

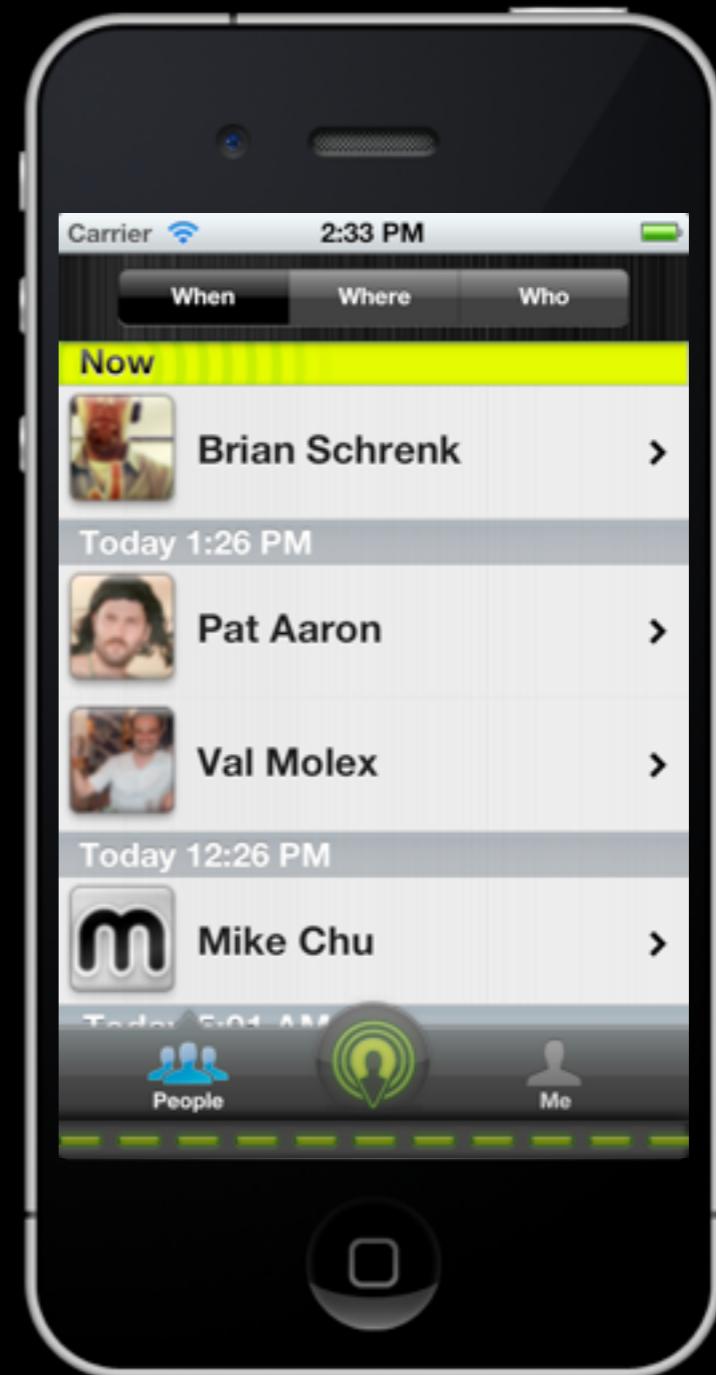
What's New in UIAutomation?

Alex Vollmer
@alexvollmer



Meechu

<http://meechuapp.com>



PREVIOUSLY ON
UIAUTOMATION...

Quick Review

- UIAutomation is part of Instruments
- Tests are written in JavaScript
- Runs on the device and simulator

What Have We Accomplished?

What Have We Accomplished?

High-level language

SCM-compatible

Runs on device & simulator

Rich API we can extend & enhance

Ability to be completely automated

Test your app in various states

What Have We Accomplished?

- High-level language
- SCM-compatible
- Runs on device & simulator
- Rich API we can extend & enhance
- Ability to be completely automated
- Test your app in various states

What Have We Accomplished?

- High-level language
 - SCM-compatible
- Runs on device & simulator
- Rich API we can extend & enhance
- Ability to be completely automated
- Test your app in various states

What Have We Accomplished?

- High-level language
 - SCM-compatible
 - Runs on device & simulator
- Rich API we can extend & enhance
- Ability to be completely automated
- Test your app in various states

What Have We Accomplished?

- High-level language
 - SCM-compatible
 - Runs on device & simulator
 - Rich API we can extend & enhance
- Ability to be completely automated
- Test your app in various states

What Have We Accomplished?

- High-level language
 - SCM-compatible
 - Runs on device & simulator
 - Rich API we can extend & enhance
- ~~Ability to be completely automated~~

Test your app in various states

What Have We Accomplished?

- High-level language
 - SCM-compatible
 - Runs on device & simulator
 - Rich API we can extend & enhance
- ~~Ability to be completely automated~~
- ~~Test your app in various states~~

UIAHost

UIAHost

```
var target = UITarget.localTarget();
var host = target.host();
var result = host.performTaskWithPathArgumentsTimeout("/usr/bin/cal",
  ["2012"], 5)
```

Cool UIAHost Tricks

Cool UIAHost Tricks

- Extend test scripts in other languages

Cool UIAHost Tricks

- Extend test scripts in other languages
- Configure your tests dynamically

Cool UIAHost Tricks

- Extend test scripts in other languages
- Configure your tests dynamically
- Externalize your validation (more on this in a bit...)

Script Recording

Xcode File Edit View Navigate Editor Product Window Help iShowU HD 00:00:00 15:39:06

Meechu.xcodeproj — NewStreetAddressViewController.m

Build Meechu: Succeeded | Today at 3:36 PM

Project 1 4

Run Stop Scheme Breakpoints Editor View Organizer

Build Issues Debug NewStreetAddressViewController.m

Meechu > Classes > Controllers > Address Book > NewStreetAddressViewController.m -saveAccountWithAddressDictionary:type:

```
// Releases the view if it doesn't have a superview.
[super didReceiveMemoryWarning];

// Release any cached data, images, etc that aren't in use.
}

#pragma mark - Saving

- (void)saveAccountWithAddressDictionary:(NSDictionary *)address type:(NSString *)addressType
{
    NSString *street = [address objectForKey:(NSString *)kABPersonAddressStreetKey];
    NSString *city = [address objectForKey:(NSString *)kABPersonAddressCityKey];
    NSString *state = [address objectForKey:(NSString *)kABPersonAddressStateKey];
    NSString *zip = [address objectForKey:(NSString *)kABPersonAddressZIPKey];
    NSString *country = [address objectForKey:(NSString *)kABPersonAddressCountryKey];
    NSString *countryCode = [address objectForKey:(NSString *)kABPersonAddressCountryCodeKey];

    NSError *error;
    NSData *addressJSON = [NSJSONSerialization dataWithJSONObject:address options:0 error:&error];
    if (addressJSON) {
        NSString *JSON = [[[NSString alloc] initWithData:addressJSON encoding:NSUTF8StringEncoding] autorelease];

        NSManagedObjectContext *moc = [self createManagedObjectContext];
        if ([MUAccount accountExistsWithIdentifier:JSON type:MUConnectionTypeStreetAddress inContext:moc]) {
            NSString *message = NSLocalizedString(@"You are already sharing this street address",
                                                @"You are already sharing this street address");

            UIAlertView *alert = [[[UIAlertView alloc] initWithTitle:nil
                                                       message:message
                                                       delegate:nil
                                                       cancelButtonTitle:NSLocalizedString(@"OK", @"OK Button")
                                                       otherButtonTitles:nil] autorelease];

            [alert show];
        }
        else {
            MUAccount *account = [MUAccount insertInManagedObjectContext:moc];
            account.connectionType = MUConnectionTypeStreetAddress;
            account.identifier = JSON;
        }
    }
}
```

Command-line Invocation

Command-line Invocation

```
/usr/bin/instruments \  
-t /Developer/Platforms/iPhoneOS.platform/Developer/Library/  
Instruments/PlugIns/AutomationInstrument.bundle/Contents/  
Resources/Automation.tracetemplate \  
Twitter.app \  
-e UIASCRIPT tests/integration/twitter-test.js \  
-e UIARESULTSPATH .
```

Command-line Invocation

```
/usr/bin/instruments \  
-t /Developer/Platforms/iPhoneOS.platform/Developer/Library/  
Instruments/PlugIns/AutomationInstrument.bundle/Contents/  
Resources/Automation.traceTemplate \  
Twitter.app \  
-e UIASCRIPT tests/integration/twitter-test.js \  
-e UIARESULTSPATH .
```

Command-line Invocation

Command-line Invocation

It's Unix! I can pipe stuff together! Fabulous!

Command-line Invocation

It's Unix! I can pipe stuff together! Fabulous!

There's just one problem...



Wednesday, April 25, 12



Mr. Tupolev, one
exit code only, if you
please

<http://gist.github.com/2363450>

Continuous Integration

The Holy Grail



Jenkins

Jenkins

Jenkins

- Check code out from favorite SCM

Jenkins

- Check code out from favorite SCM
- Run Xcode build

Jenkins

- Check code out from favorite SCM
- Run Xcode build
 - Unit-tests too!

