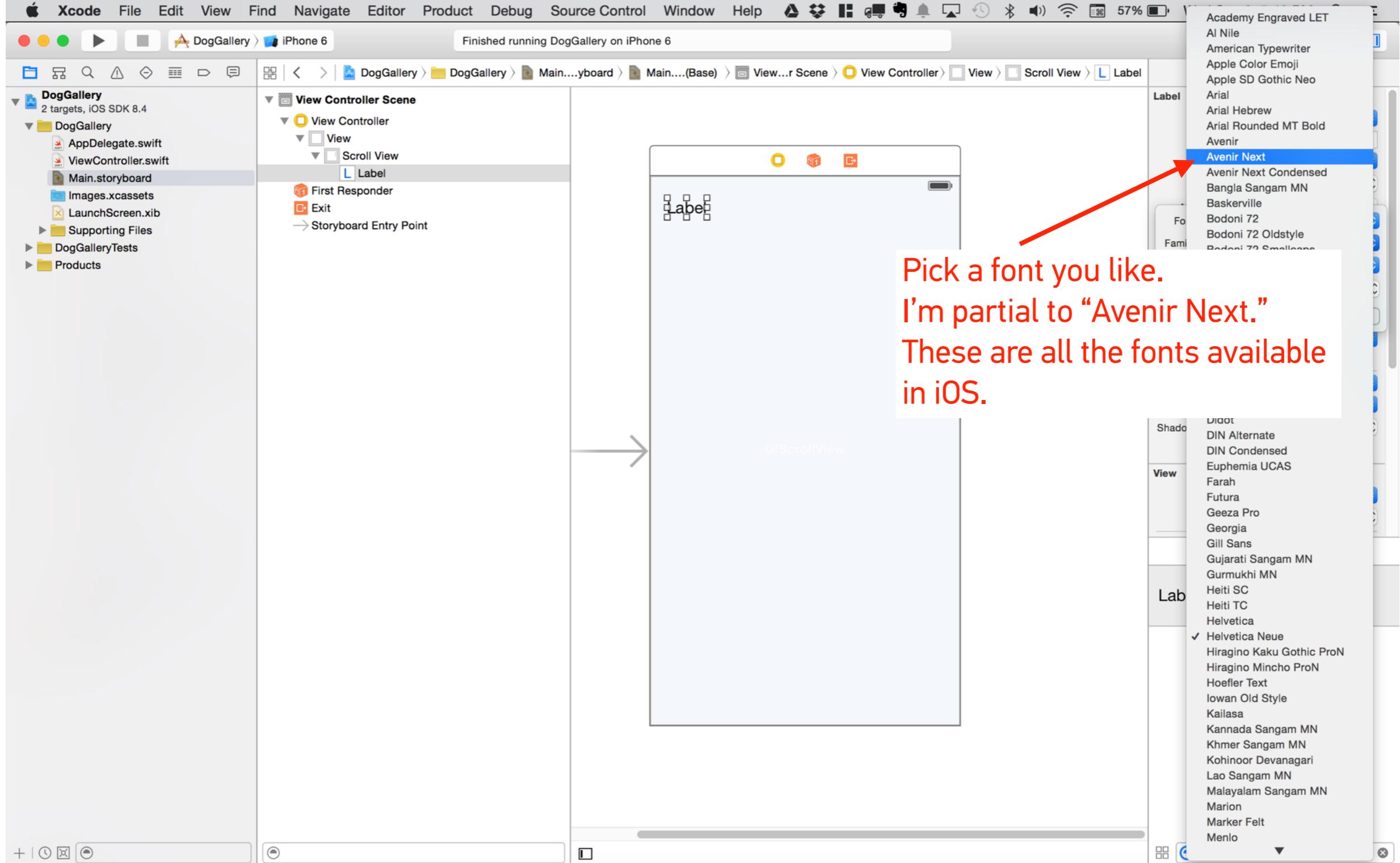
Tag 0

Label - A variably sized amount of static text.

label

Select “Custom”

The screenshot shows the Xcode interface with the storyboard editor open. On the left, the Project Navigator displays the 'DogGallery' project structure. In the center, the storyboard scene shows a scroll view containing a single label with the text 'Label'. On the right, the Attributes Inspector is open, showing settings for the selected 'Label' object. The 'Text' dropdown is set to 'Plain', and the 'Color' dropdown is set to 'Custom'. A red arrow points from the text 'Select "Custom"' to the 'Color' dropdown menu, which is currently expanded to show the 'Custom' option. The 'Label' object in the scroll view has a bounding box of approximately [150, 480, 680, 750]. The scroll view itself has a bounding box of approximately [480, 150, 680, 750]. The Attributes Inspector is located at the bottom right of the interface.



Label

Text Plain Label Color Default Font Avenir Next Regula... Alignment Lines Behavior Enabled Baseline Align Baselines Line Breaks Truncate Tail Autoshrink Fixed Font Size Tighten Letter Spacing Highlighted Default Shadow Default Shadow Offset 0 -1 Horizontal Vertical

View

Mode Left Tag 0

Label Label - A variably sized amount of static text.

W: 236.0 H: 21.0

Label

First Responder Exit Storyboard Entry Point

View Controller Scene

View Controller View Scroll View Label

Main.storyboard

Main....yboard Main....(Base) View...r Scene View Controller View Scroll View Label

DogGallery

2 targets, iOS SDK 8.4

DogGallery

AppDelegate.swift ViewController.swift Main.storyboard Images.xcassets LaunchScreen.xib Supporting Files

DogGalleryTests Products

Make the label wider by dragging on the tiny handle on the right.

A screenshot of the Xcode interface showing a storyboard scene. The storyboard tree on the left shows a 'Main.storyboard' file selected. In the center canvas, there is a scroll view containing a single label. A red arrow points to the right edge of the label's bounding box, which has dimensions W: 236.0 and H: 21.0 displayed above it. The rightmost pane of the interface is the Attribute Inspector, showing various properties for the selected 'Label' object, such as Text (Plain), Color (Default), and Font (Avenir Next Regular). A large red text overlay in the center says 'Make the label wider by dragging on the tiny handle on the right.' The bottom status bar shows the Xcode version (5.0), battery level (57%), and current date and time (Wed Sep 9 5:44 PM).

Label

Text Plain

Color Default

Font Avenir Next Regula...

