Automated UI testing and automated feature testing of iPhone and iPad applications using the Calabash framework

UI Testing

with Calabash

Contact: Niklas Saers
+45 207 27 700
njs@trifork.com
@niklassaers
Skype: njstrifork

Quick Disclaimer

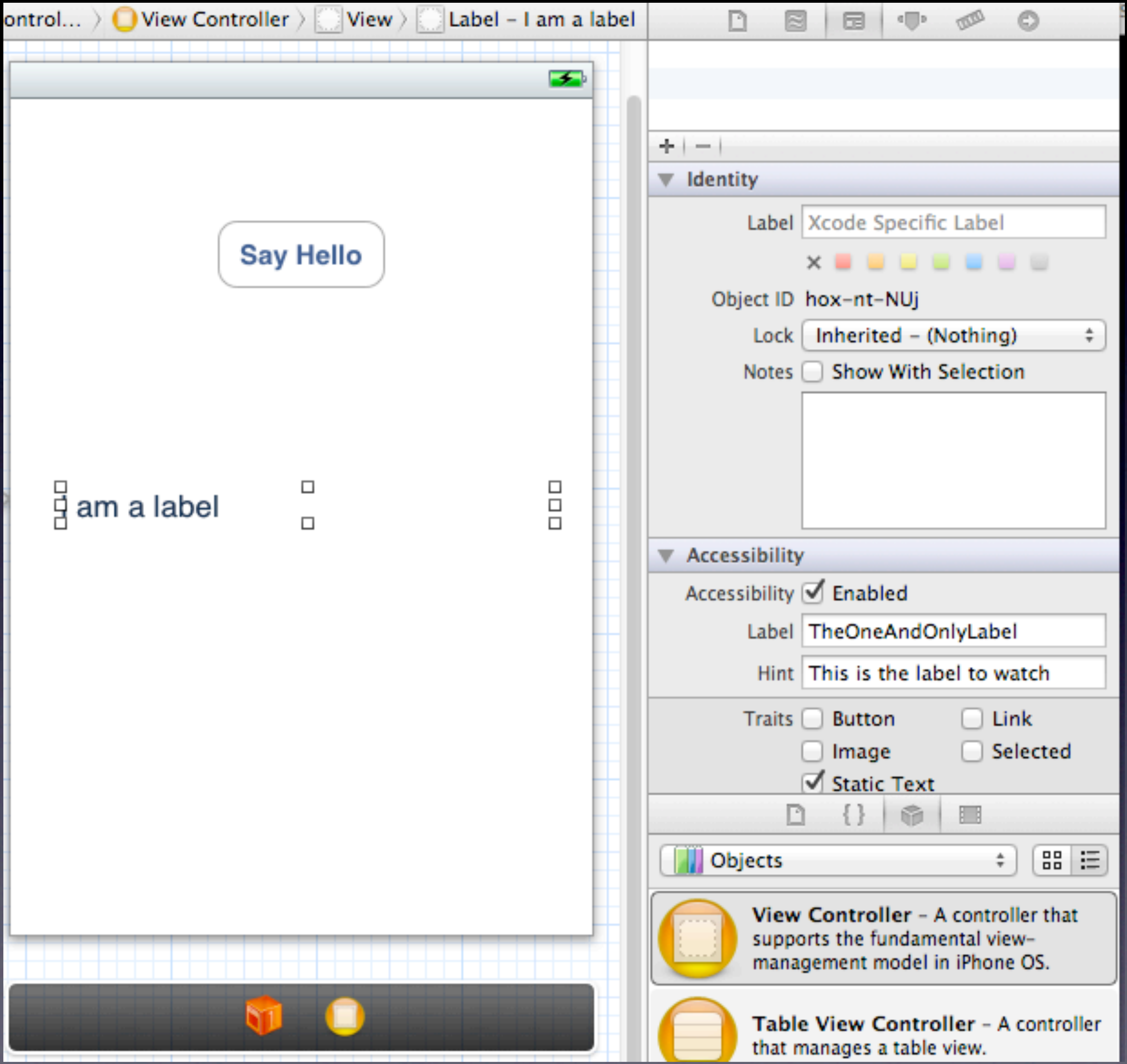
- We meet frequently
- The company I work for has some stock in Less Painful, and I have some stock in our company
- But really, I just think this is seriously cool, and you should use this!