Jenkins

- Check code out from favorite SCM
- Run Xcode build
 - Unit-tests too!
- Run UIAutomation from command line

Jenkins

- Check code out from favorite SCM
- Run Xcode build
 - Unit-tests too!
- Run UIAutomation from command line
- Notifications for build failures and recovery

tuneup_js

tuneup_js

If I don't pimp it, no one else will

tuneup_js

If I don't pimp it, no one else will

http://github.com/alexvollmer/tuneup_js

Test Structure & Assertions

Test Structure & Assertions

```
test("Main Screen", function(target, app) {  
    // test body goes here  
});
```

Test Structure & Assertions

```
test("Main Screen", function(target, app) {  
    // test body goes here  
});
```

```
fail("You suck");  
assertTrue(foo == "bar");  
assertTrue(foo == "bar",  
        "Hey! foo isn't 'bar'!");
```

More Assertions

```
assertFalse(23 == total);
```

```
assertMatch(/.* goes here/, title);
```

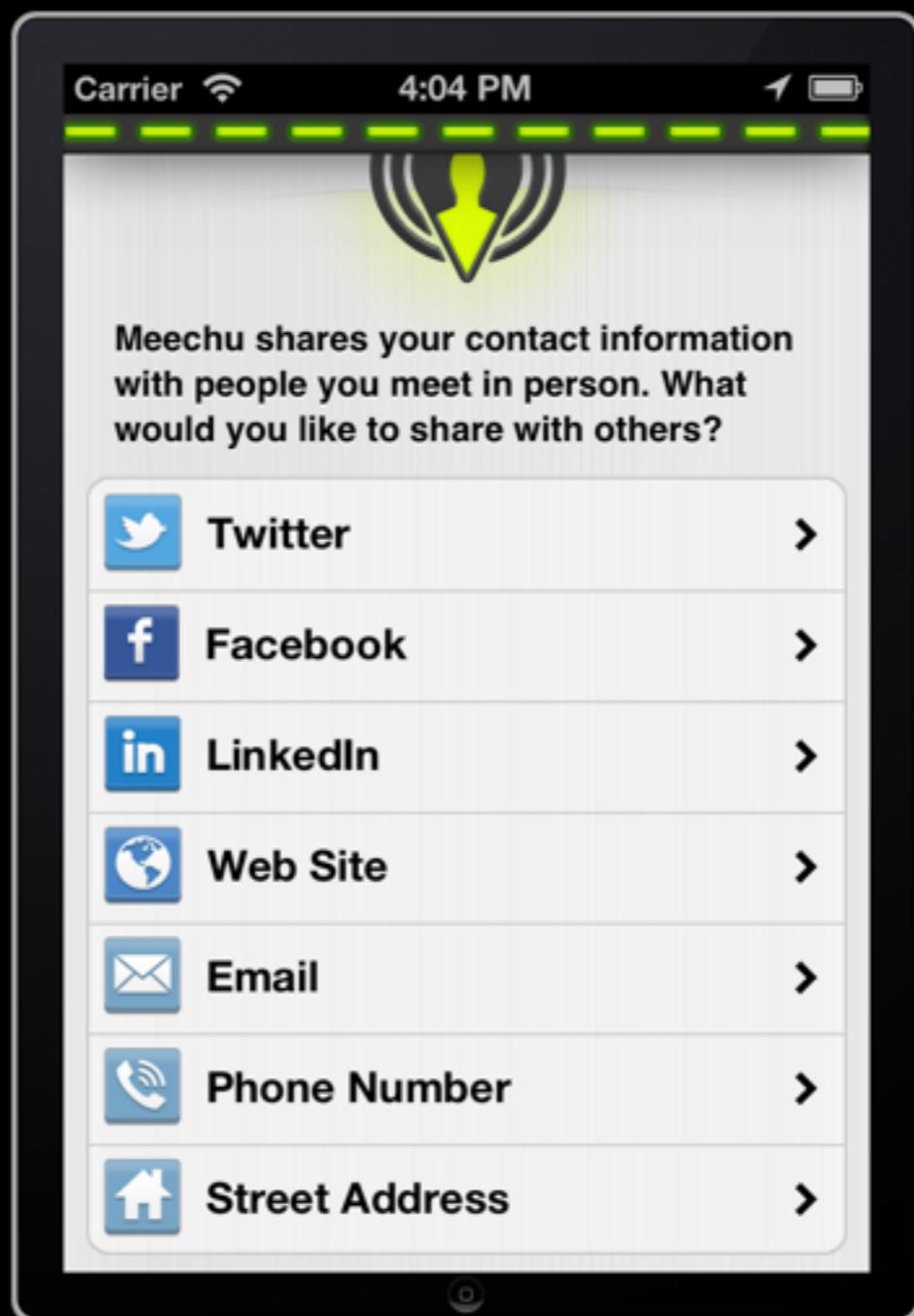
```
assertEquals(22, total);
```

```
assertNull(errorBox);
```