Alignment

Lines 1

Behavior Enabled

Baseline Align Baselines

Line Breaks Truncate Tail

Autoshrink Fixed Font Size

Highlighted Default

Shadow Default

Shadow Offset 0 -1

Horizontal Vertical

View

Mode Left

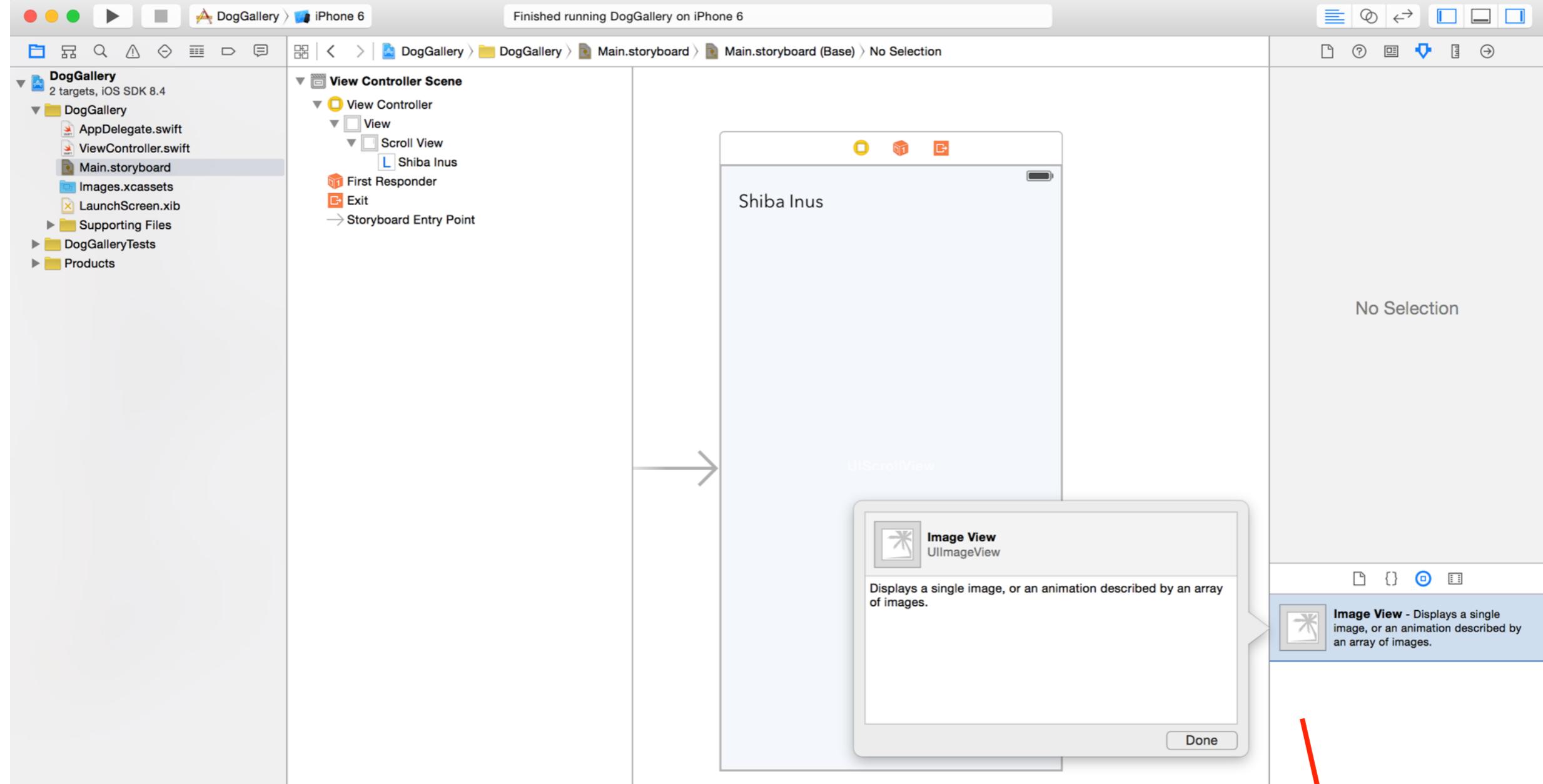
Tag 0

Label - A variably sized amount of static text.