Calabash-ios

- Open source project
- iOS framework
  - HTTP server that supports querying and touching views via the accessibility inspector
- Ruby HTTP client
- Cucumber/Gherkin features

- Calabash-android also available

Setting Accessibility label & hint



Identity

Label: Xcode Specific Label

Object ID: hox-nt-NUj

Lock: Inherited – (Nothing)

Notes: □ Show With Selection

Accessibility

Accessibility: ☑ Enabled

Label: TheOneAndOnlyLabel

Hint: This is the label to watch

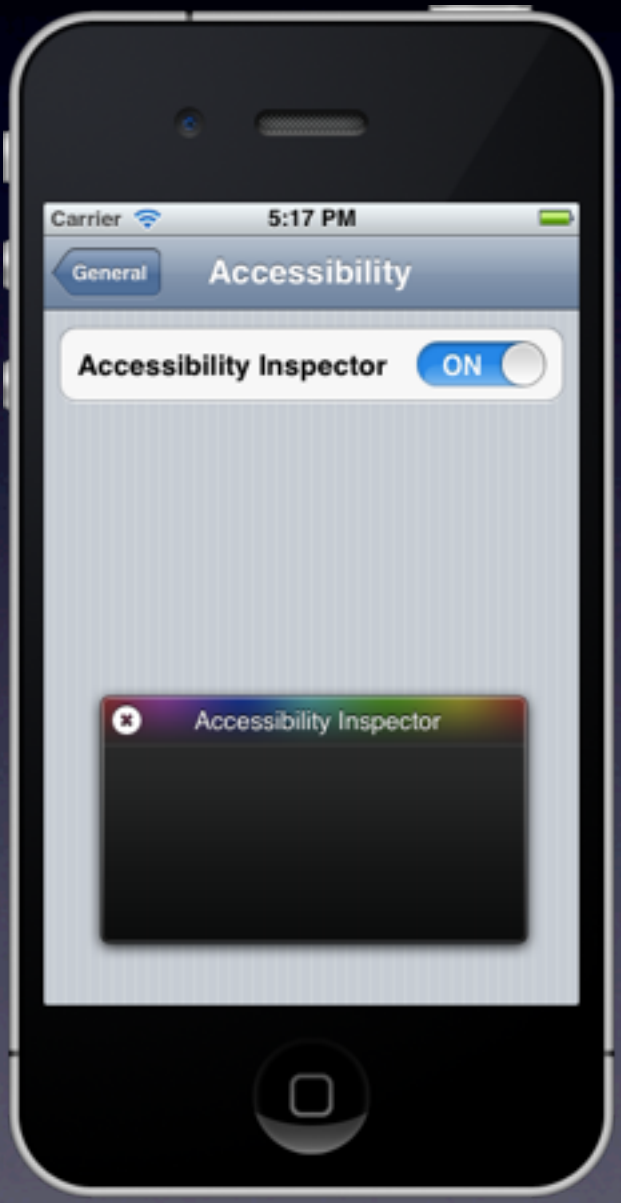
Traits: □ Button □ Link
□ Image □ Selected
☑ Static Text