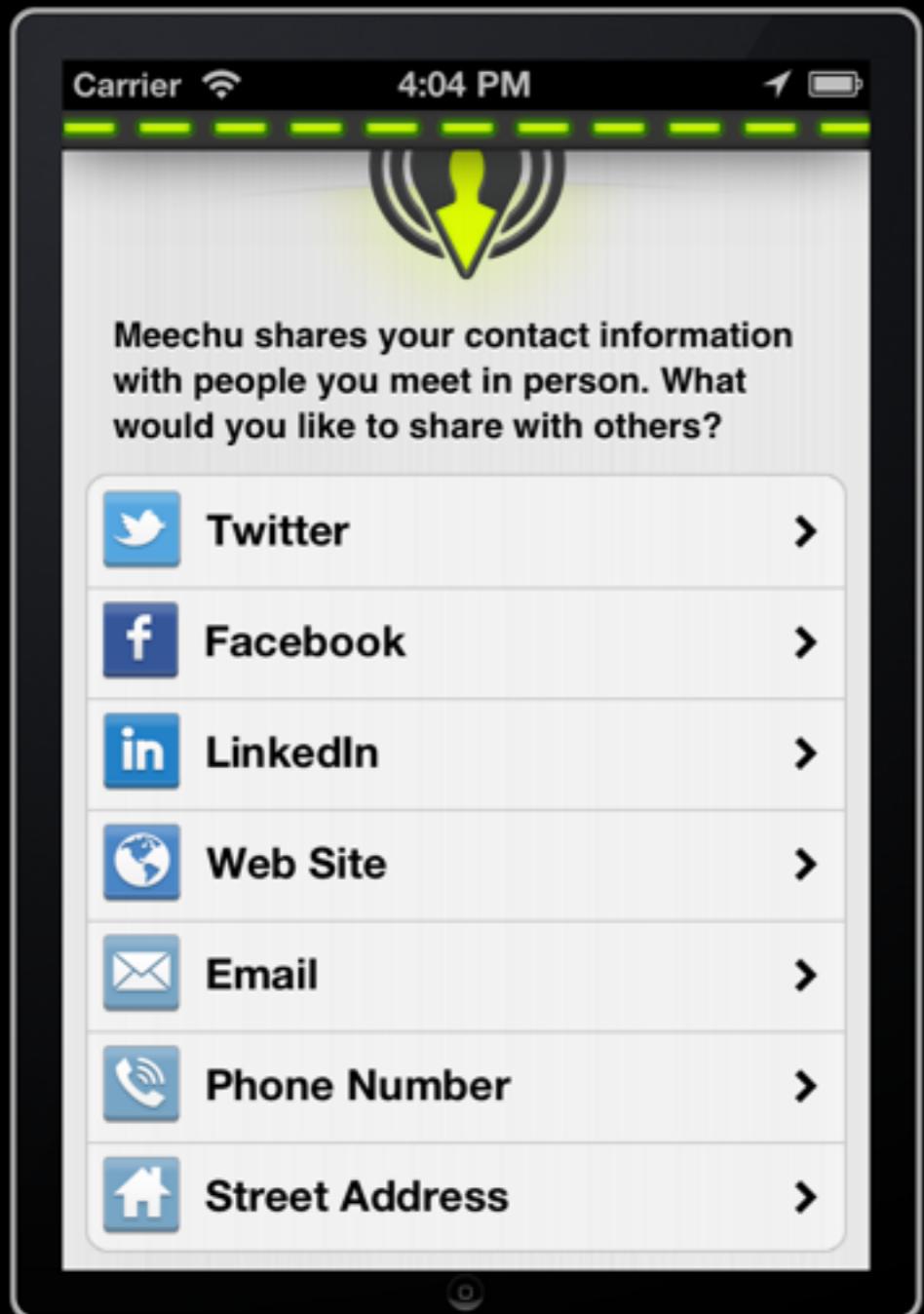
```
assertNotNull(goldenTicket);
```

Screen Assertions

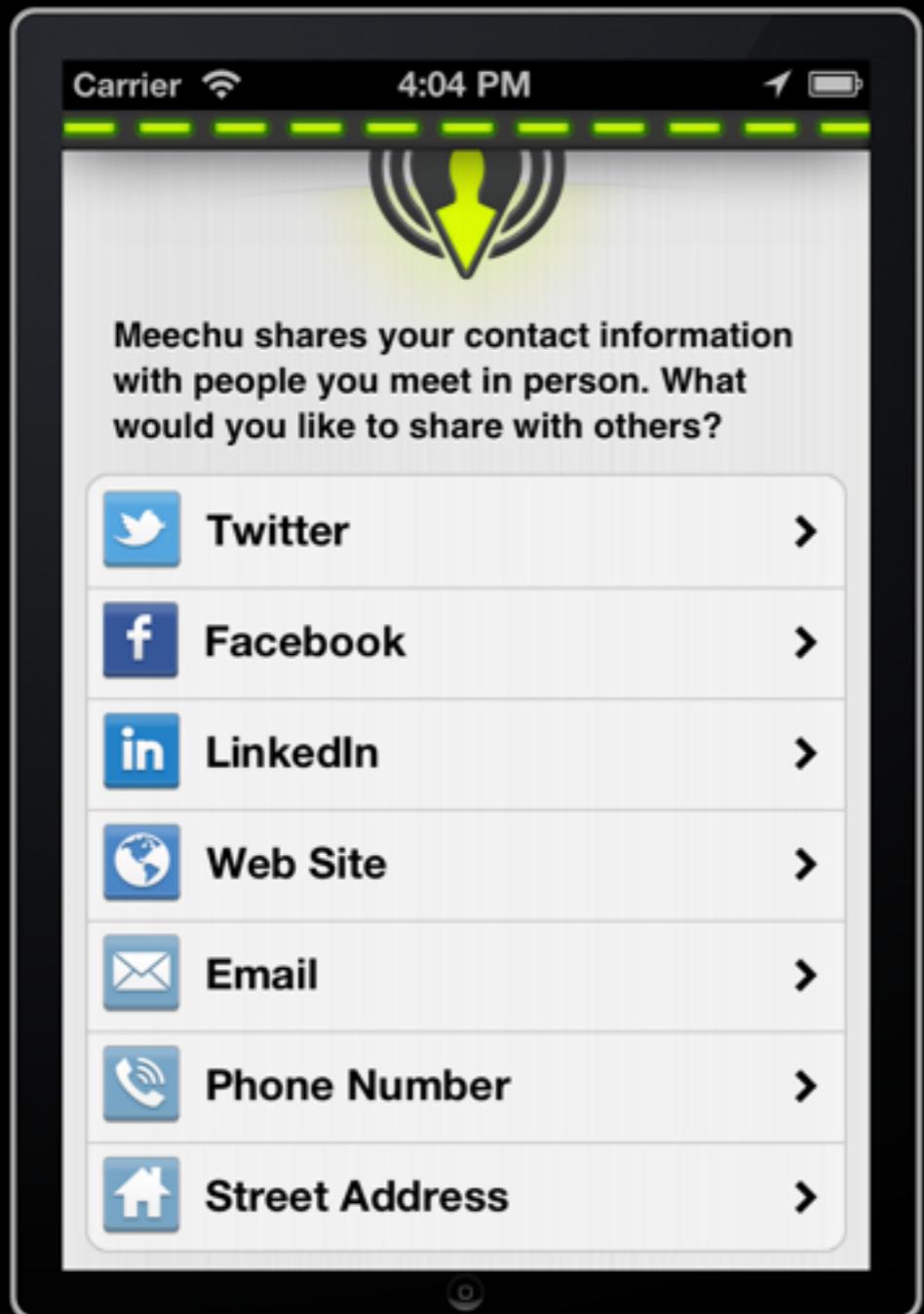
Screen Assertions



Screen Assertions

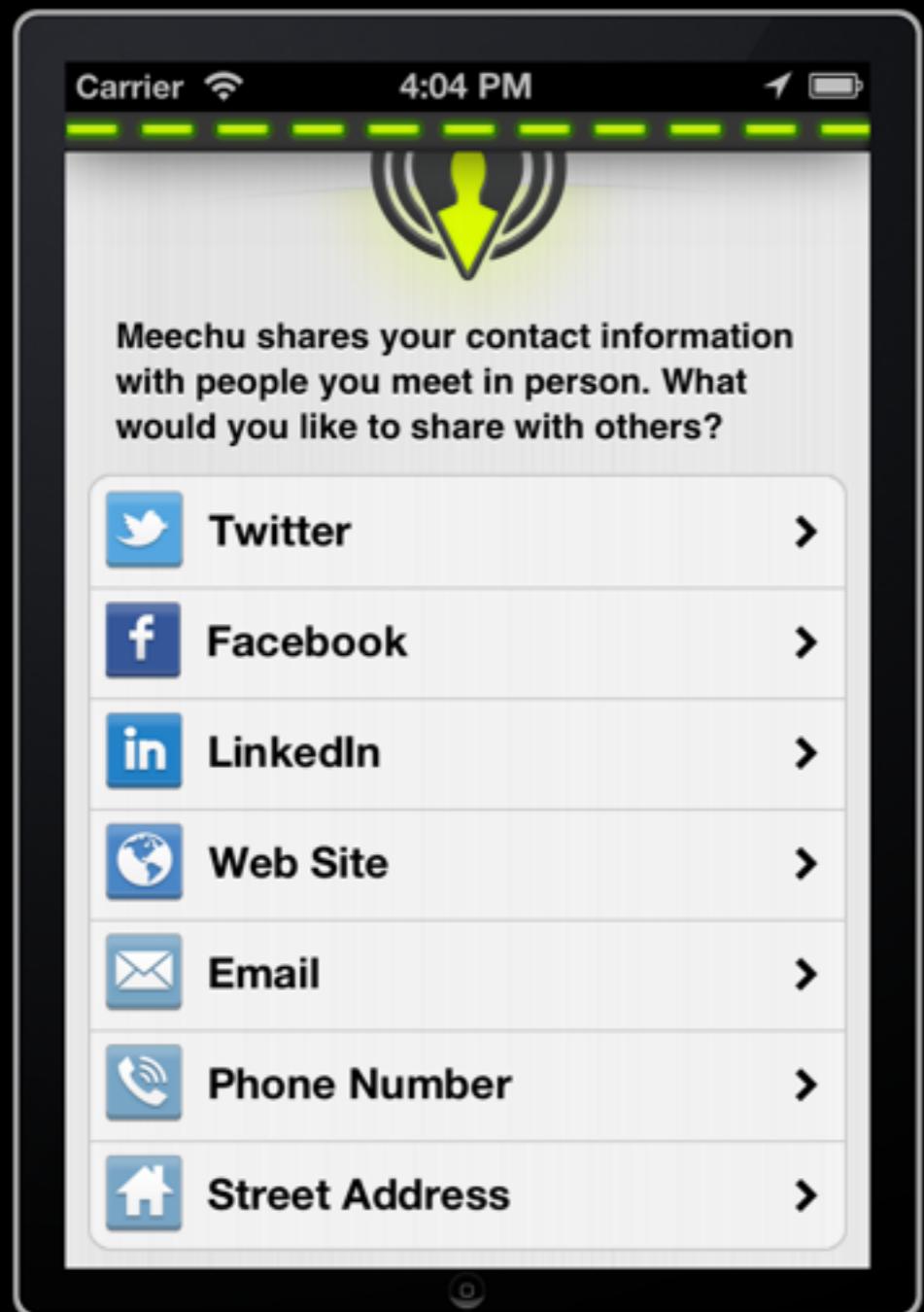


Screen Assertions



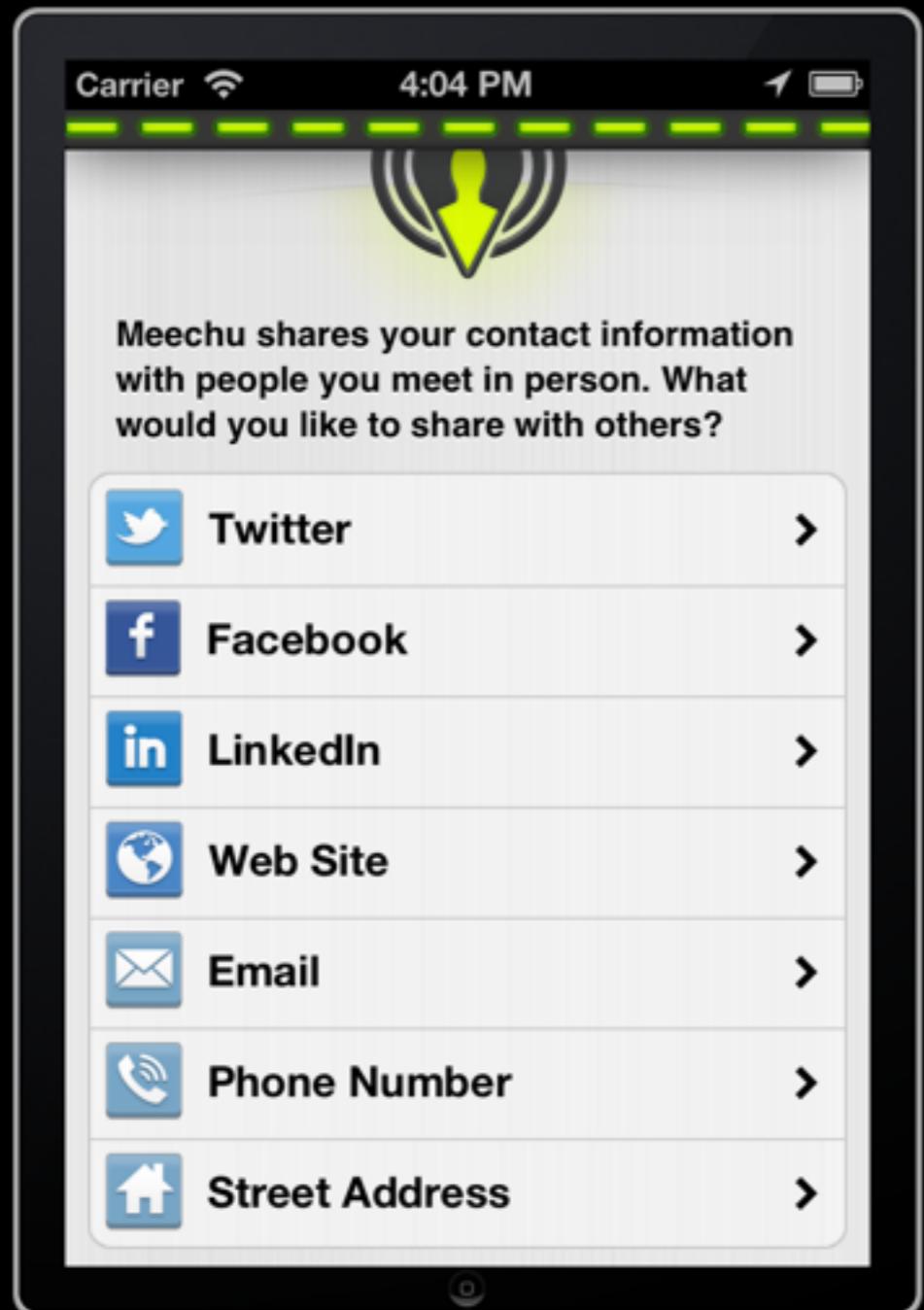
- Match the header label

Screen Assertions



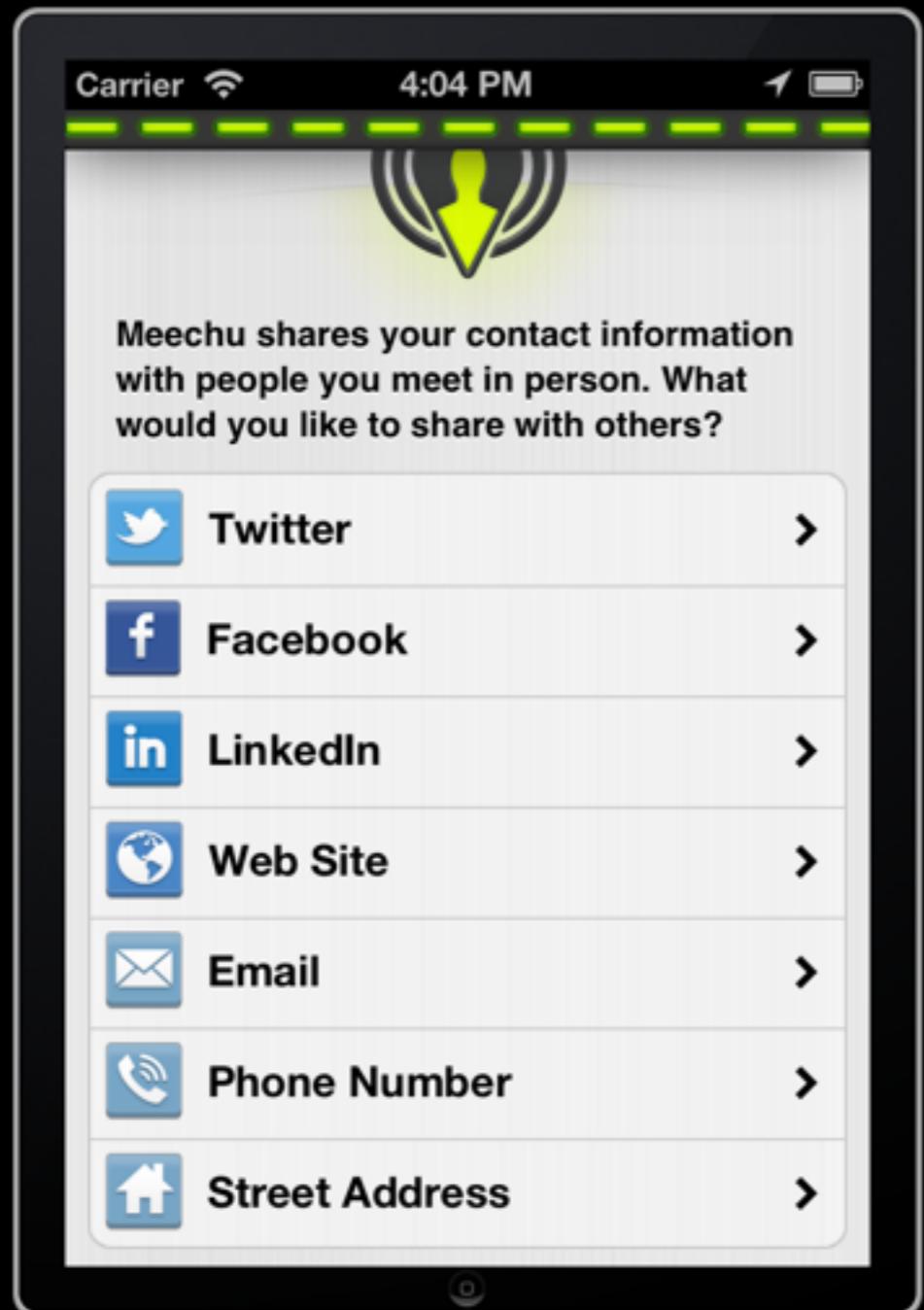