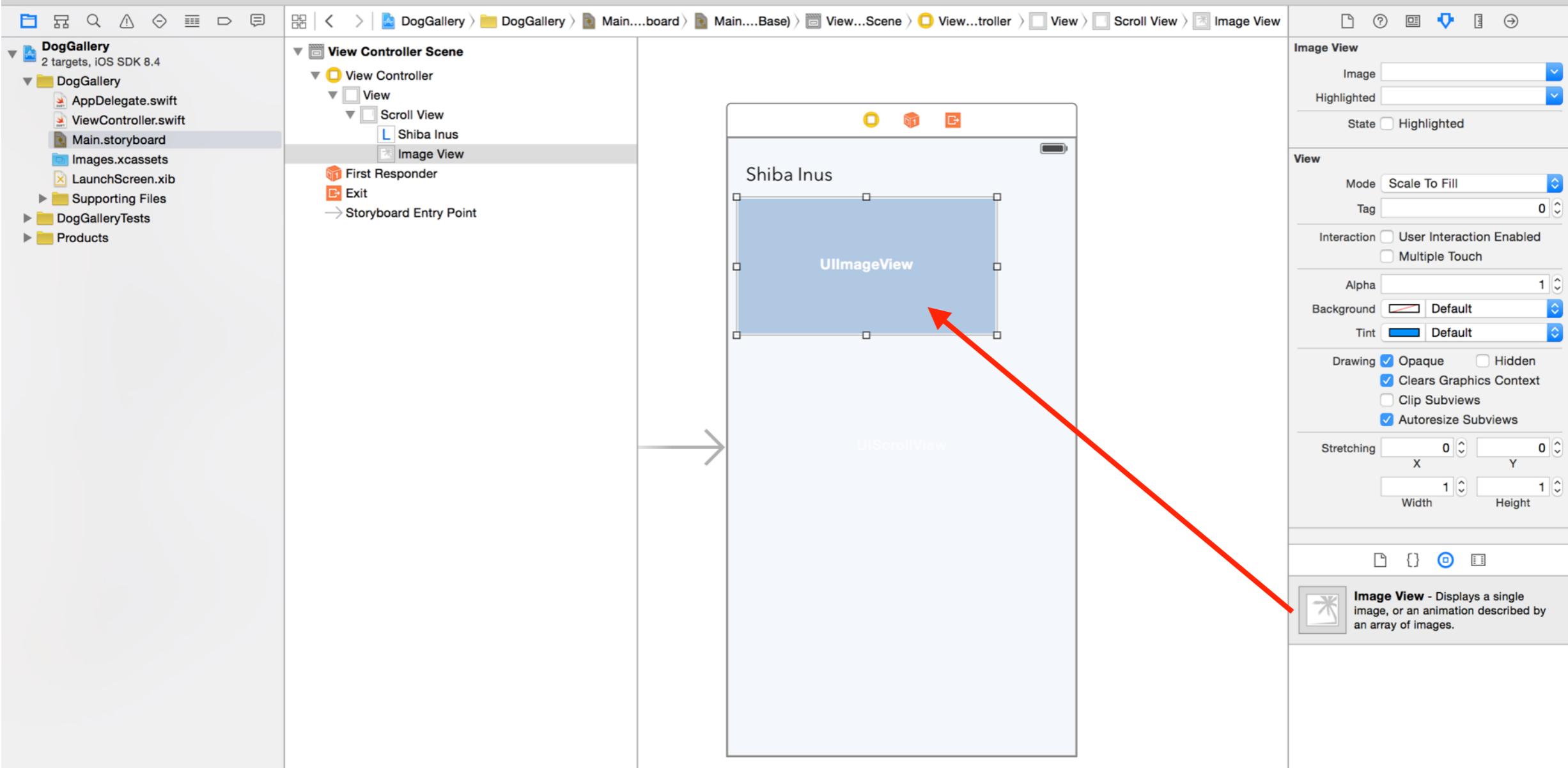
Shiba Inus

Change the text by first double-clicking on the Label, then typing.

Add a title that describes the images you'll be inserting.



Search for “image” in the Library search bar.
Look for Image View.



Add an Image View to the Scroll View by dragging it from the Library onto the canvas.

DogGallery > iPhone 6 Finished running DogGallery on iPhone 6

DogGallery
2 targets, iOS SDK 8.4
DogGallery
AppDelegate.swift
ViewController.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
DogGalleryTests
Products

View Controller Scene
View Controller
View
ScrollView
Shiba Inus
Image View
First Responder
Exit
Storyboard Entry Point

W: 104.0
H: 104.0

Shiba Inus

UIImageView

UIScrollView

Image View
Image
Highlighted
State Highlighted

View
Mode Scale To Fill
Tag 0
Interaction User Interaction Enabled
 Multiple Touch
Alpha 1
Background Default
Tint Default
Drawing Opaque Hidden
 Clears Graphics Context
 Clip Subviews
 AutoresizesSubviews
Stretching 0 0
X 1 Y 1
Width Height

Image View - Displays a single image, or an animation described by an array of images.

Resize and position the Image View to be square-like.
As you resize, note the tooltip that tells you how big it is.

These units are “points,” which are a special unit used by iOS that enable us to not worry about the actual pixel density or resolution of any particular device.

Note also the guides that appear as you drag the Image View around to help you align different Views.

shiba inu - Google Search

https://www.google.com/search?q=shiba+inu&espv=2&biw=1680&bih=928&source=lnms&tbo=isch&sa=X&ved=0CAYQ_AUoAWo...

William

Apps Inbox 8 Momomo Feedly NoDesk IFTTT WyzAnt Songkick General Assembly Finance Social Other Bookmarks

shiba inu

Web Images Videos Shopping News More Search tools William SafeSearch

Puppies Doge Black White Smile

Perform a Google Image search for something, like a dog breed or kittens. Download 5 to 10 images and place them in a folder on your Desktop.

The screenshot shows a Google Images search results page for 'shiba inu'. At the top, there's a navigation bar with links for Web, Images (which is selected), Videos, Shopping, News, More, and Search tools. Below the search bar, there are five categories with thumbnail images: Puppies, Doge, Black, White, and Smile. The main content area displays a grid of Shiba Inu dog images. The first row contains five images: a smiling Shiba Inu, a Shiba Inu in a red harness, a close-up of a Shiba Inu's face, a Shiba Inu standing, and a black and tan Shiba Inu sitting. The second row contains five images: a Shiba Inu standing, a Shiba Inu standing, a black and tan Shiba Inu standing, a Shiba Inu walking in snow, and a close-up of a Shiba Inu's face. The third row contains five images: a close-up of a Shiba Inu's face, a Shiba Inu standing, a Shiba Inu walking, a Shiba Inu standing, and a Shiba Inu wearing a suit and tie. The bottom right corner of the image shows two small, partially visible images of Shiba Inus.