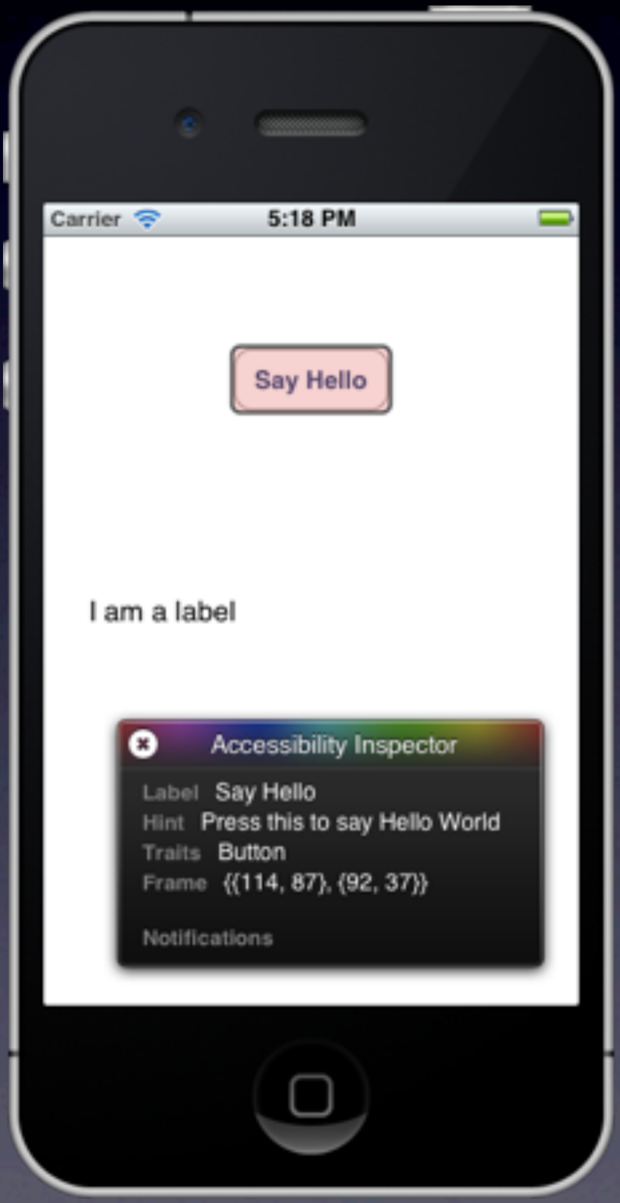
Objects

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

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

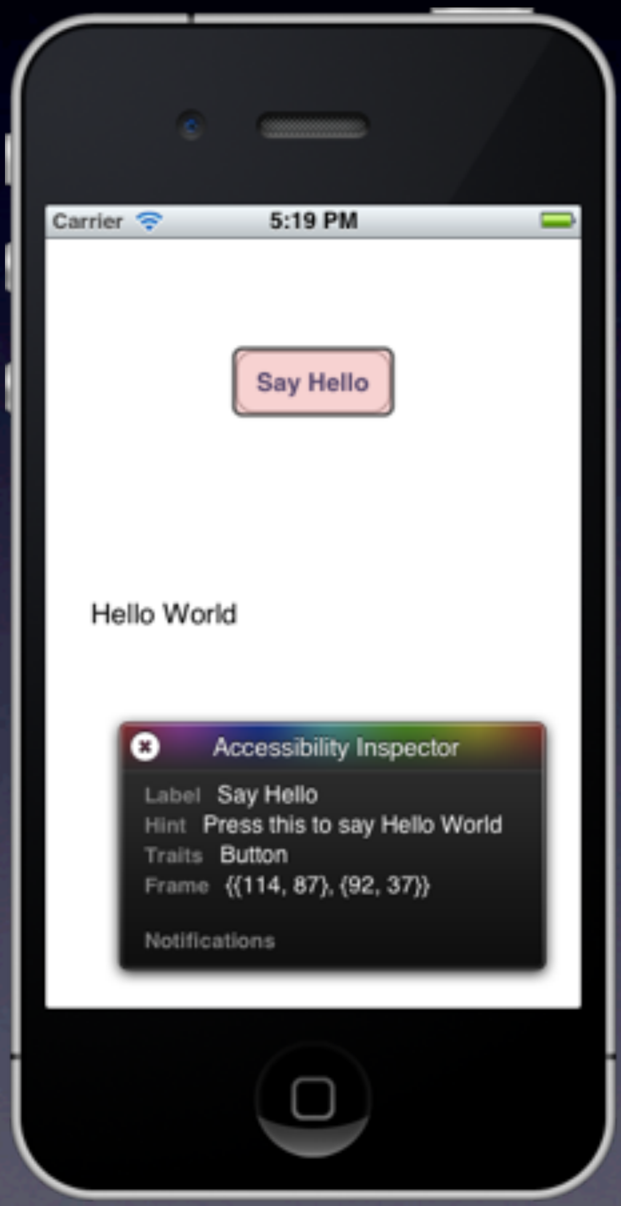
Our Hello World





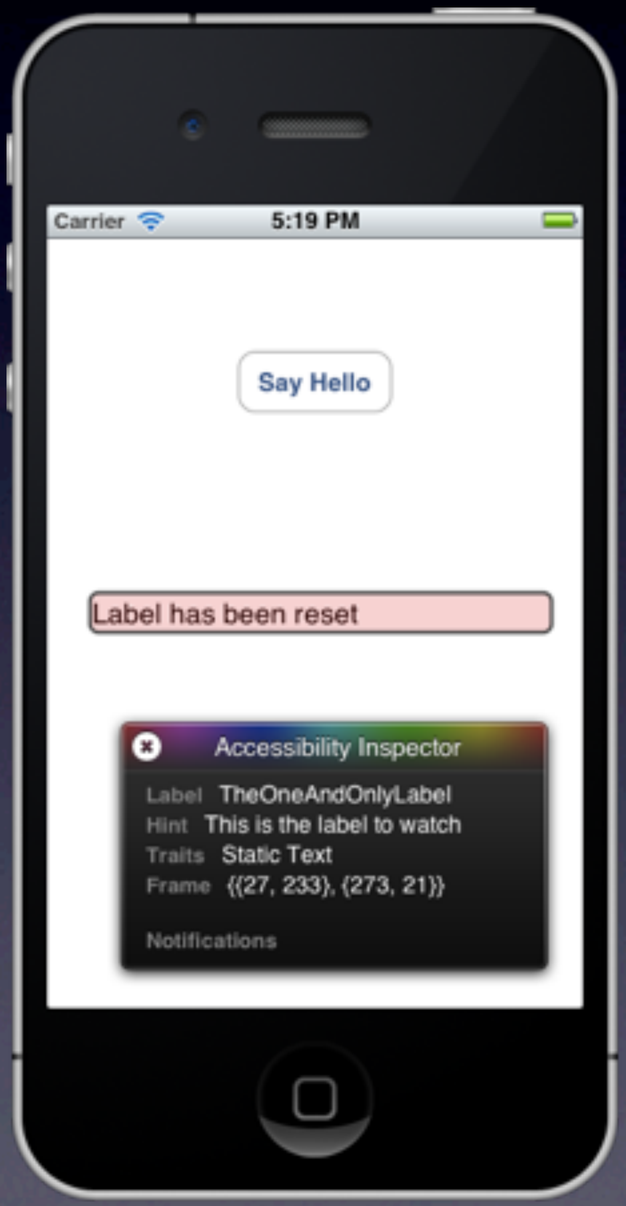
Label Say Hello
Hint Press this to say Hello World
Traits Button
Frame {{114, 87}, {92, 37}}

Notifications



Label Say Hello
Hint Press this to say Hello World
Traits Button
Frame {{114, 87}, {92, 37}}

Notifications



Label TheOneAndOnlyLabel