- Match the header label
- There should be seven rows

Screen Assertions



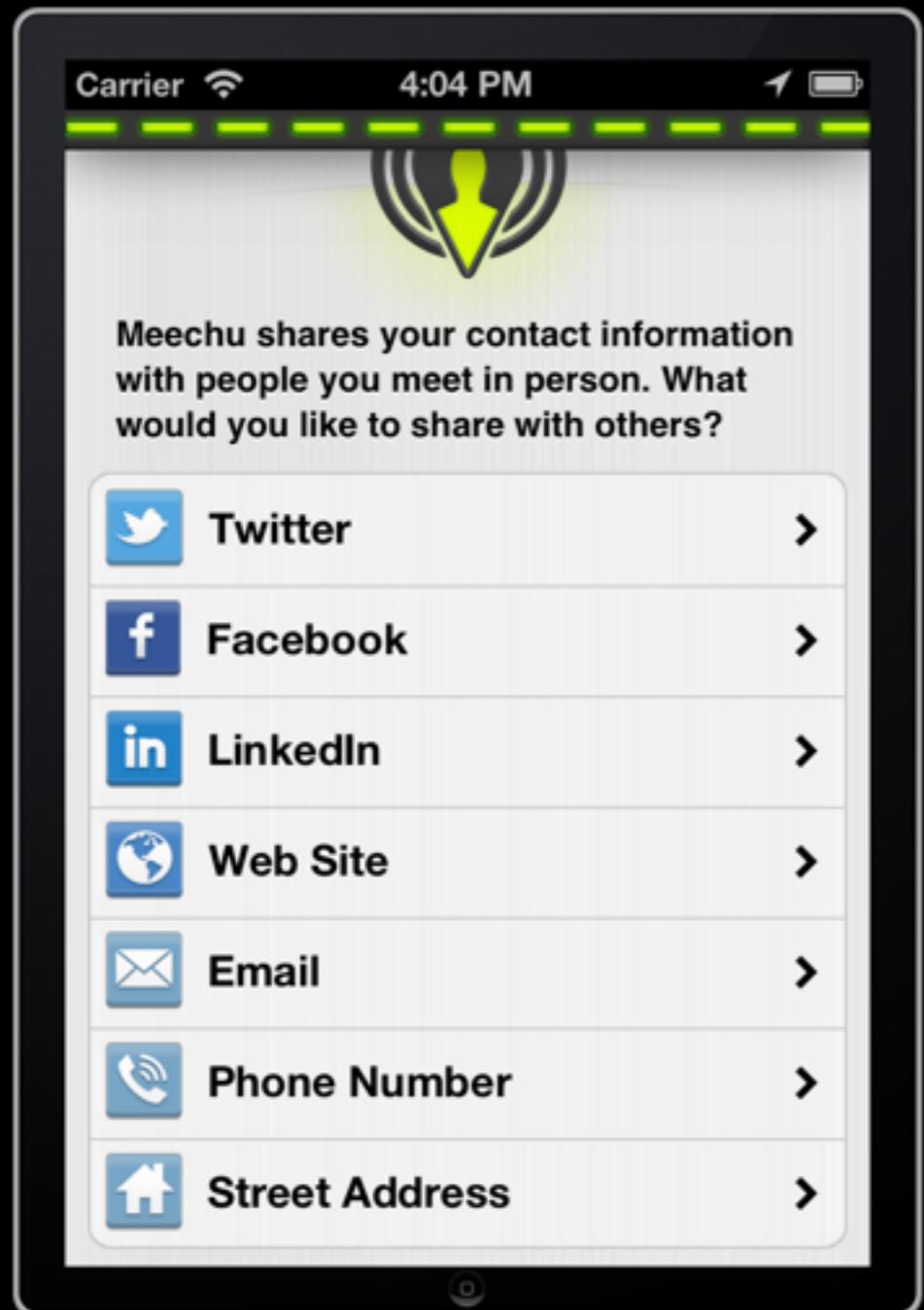
- Match the header label
- There should be seven rows
- The order of the rows should be:

Screen Assertions



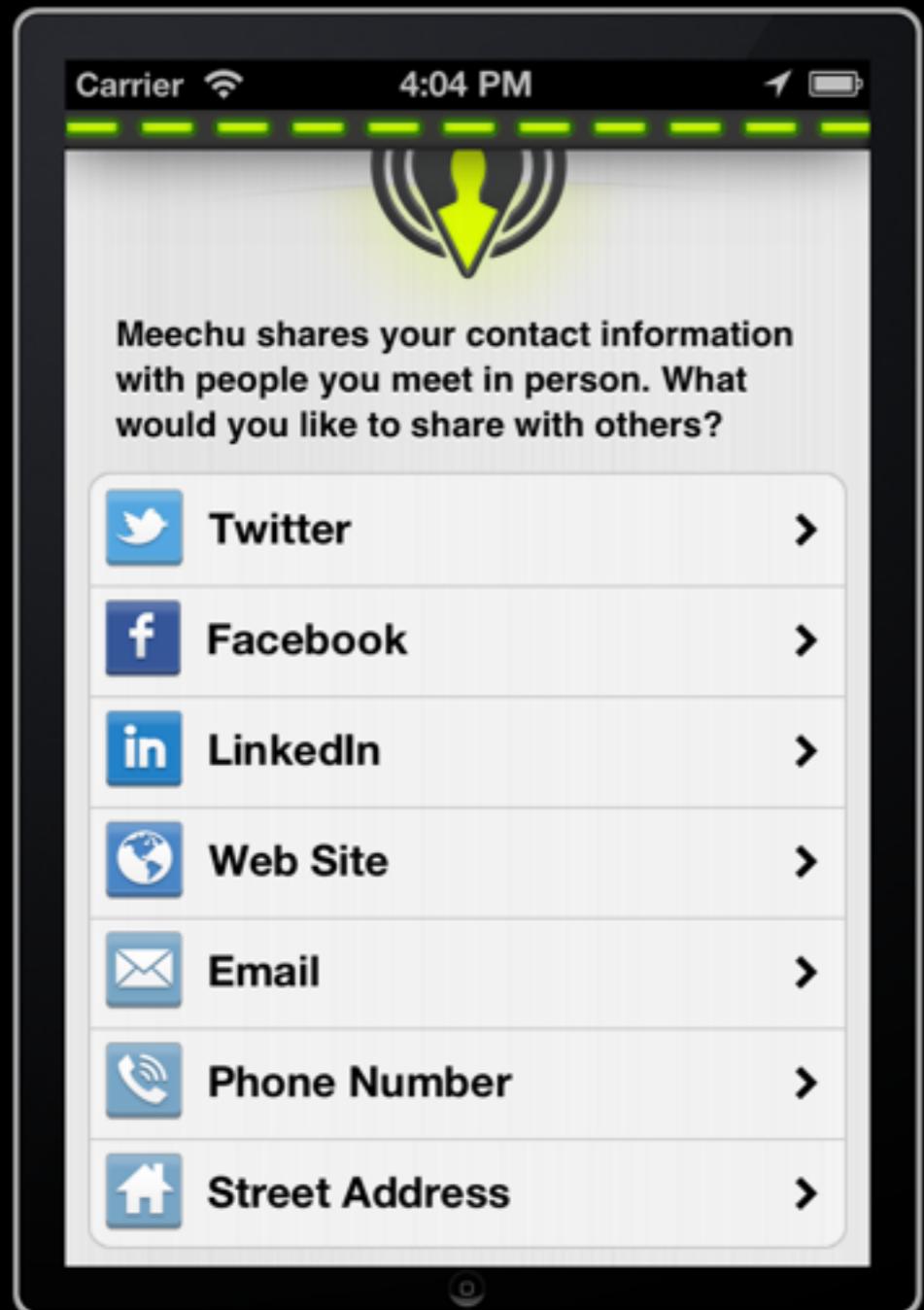
- Match the header label
- There should be seven rows
- The order of the rows should be:
 - Twitter