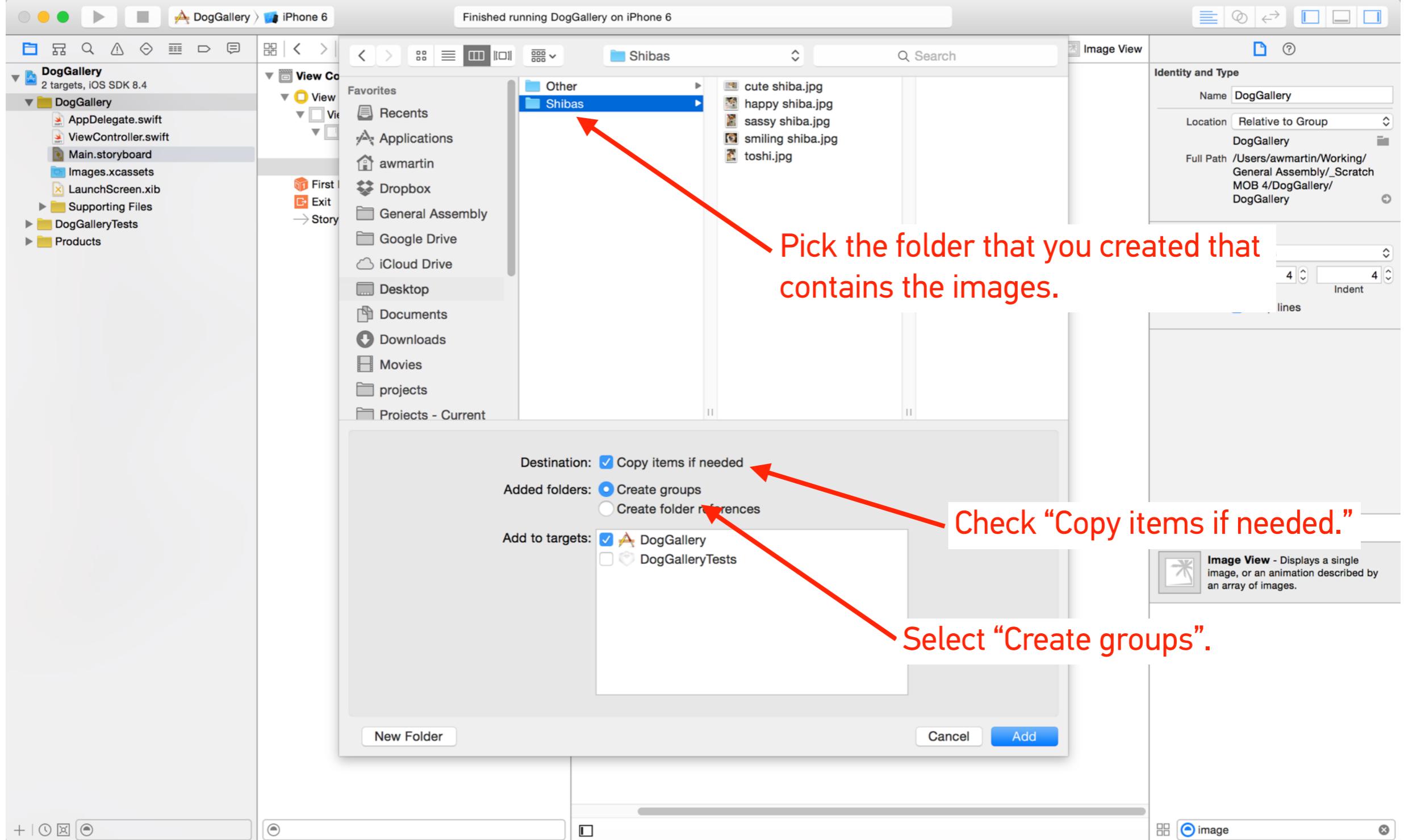


Right-click (or two-finger click, or CTRL+click) on the Group (yellow folder) that matches the name of your app. In my case, this is “DogGallery.”

Select Add Files to “YourApp” ...

The screenshot shows the Xcode Project Navigator on the left, displaying the "DogGallery" project structure. A context menu is open over the "DogGallery" group, with the "Add Files to 'DogGallery'..." option highlighted. A red arrow points from the text "Select Add Files to ‘YourApp’ ..." to this menu item. Another red arrow points from the text "Right-click (or two-finger click, or CTRL+click) on the Group (yellow folder) that matches the name of your app. In my case, this is ‘DogGallery.’" to the "DogGallery" group in the Project Navigator.

The main canvas shows a "View Controller Scene" containing a "UIScrollView" which contains a single "UIImageView". The Identity and Type inspector on the right shows the "Name" field set to "DogGallery". The Utilities panel on the right shows the "Image View" component details.



DogGallery | Build DogGallery: Succeeded | Today at 5:37 PM

DogGallery > DogGallery > Main.storyboard > Main.storyboard (Base) > No Selection

Identity and Type

Name: Shibas

Location: Relative to Group
Shibas

Full Path: /Users/awmartin/Working/General Assembly/_Scratch MOB 4/DogGallery/DogGallery/Shibas

Text Settings

Indent Using: Spaces

Widths: Tab 4 Indent 4

Wrap lines

View Controller Scene

View Controller

Shiba Inus

cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg

Check to make sure a new Group has been added that matches the original name of your folder. Expand it to view the available image names.

UIScrollView

View Controller - A controller that supports the fundamental view-management model in iOS.

Navigation Controller - A controller that manages navigation through a hierarchy of views.

Table View Controller - A controller that manages a table view.

Tab Bar Controller - A controller that manages a set of view controllers that represent tab ba...

Split View Controller - A

DogGallery > iPhone 6 DogGallery | Build DogGallery: Succeeded | Today at 5:37 PM

DogGallery | View Controller Scene

View Controller Scene

View Controller

View

ScrollView

Shiba Inus

ImageView

First Responder

Exit

Storyboard Entry Point

Shiba Inus

UIImageView

Note that the Document Outline has the Image View as contained by the Scroll View.

UIScrollView

Image View

Image: cute shiba.jpg

Highlighted: happy shiba.jpg

State: sassy shiba.jpg

Mode: smiling shiba.jpg

Tag: toshi.jpg

Interaction: User Interaction Enabled

Multiple Touch

Alpha: 1

Background: Default

Tint: Default

Drawing: Opaque (checked), Hidden (unchecked)

Clears Graphics Context (checked)

Clip Subviews (unchecked)

Autoresizing Subviews (checked)

Stretching: X: 0, Y: 0, Width: 1, Height: 1

cute shiba.jpg

happy shiba.jpg

sassy shiba.jpg

smiling shiba.jpg

toshi.jpg

Select the Image View.

In the Attributes Inspector, click on the “Image” property, where you should see a dropdown of the images you’ve imported.

Select one of them.

The screenshot shows the Xcode interface with the storyboard editor open. In the center, there's a scroll view containing a single image view labeled "UIImageView". To the right of the scroll view, the "Attributes Inspector" is visible, showing the "Image" dropdown menu with several options: "cute shiba.jpg", "happy shiba.jpg", "sassy shiba.jpg", "smiling shiba.jpg", "toshi.jpg", and "highlighted image". A red arrow points from the text "Select the Image View." to the UIImageView in the storyboard. Another red arrow points from the text "In the Attributes Inspector, click on the ‘Image’ property, where you should see a dropdown of the images you’ve imported." to the "Image" dropdown in the Attributes Inspector. Below the storyboard, a "File Inspector" is partially visible, showing a list of imported image files: "cute shiba.jpg", "happy shiba.jpg", "sassy shiba.jpg", "smiling shiba.jpg", and "toshi.jpg".

DogGallery > iPhone 6 DogGallery | Build DogGallery: Succeeded | Today at 5:37 PM

DogGallery 2 targets, iOS SDK 8.4

DogGallery Shibas

- cute shiba.jpg
- happy shiba.jpg
- sassy shiba.jpg
- smiling shiba.jpg
- toshi.jpg

AppDelegate.swift

ViewController.swift

Main.storyboard

Images.xcassets

LaunchScreen.xib

Supporting Files

Info.plist

DogGalleryTests

Products

View Controller Scene

View Controller

View

ScrollView

Shiba Inus

toshi.jpg

First Responder

Exit

Storyboard Entry Point

Image View

Image toshi.jpg

Highlighted

State Highlighted

View

Mode Scale To Fill

Tag 0

Interaction User Interaction Enabled

Multiple Touch

Alpha 1

Background Default

Tint Default

Drawing Opaque Hidden

Clears Graphics Context

Clip Subviews

AutoresizesSubviews

Stretching 0 X 0 Y

Width 1 Height 1

cute shiba.jpg

happy shiba.jpg

sassy shiba.jpg

smiling shiba.jpg

toshi.jpg

You should see the image pop up in the Image View.