Hint This is the label to watch
Traits Static Text
Frame {{27, 233}, {273, 21}}

Notifications

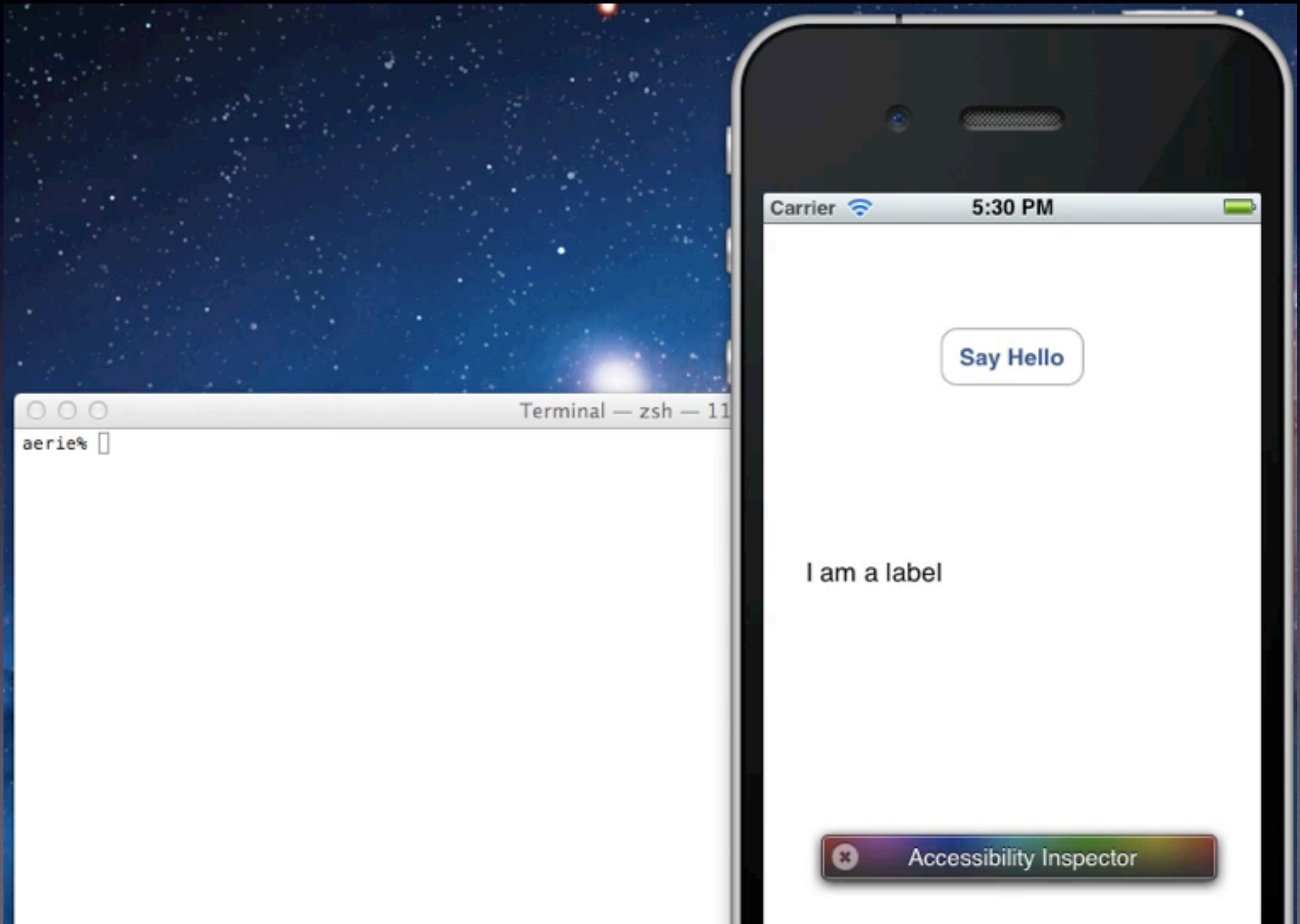
Writing our test

```
Feature: Button Pressing
  As a user
  I want to show hello world and make it go away
  So I can get the hang of these tests

Scenario: Say hello and remove it again
  Given the app is running
  And I touch the "Say Hello" button
  Then I see the text "Hello World"
  And I swipe right on "TheOneAndOnlyLabel"
  Then I don't see the text "Hello World"
```