Screen Assertions



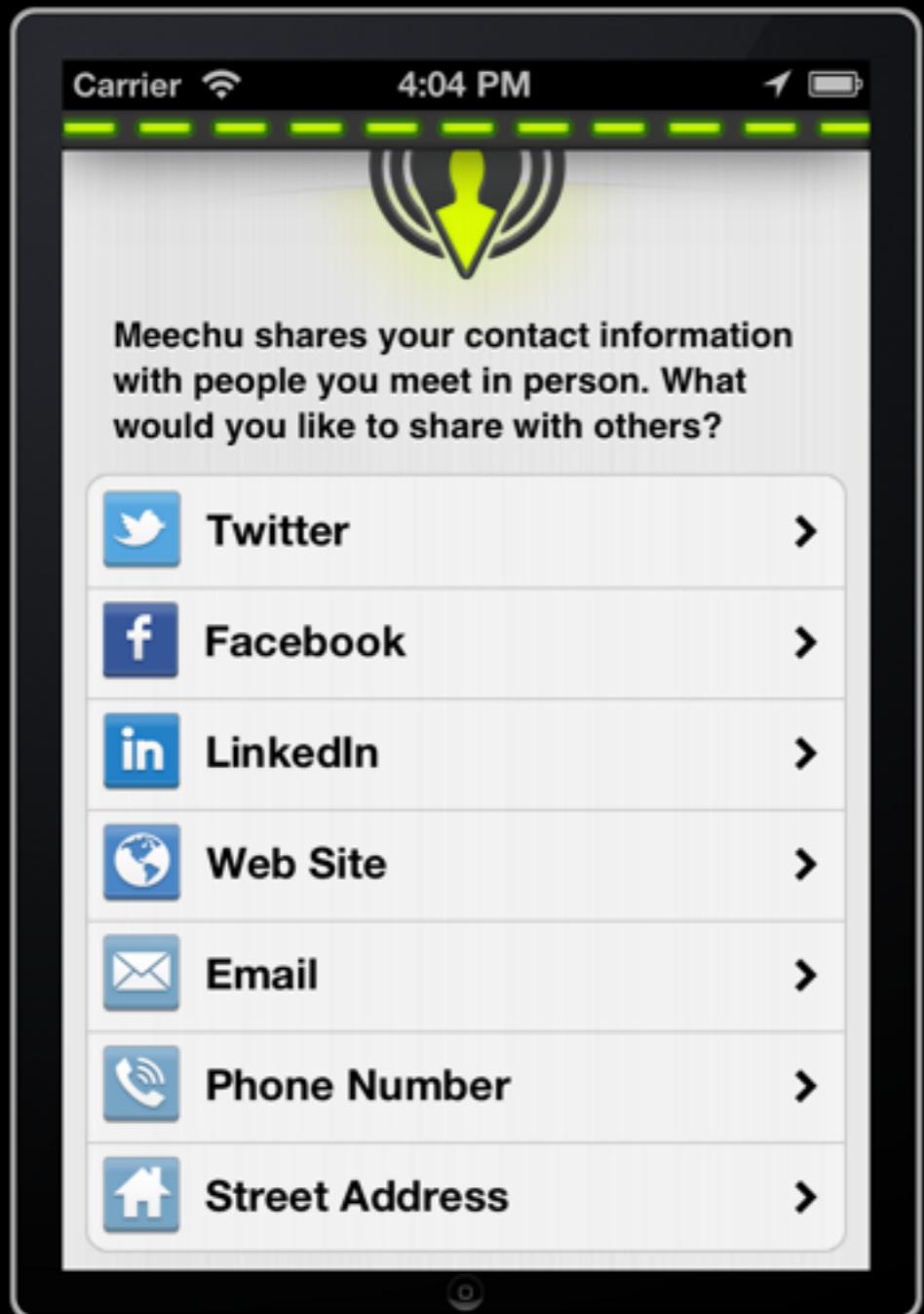
- Match the header label
- There should be seven rows
- The order of the rows should be:
 - Twitter
 - Facebook

Screen Assertions



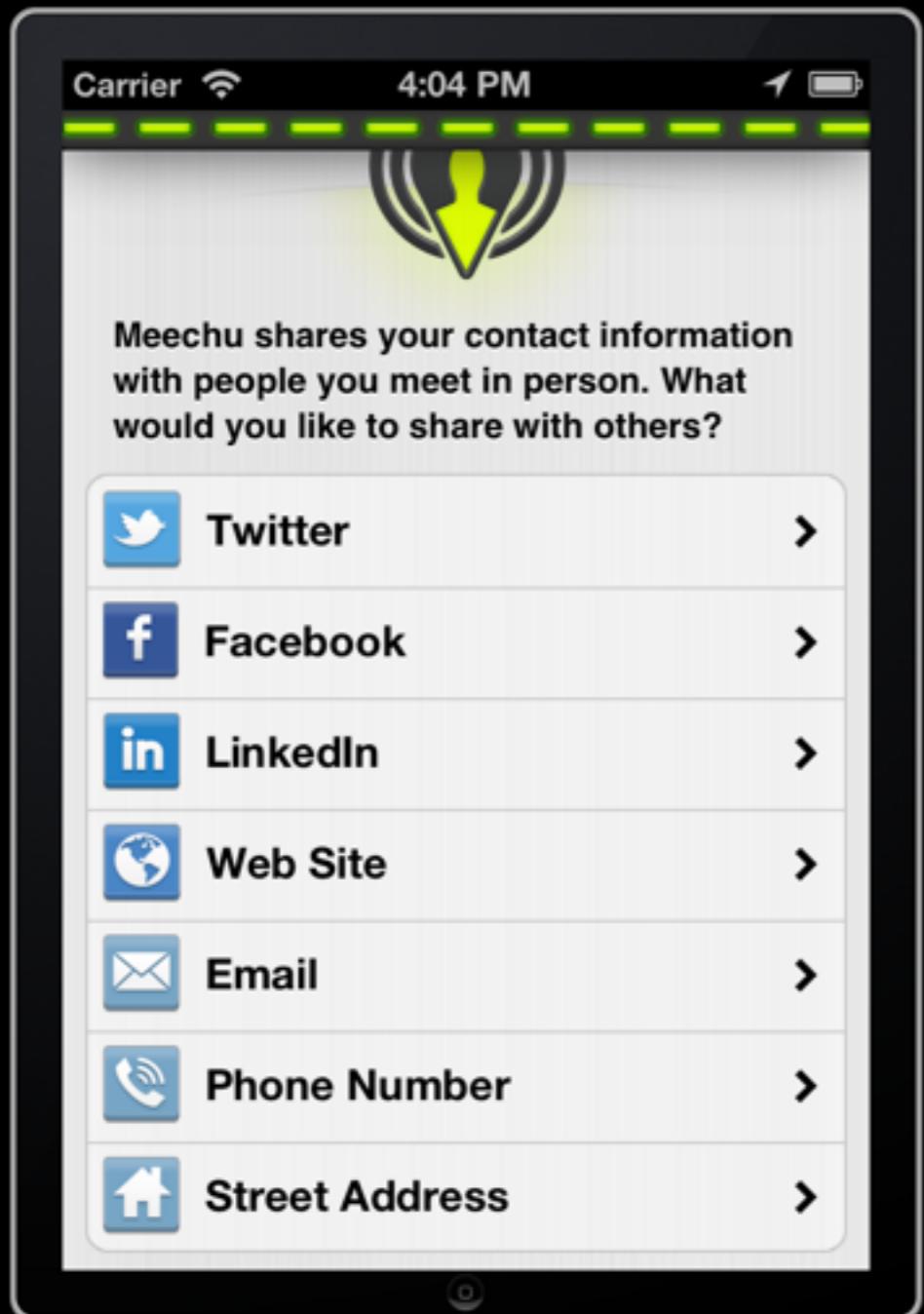
- Match the header label
- There should be seven rows
- The order of the rows should be:
 - Twitter
 - Facebook
 - LinkedIn

Screen Assertions



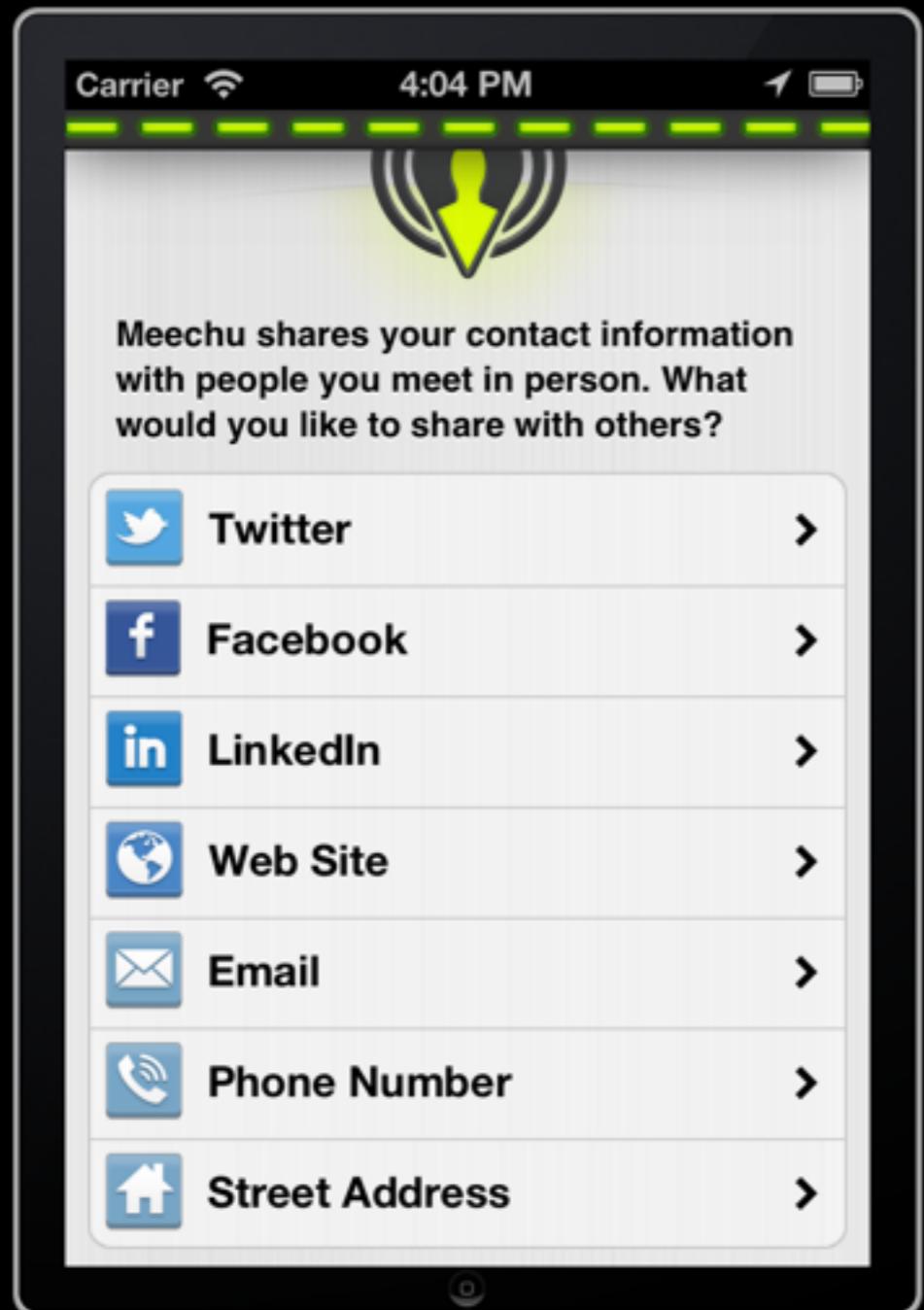
- Match the header label
- There should be seven rows
- The order of the rows should be:
 - Twitter
 - Facebook
 - LinkedIn
 - Web Site