Note how the image is distorted.

The screenshot shows the Xcode interface with a storyboard project named 'DogGallery'. The storyboard scene contains a scroll view with a placeholder image labeled 'Shiba Inus'. A specific image file, 'toshi.jpg', is selected in the document outline, and its preview is shown in the scroll view. A red arrow points from the text 'You should see the image pop up in the Image View.' to the scroll view. Below the scroll view, the text 'Note how the image is distorted.' is displayed. The right side of the screen shows the Attributes Inspector for the scroll view, where the 'Mode' is set to 'Scale To Fill'. The scroll view's width and height are both set to 1. The scroll view is positioned at the bottom of the storyboard scene, overlapping the text 'Note how the image is distorted.'

DogGallery
2 targets, iOS SDK 8.4
DogGallery
Shibas
cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg
AppDelegate.swift
ViewController.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
Info.plist
DogGalleryTests
Products

View Controller Scene
View Controller
View
ScrollView
Shiba Inus
toshi.jpg
First Responder
Exit
Storyboard Entry Point

Image View
Image toshi.jpg
Highlighted
State Highlighted

View
Mode Scale To Fill
 Aspect Fit
 Aspect Fill
Interaction
Redraw
Center
Top
Bottom
Left
Right
Top Left
Right
From Left
From Right

To resize the image properly,
Click on the “Mode” dropdown,
and select “Aspect Fill.”

This maintains the aspect ratio
of the image and “fills” the entire
Image View.

You can experiment with others
to see what they do.

smiling shiba.jpg
Anilin

DogGallery > iPhone 6 Finished running DogGallery on iPhone 6

DogGallery
2 targets, iOS SDK 8.4
DogGallery
Shibas
cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg
AppDelegate.swift
ViewController.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
Info.plist
DogGalleryTests
Products

View Controller Scene
View Controller
View
ScrollView
Shiba Inus
toshi.jpg
First Responder
Exit
Storyboard Entry Point

Image View
Image toshi.jpg
Highlighted
State Highlighted

View
Mode Aspect Fill
Tag 0
Interaction User Interaction Enabled
 Multiple Touch
Alpha 1
Background Default
Tint Default
Drawing Opaque Hidden
 Clears Graphics Context
 Clip Subviews
 Autoresizes Subviews
Stretching X 0 Y 0
Width 1 Height 1

Check “Clip Subviews.”
This ensures that our image doesn't spill outside of the square Image View bounds.

cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg

Again, we're going to change the layout rules for this View.

Open the Size Inspector.

DogGallery
2 targets, iOS SDK 8.4
DogGallery
Shibas
cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg
AppDelegate.swift
ViewController.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
Info.plist
DogGalleryTests
Products

View Controller Scene
View Controller
View
ScrollView
Shiba Inus
toshi.jpg
First Responder
Exit
Storyboard Entry Point

Shiba Inus

UIScrollView

View
Show Frame Rectangle
X: 8 Y: 54
Width: 104 Height: 104
Autoresizing
iOS 6/7 Deltas
ΔX: 0 ΔY: 0
ΔWidth: 0 ΔHeight: 0
Arrange Position View

cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg

DogGallery > iPhone 6 Finished running DogGallery on iPhone 6

DogGallery
2 targets, iOS SDK 8.4
DogGallery
Shibas
cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg
AppDelegate.swift
ViewController.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
Info.plist
DogGalleryTests
Products

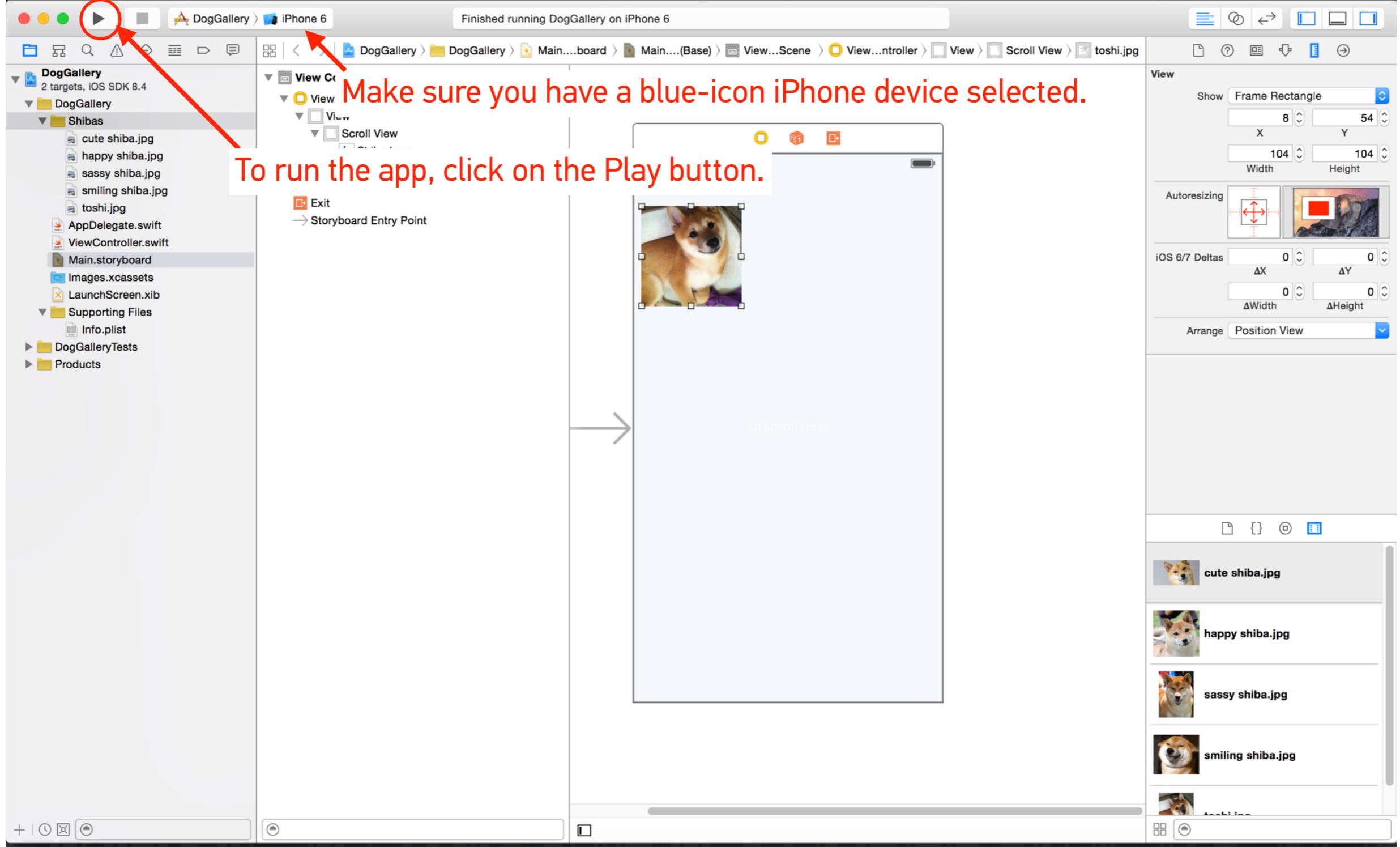
View Controller Scene
View Controller
View
ScrollView
Shiba Inus
toshi.jpg
First Responder
Exit
Storyboard Entry Point