calabash.sh: NO_LAUNCH=1 DEVICE=iphone OS=ios5 cucumber

Test results



```
aerie% 
```

Defining our own steps

- Lots of predefined steps
- Defined in Ruby
- Stubbed when running the test
- Use the interactive shell and copy-paste it into your step definitions

Defining our own steps

- Accessibility inspector
- Accessibility label & hint

- run irb_ios5.sh

  ```
  query("view")
  query("label marked:'TheOneAndOnlyLabel'")
  swipe "right", {:query => "label marked:'TheOneAndOnlyLabel'"}
  touch("button marked:'Say Hello'")
  ```
- record_begin / record_end “myrecording” / playback “myrecording”

  ```
  interpolate "myrecording", :start => "imageView index:2", :end =>
  "imageView index:0"=> ["> - YellowSquare.png"]
  ```

- Inspiration: calabash_steps.rb
- Is the label A to the right of button B?

Examples

```
Given /^I playback recording "([^"]*)" at label "([^"]*)"$/ do |gesture, label|
  playback gesture, {:query => "label text:'#{label}'"}
end

Given /^I remove all my documents$/ do
  docs = query("view:'PdfThumbnailView'")
  docs.each do |pdfView|
    touch("button marked:'EditCards'")
    touch("view:'PdfThumbnailView' index:0 button marked:'card function delete'")
    touch("button marked:'Delete file'")
    touch("button marked:'EditCards'")
  end
end
```

How do I get it?

- cd ~/projects/myProject
- sudo gem install calabash-cucumber
- calabash-ios setup
- calabash-ios gen

- You’re ready to write your feature tests

- DEVICE=iphone OS=ios5 cucumber

```
git clone git://github.com/calabash/calabash-ios.git
git clone git://github.com/calabash/calabash-ios-server.git
```

Quick review

- Easy to install in your XCode project
- Quick to write natural language tests
- Easy to define custom steps
- Easy to run your tests over and over again
- Takes screenshots when they fail
- Easy to integrate with your build setup

- Quick plug: Less Painful has a service to test your build on all iOS devices (including robots to rotate them)

The enterprise case

- Users, product owners, testers, they all write tests
- In a natural language

```
Given I’m logged in correctly
And I write a tweet with the text “1234”
And I look in my timeline
Then I see my tweet with text “1234”
```

The indie case

- Quickly write tests
- Write easy stubs for what to do
- Make steps only for what is new

Quick heads-up

- Don’t ship your app including calabash, it uses private APIs! (That’s why it made a new scheme for you)
- This code is available at my blog starting now
- All my reusable steps will be posted by this weekend
- Ask me any Calabash question on Twitter (this month) @niklassaers

Questions?

- No time for questions, but ask me for a demo on my production code to see it in action