Screen Assertions



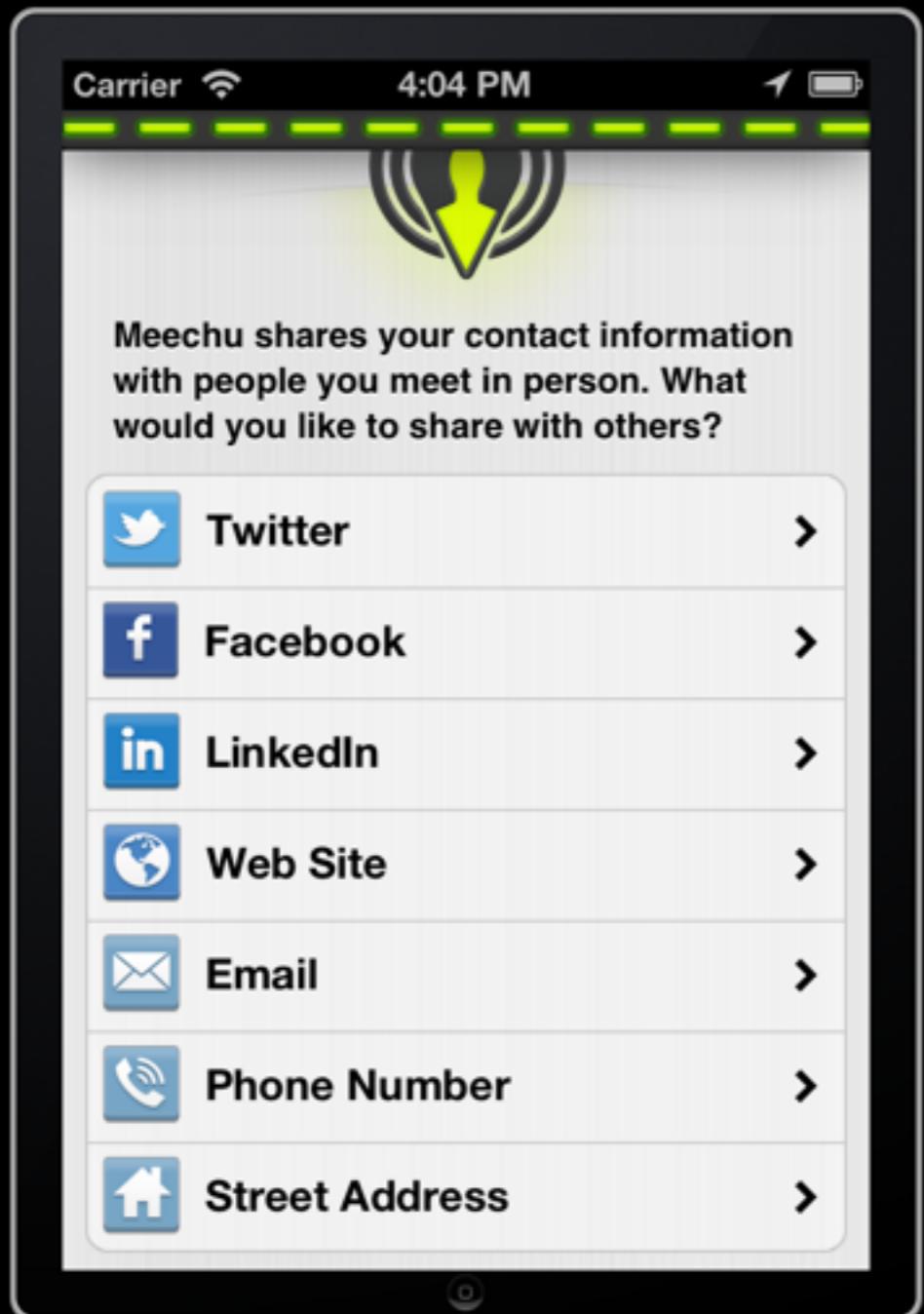
- Match the header label
- There should be seven rows
- The order of the rows should be:
 - Twitter
 - Facebook
 - LinkedIn
 - Web Site
 - Email

Screen Assertions



- Match the header label
- There should be seven rows
- The order of the rows should be:
 - Twitter
 - Facebook
 - LinkedIn
 - Web Site
 - Email
 - Phone Number
 - Street Address

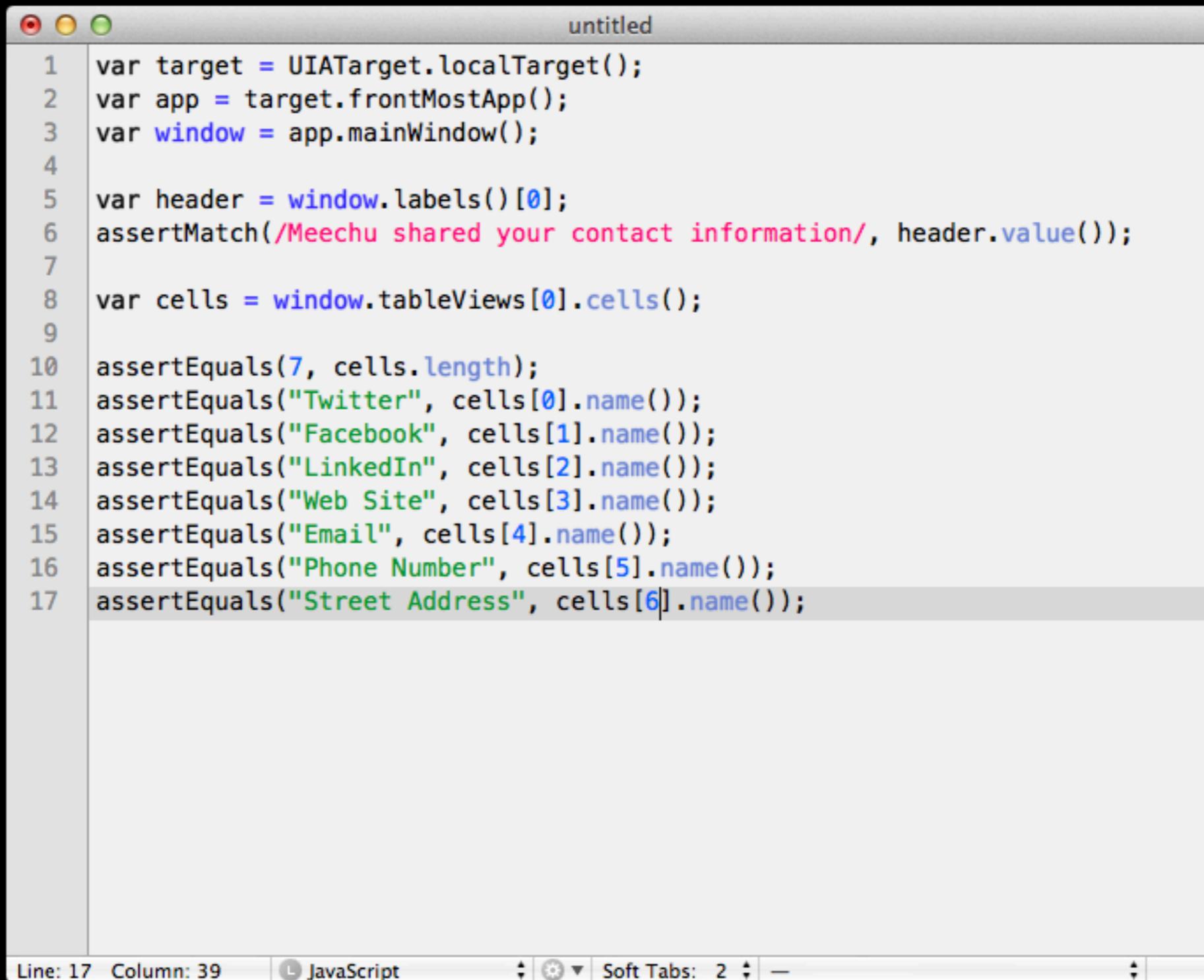
Screen Assertions



- Match the header label
- There should be seven rows
- The order of the rows should be:
 - Twitter
 - Facebook
 - LinkedIn
 - Web Site
 - Email
 - Phone Number
 - StreetAddress