View
Show Frame Rectangle
X 8 Y 54
Width 104 Height 104
Autoresizing
iOS 6/7 Deltas
ΔX 0 ΔY 0
ΔWidth 0 ΔHeight 0
Arrange Position View

Shiba Inus

Click the struts off and the springs on, so the Autoresizing panel looks like this.

cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg



The iOS simulator will appear. Congratulations!
If it doesn't appear, wait for a bit, then check for the Dock icon.
Click on that icon to focus the Simulator.

DogGallery

2 targets, iOS SDK 8.4

DogGallery

Shibas

- cute shiba.jpg
- happy shiba.jpg
- sassy shiba.jpg
- smiling shiba.jpg
- toshi.jpg

AppDelegate.swift

ViewController.swift

Main.storyboard

Images.xcassets

LaunchScreen.xib

Supporting Files

Info.plist

DogGalleryTests

Products

View Controller Scene

View Controller

View

ScrollView

Shiba Inus

toshi.jpg

First Responder

Exit

Storyboard Entry Point

Shiba Inus

UIScrollView

Carrier 5:52 PM

Shiba Inus

smiling shiba.jpg

ba.jpg

ba.jpg

smiling shiba.jpg

ba.jpg

smiling shiba.jpg

DogGallery

DogGallery > iPhone 6 Finished running DogGallery on iPhone 6

DogGallery
2 targets, iOS SDK 8.4
DogGallery
Shibas
cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg
AppDelegate.swift
ViewController.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
Info.plist
DogGalleryTests
Products

View Controller Scene
View Controller
View
ScrollView
Shiba Inus
toshi.jpg
First Responder
Exit
Storyboard Entry Point

Custom Class
Class UIScrollView
Module None

Identity
Restoration ID

User Defined Runtime Attributes
keyPath Boolean

Document
Label Xcode Specific Label
Object ID ENU-OC-BzI
Lock Inherited - (Nothing)
Notes
No Font

cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg

The Scroll View doesn't scroll yet. So let's configure it to do so, so we can add more images than can be shown.

Open the Identity Inspector.

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure for "DogGallery".
- Document Outline:** Shows the "View Controller Scene" hierarchy: View Controller → View → Scroll View.
- Preview Area:** Displays a simulated iPhone 6 screen titled "Shiba Inus" showing a Shiba Inu image.
- Identity Inspector:** Opened on the right, showing the "Custom Class" section set to "UIScrollView".
- File Inspector:** Opened on the right, showing "Document" settings like "Label" and "Object ID".
- Attributes Inspector:** Opened on the right, showing "Runtime Attributes" for "Type" and "Value".
- Assistant Editor:** Shows a list of images: "cute shiba.jpg", "happy shiba.jpg", "sassy shiba.jpg", "smiling shiba.jpg", and "toshi.jpg".

DogGallery > iPhone 6 Finished running DogGallery on iPhone 6

DogGallery
2 targets, iOS SDK 8.4
DogGallery
Shibas
cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg
AppDelegate.swift
ViewController.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
Info.plist
DogGalleryTests
Products

View Controller Scene
View Controller
View
ScrollView
Shiba Inus
toshi.jpg
First Responder
Exit
Storyboard Entry Point

Custom Class
Class UIScrollView
Module None

Identity
Restoration ID

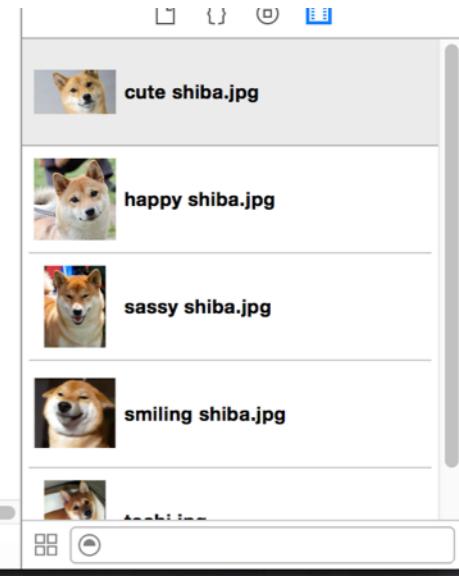
User Defined Runtime Attributes
keyPath Boolean

Document
Label Xcode Specific Label
Object ID ENU-OC-BzI
Lock Inherited - (Nothing)
Notes
No Font

Under "User Defined Runtime Attributes," click the "+" icon.

The screenshot shows the Xcode interface with the storyboard open. On the left is the file navigator, and the center is the storyboard editor. A scroll view containing images of Shiba Inus is selected. The right side shows the attributes inspector for the scroll view. A red arrow points to the '+' icon in the 'User Defined Runtime Attributes' section, which is highlighted in blue. A callout bubble with red text provides instructions: 'Under "User Defined Runtime Attributes," click the "+" icon.'

We'll rarely use the User Defined Runtime Attributes feature, but it's good to know. It's responsible for enabling us to use IB to set parameters on a View that would normally be set in code.



The screenshot shows the Xcode interface with the project 'DogGallery' selected. The storyboard file 'Main.storyboard' is open, showing a single scene with a 'View Controller' at the top level. Inside the controller, there is a 'View' which contains a 'ScrollView'. Inside the scroll view, there are two image views: one labeled 'Shiba Inus' and another labeled 'toshi.jpg'. The scroll view is highlighted with a gray selection bar. In the bottom right corner of the scroll view, the text 'UIScrollView' is visible. On the far left, the project navigator lists files like 'AppDelegate.swift', 'ViewController.swift', and 'Main.storyboard'. The bottom right of the screen shows a list of images: 'cute shiba.jpg', 'happy shiba.jpg', 'sassy shiba.jpg', 'smiling shiba.jpg', and 'toshi.jpg'. The 'contentSize' attribute of the scroll view is currently selected in the Attributes Inspector, indicated by a blue highlight.

Scroll Views are actually a View within a View.

The outer frame clips the inner content, which may be larger than the frame.

So, we're going to give the Scroll View some information about how big the content is inside of it.

Custom Class
Class `UIScrollView`
Module `None`

Identity
Restoration ID

User Defined Runtime Attributes
Key Path `contentSize` Type `Boolean`

Document
Label Xcode Specific Label
Object ID ENU-OC-BzI

Double-click on the “keyPath” value and change it to “contentSize.” Note the capitalization. Press Return.

cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg

Custom Class
Class `UIScrollView`
Module `None`

Identity
Restoration ID

User Defined Runtime Attributes

| Key Path | Type | Value |
|-------------|------------------|------------------|
| contentSize | Boolean | Number |
| | String | String |
| | Localized String | Localized String |
| | Point | Point |
| | Size | Rect |
| | Range | Range |
| | Color | Color |
| | Image | Image |
| | Nil | Nil |

Document

Under "Type," select "Size."

Notes No Font

File { } @

cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg

Main.storyboard

DogGallery

DogGallery Tests

Products

DogGallery

DogGallery

Main.storyboard

Main.storyboard (Base)

View Controller Scene

View Controller

View

ScrollView

Shiba Inus

toshi.jpg

First Responder

Exit

Storyboard Entry Point

Shiba Inus

cute shiba.jpg

happy shiba.jpg

sassy shiba.jpg

smiling shiba.jpg

toshi.jpg

DogGallery > iPhone 6 Finished running DogGallery on iPhone 6

DogGallery
2 targets, iOS SDK 8.4
DogGallery
Shibas
cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg
AppDelegate.swift
ViewController.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
Info.plist
DogGalleryTests
Products

View Controller Scene
View Controller
View
ScrollView
Shiba Inus
toshi.jpg
First Responder
Exit
Storyboard Entry Point

Custom Class
Class UIScrollView
Module None

Identity
Restoration ID

User Defined Runtime Attributes
contentSize Size {320, 1000}

Document
Label Xcode Specific Label
Object ID ENU-OC-BzI

UIScrollView

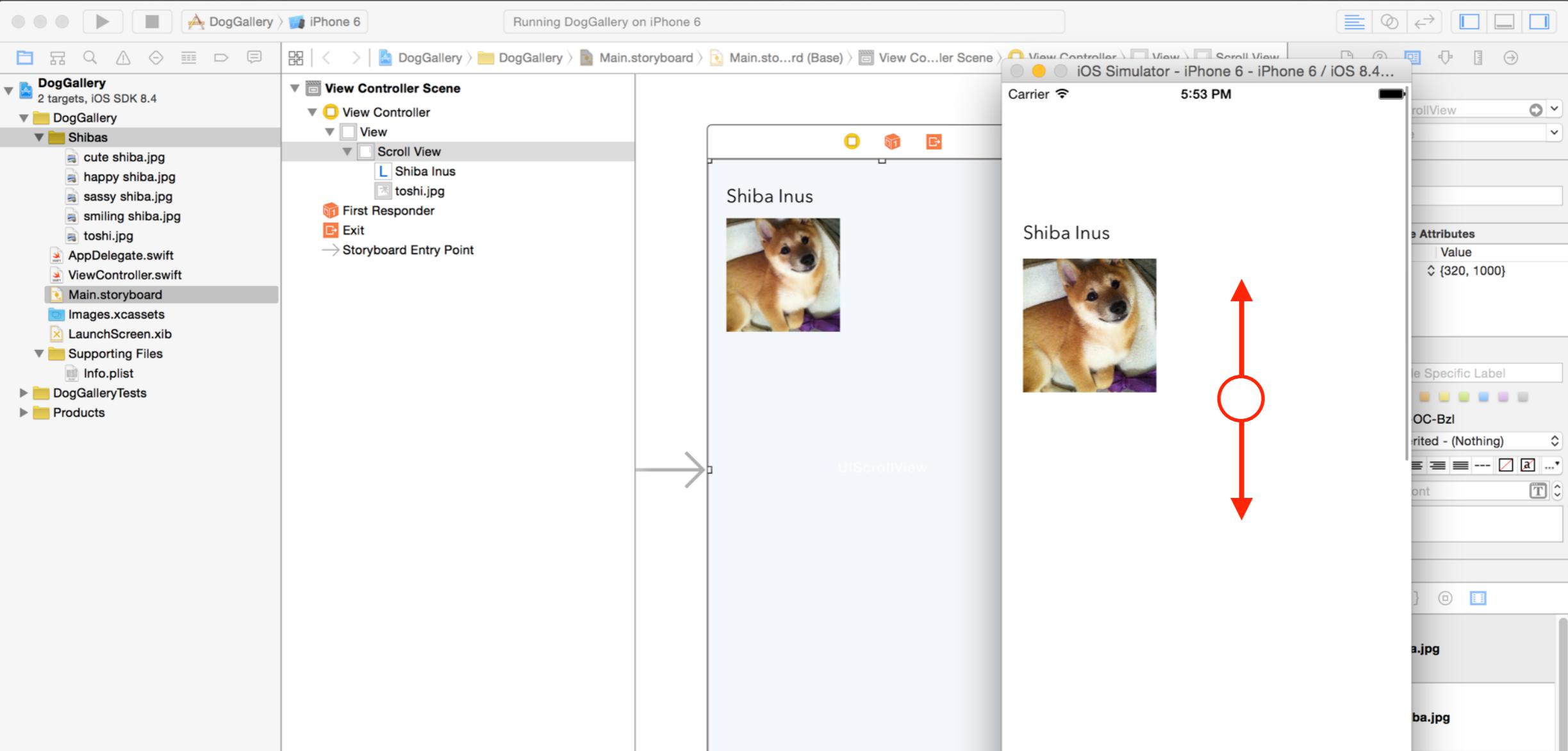
Shiba Inus

Change the value to {320, 1000}.
This is the width and height in points
respectively.