Screen Assertions

First, the hard way...



```
untitled
1 var target = UIATarget.localTarget();
2 var app = target.frontMostApp();
3 var window = app mainWindow();

4
5 var header = window.labels()[0];
assertMatch(/Meechu shared your contact information/, header.value());

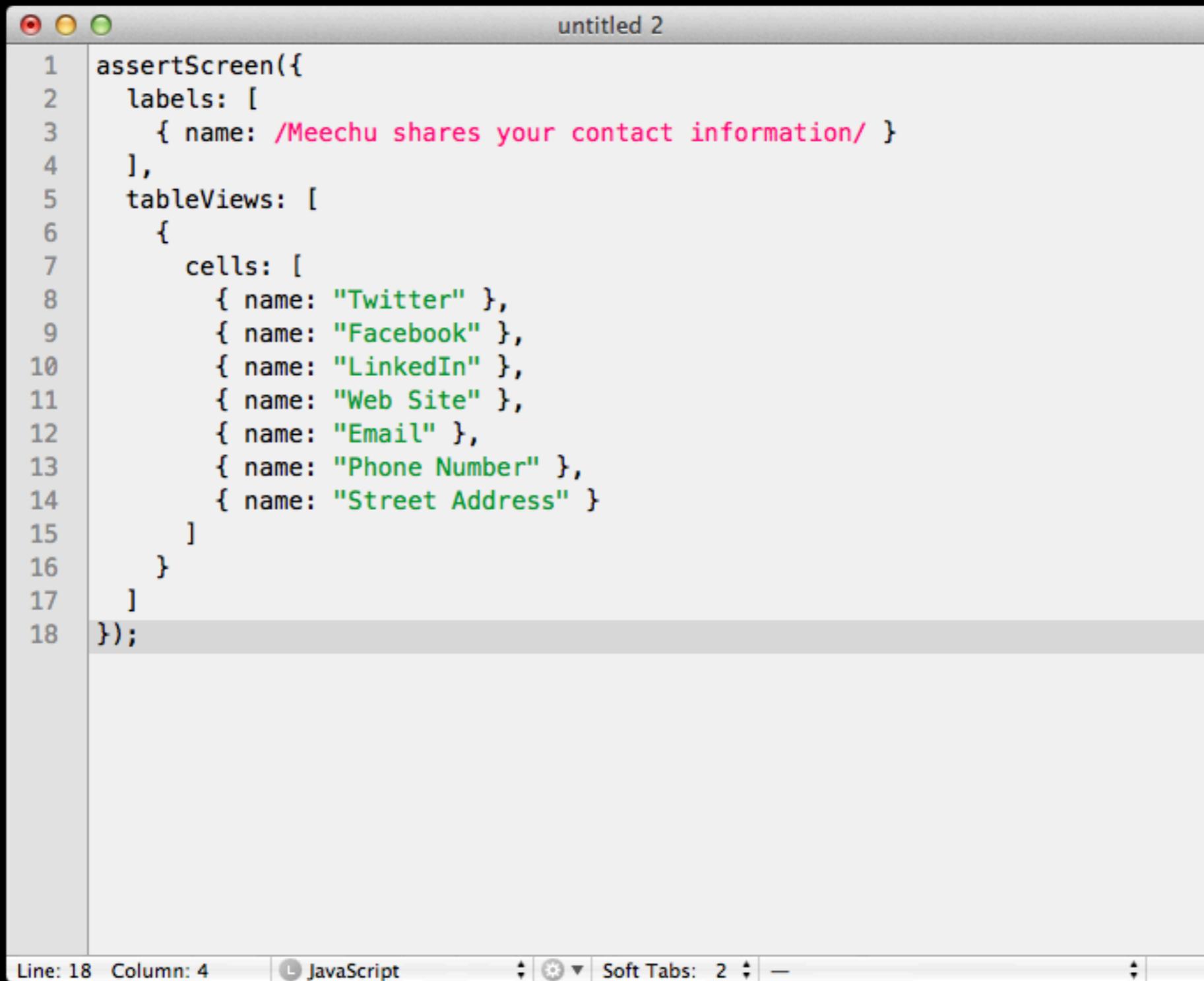
6
7 var cells = window.tableViews[0].cells();

8
9
10 assertEquals(7, cells.length);
11 assertEquals("Twitter", cells[0].name());
12 assertEquals("Facebook", cells[1].name());
13 assertEquals("LinkedIn", cells[2].name());
14 assertEquals("Web Site", cells[3].name());
15 assertEquals("Email", cells[4].name());
16 assertEquals("Phone Number", cells[5].name());
17 assertEquals("Street Address", cells[6].name());
```

Line: 17 Column: 39 L JavaScript Soft Tabs: 2 —

Screen Assertions

Using assertScreen()



The screenshot shows a code editor window titled "untitled 2". The code is written in JavaScript and uses the Meechu library's assertScreen function to check for specific UI elements on a screen. The code is as follows:

```
1 assertScreen({
2   labels: [
3     { name: /Meechu shares your contact information/ }
4   ],
5   tableViews: [
6     {
7       cells: [
8         { name: "Twitter" },
9         { name: "Facebook" },
10        { name: "LinkedIn" },
11        { name: "Web Site" },
12        { name: "Email" },
13        { name: "Phone Number" },
14        { name: "Street Address" }
15      ]
16    }
17  ]
18});
```

The code editor interface includes standard OS X-style window controls (red, yellow, green buttons) and a status bar at the bottom with the text "Line: 18 Column: 4" and "JavaScript".

JavaScript Extensions

Object Extensions

```
extend(screen, {  
    name: "dorkboy";  
});  
  
var dump = dumpProperties(obj);
```

JavaScript Extensions

Array Extensions

```
cells.each(function(i,e) {  
    doThingWithCell(e);  
});  
  
var names = cells.map(function(i,e) {  
    return transform(e);  
});  
  
var hasDork = cells.contains(function(i,e) {  
    return e.name() == "dork boy";  
});
```

JavaScript Extensions

String Extensions

```
" here I am ".trim(); // becomes 'here I am'  
" here I am ".ltrim(); // becomes 'here I am '  
" here I am ".rtrim(); // becomes ' here I am'
```

Zucchini

The screenshot shows the Zucchini iOS testing framework website displayed on an iPad. The top navigation bar includes icons for back, forward, search, and refresh, along with the URL www.zucchiniframework.org/, the time (17:50), and battery level (56%). The main content area features a large green zucchini graphic, the title "Zucchini" in a large green font, and the subtitle "A visual iOS testing framework that loves your apps". A red ribbon banner on the right says "Fork me on GitHub". Below the title, there's a section titled "How it works" with a dark brown background containing step-by-step instructions for interacting with an app's screens. To the right of this are two sections: "Test against your own screenshots" showing a grid of three iPhone 4 screenshots, and "Define your own workflow" showing a Jenkins build queue interface. At the bottom, a footer notes that a Zucchini feature file consists of sections bound to contexts of different application screens, and every screen needs to be backed up by a CoffeeScript file.

Then on the "Live" screen:
Take a screenshot
Tap "Menu"

Then on the "Menu" screen:
Take a screenshot
Tap "Log In"

Then on the "Log In" screen:
Take a screenshot
Tap "Sign up!"

Then on the "Sign Up" screen:
Take a screenshot
Fill the form with "test1", "password", "password"
Wait for "1" second
Confirm "Sign Up"
Wait for "2" seconds
Take a screenshot named "sign up_spinner"

Use natural language for interaction scenarios
Use a simple extensible DSL to specify your interaction or walkthrough scenarios.

Test against your own screenshots
Mark them as pending and promote them to reference as you progress with your project.

Jenkins

Dev iPhone 4 10:15:37, 13 December 2011

about

e2e

Help us localize this page

Define your own workflow
Use Zucchini as you design, code or test your app for regressions.

A Zucchini feature file consists of sections bound to contexts of different application screens. Every screen you proceed to needs to be backed up by a [CoffeeScript](#).

Highlights

Highlights

- Make assertions based on visual comparison

Highlights

- Make assertions based on visual comparison
- High-level DSL

Highlights

- Make assertions based on visual comparison
- High-level DSL
- Supports set up/tear-down outside of
UIAutomation (Ruby)

Highlights

- Make assertions based on visual comparison
- High-level DSL
- Supports set up/tear-down outside of UIAutomation (Ruby)
- Built for Continuous Integration

Highlights

- Make assertions based on visual comparison
- High-level DSL
- Supports set up/tear-down outside of UIAutomation (Ruby)
- Built for Continuous Integration
- Uses CoffeeScript

CoffeeScript

CoffeeScript

(no seriously, it compiles to JavaScript)