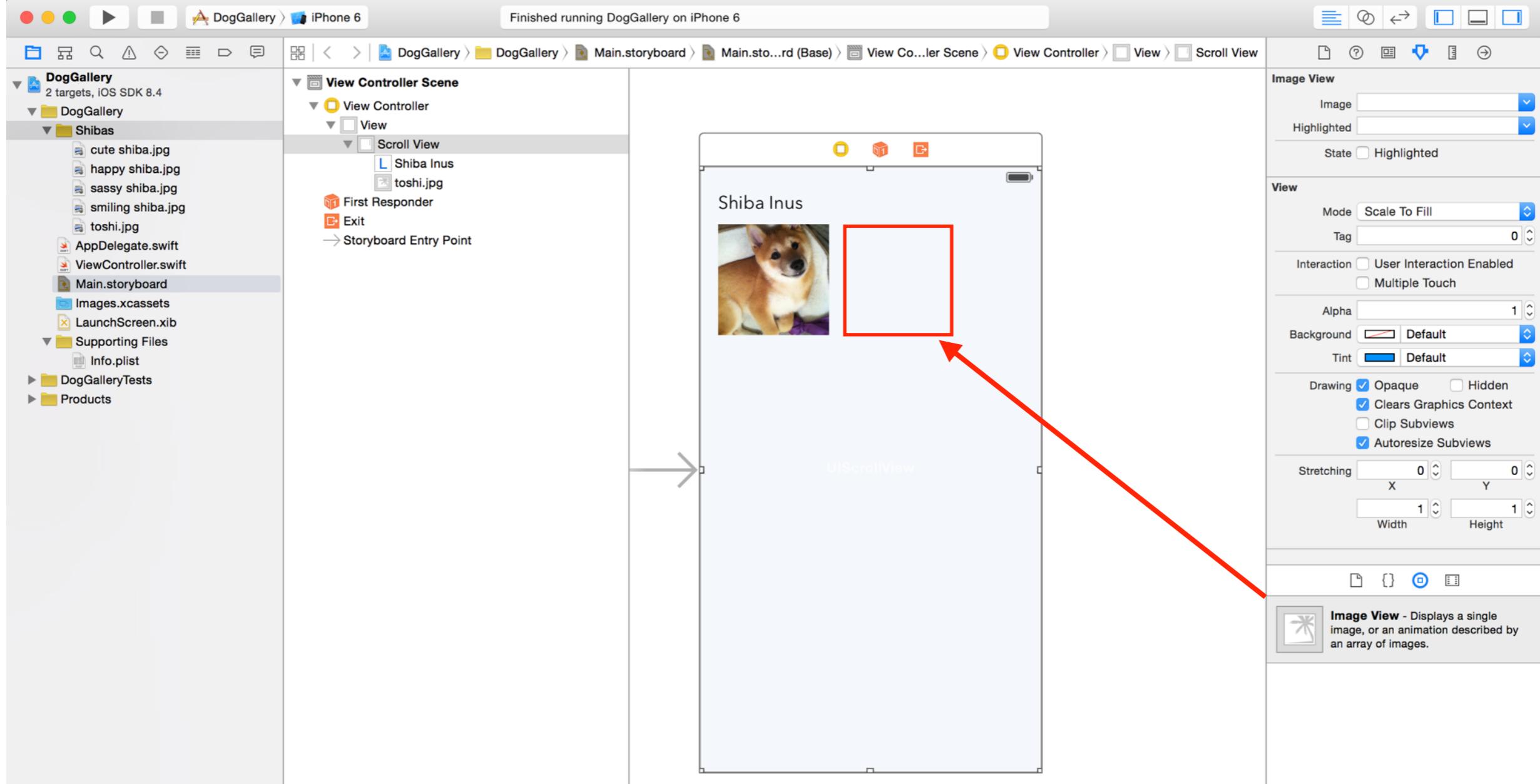
Since the Scroll View is only
320pt wide x 568pt tall, it will have to
enable scrolling vertically to show
the remaining content.

smiling shiba.jpg

smiling shiba.jpg



Running the app again (CMD + r or the Play button) should enable you to scroll the view up/down by click-dragging the view. Remember, the iOS simulator simulates an actual iPhone.



Continue to add Image Views using the procedure outlined before.
Add the rest of your images.

The screenshot shows the Xcode interface with the storyboard editor open. The left sidebar displays the project structure for 'DogGallery' with targets 'DogGallery' and 'Shibas'. The 'Main.storyboard' file is selected. The storyboard preview shows a screen titled 'Shiba Inus' displaying four Shiba Inu images in a grid. The document outline on the left shows the scene structure: 'View Controller Scene' containing 'View Controller', 'View', and a 'ScrollView' holding two images ('Shiba Inus' and 'toshi.jpg'). The right panel contains the 'Image View' settings for the selected image view, including options for mode (Scale To Fill), interaction (User Interaction Enabled), and drawing (Opaque). A note at the bottom right of the settings panel states: 'Image View - Displays a single image, or an animation described by an array of images.'

You can try a grid layout, bricks, horizontal bands, etc.

DogGallery > iPhone 6 Finished running DogGallery on iPhone 6

DogGallery
2 targets, iOS SDK 8.4
DogGallery
Shibas
cute shiba.jpg
happy shiba.jpg
sassy shiba.jpg
smiling shiba.jpg
toshi.jpg
AppDelegate.swift
ViewController.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
Info.plist
DogGalleryTests
Products

View Controller Scene
View Controller
View
ScrollView
Shiba Inus
toshi.jpg
First Responder
Exit
Storyboard Entry Point

View
Show Frame Rectangle
X: 8 Y: 54
Width: 104 Height: 104
Autoresizing
iOS 6/7 Deltas

To add Image Views below the scroll view's frame, you have to add one as usual, then adjust the X and Y values of the View's position manually.

Experiment with the different values here to move the views into place.

Soon, we'll be doing this with code, so this seemingly odd procedure won't apply, but it's good practice for laying out static content and understanding the principles of laying out Views parametrically.