CoffeeScript

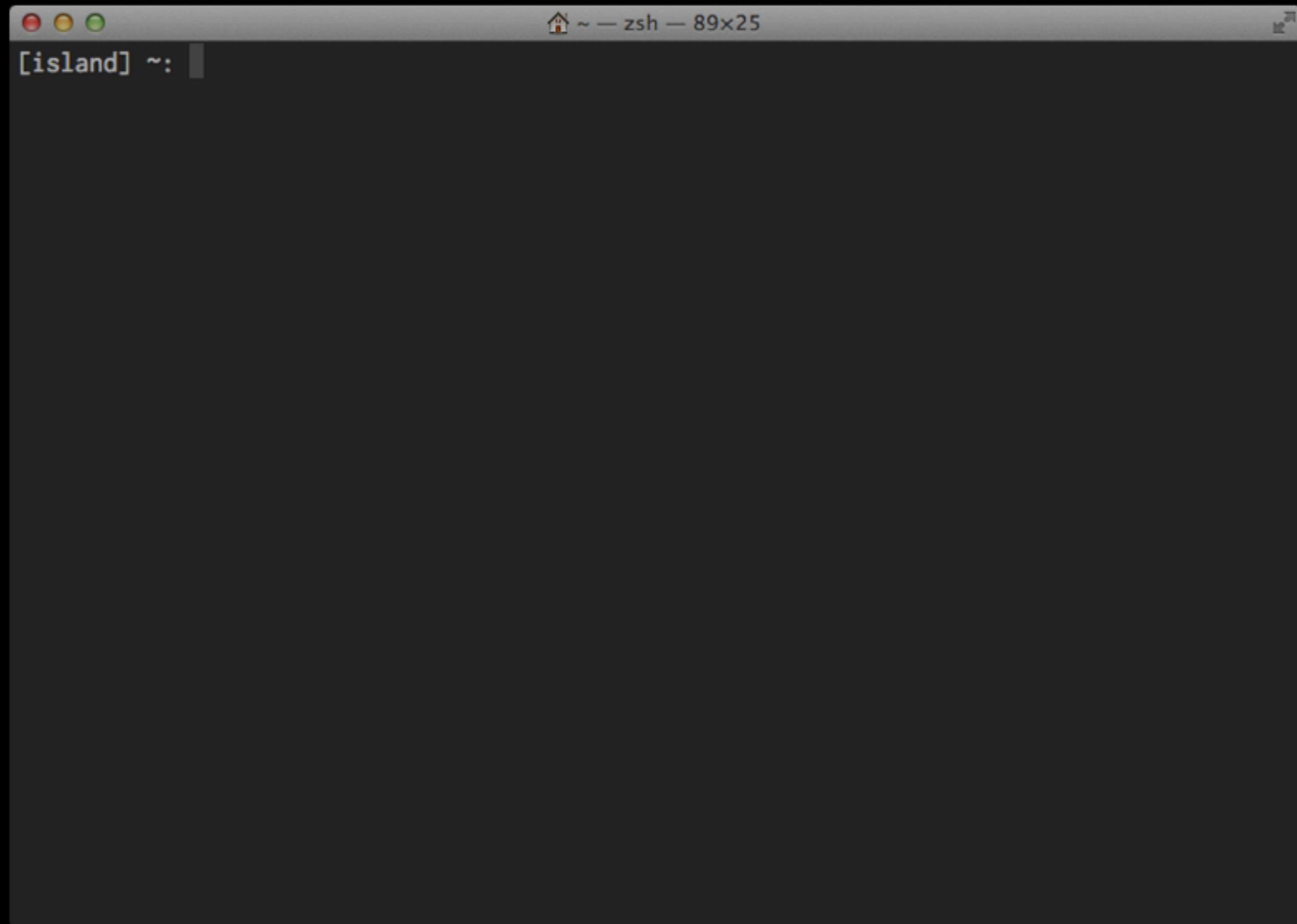
(no seriously, it compiles to JavaScript)

```
var cube, square;
square = function(x) {
  return x * x;
};
cube = function(x) {
  return square(x) * x;
};
```

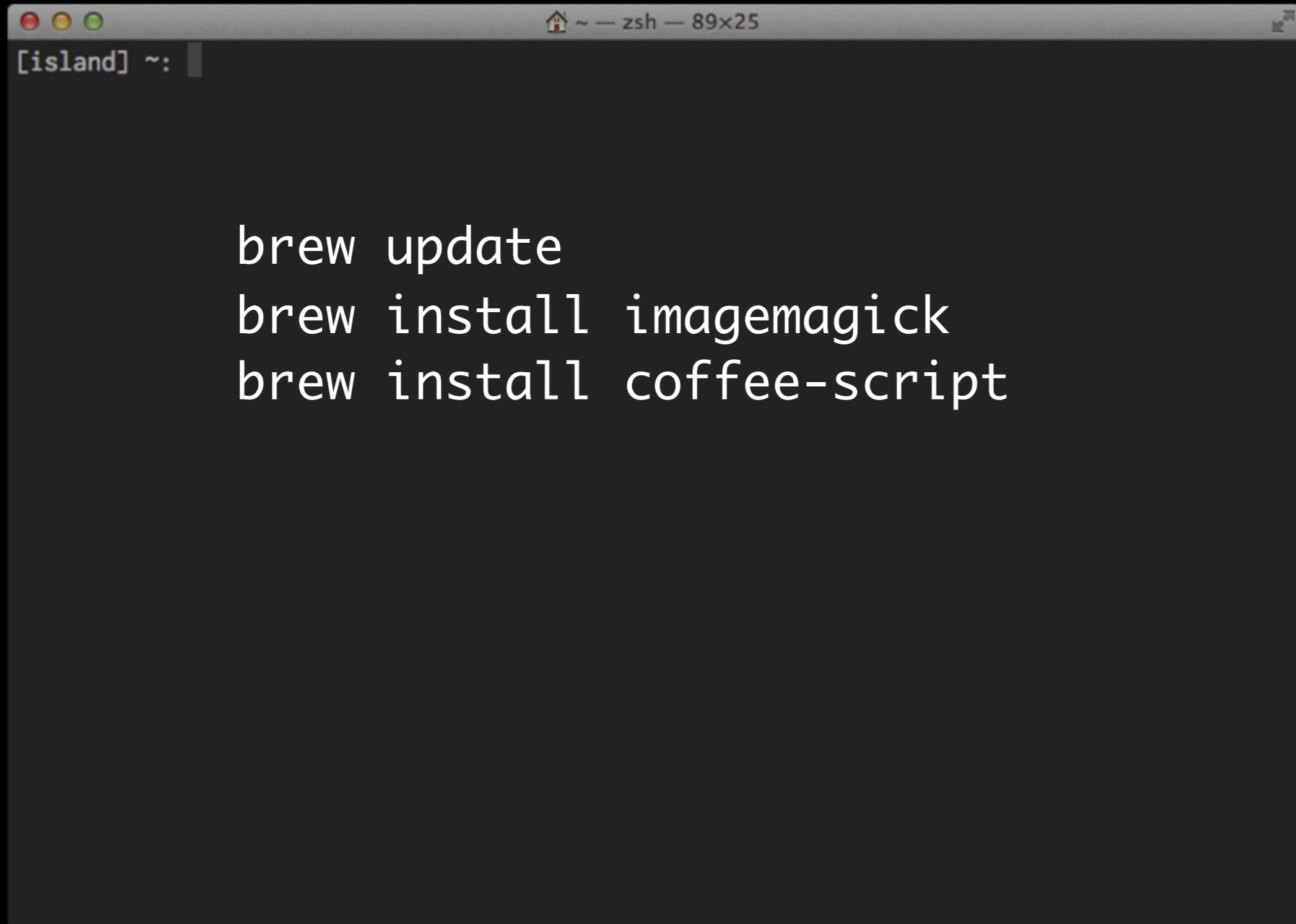
```
var cube, square;  
square = function(x) {  
    return x * x;  
};  
cube = function(x) {  
    return square(x) * x;  
};
```

```
square = (x) -> x * x  
cube   = (x) -> square(x) * x
```

Installation

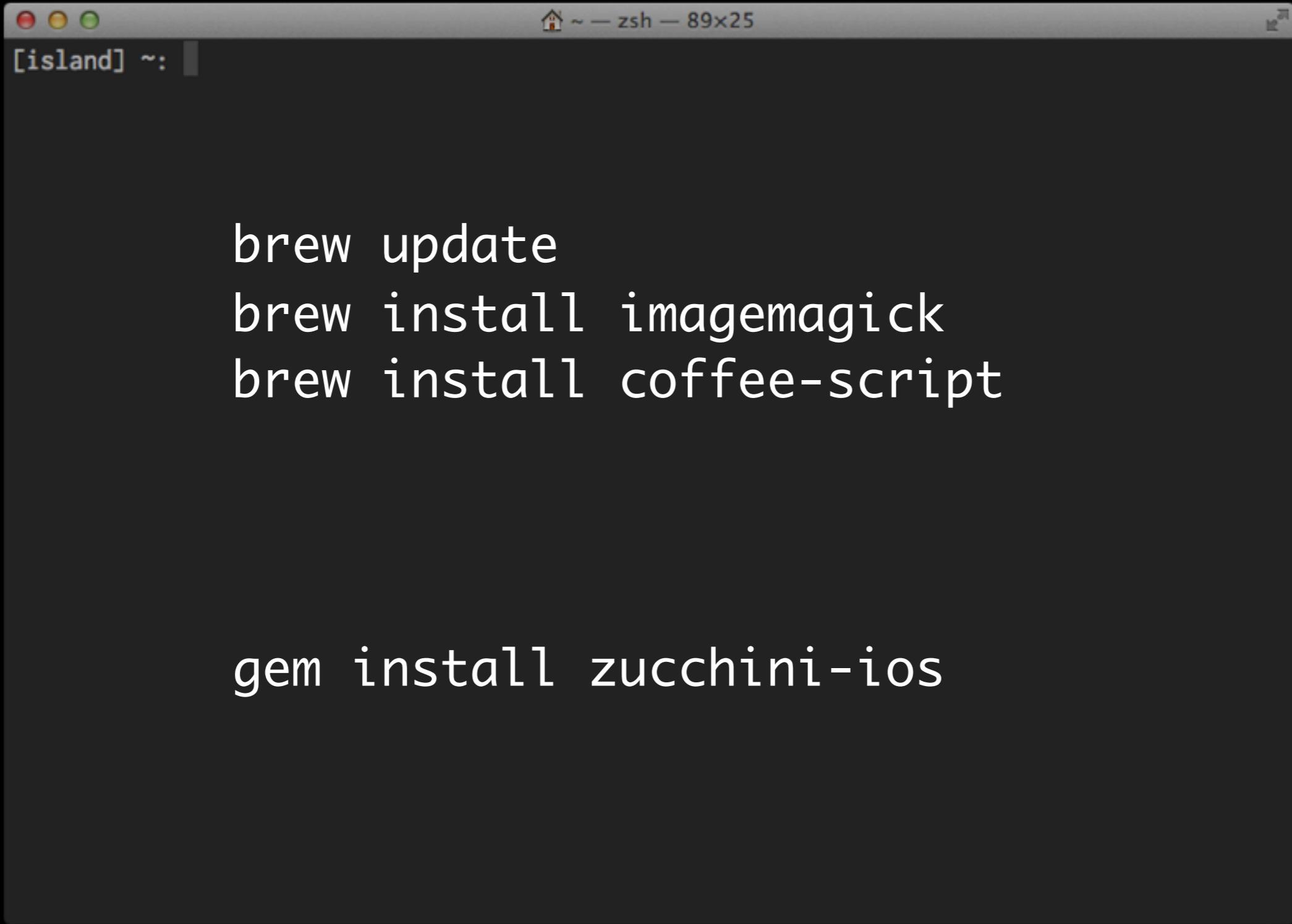


Installation



```
[island] ~: brew update  
[island] ~: brew install imagemagick  
[island] ~: brew install coffee-script
```

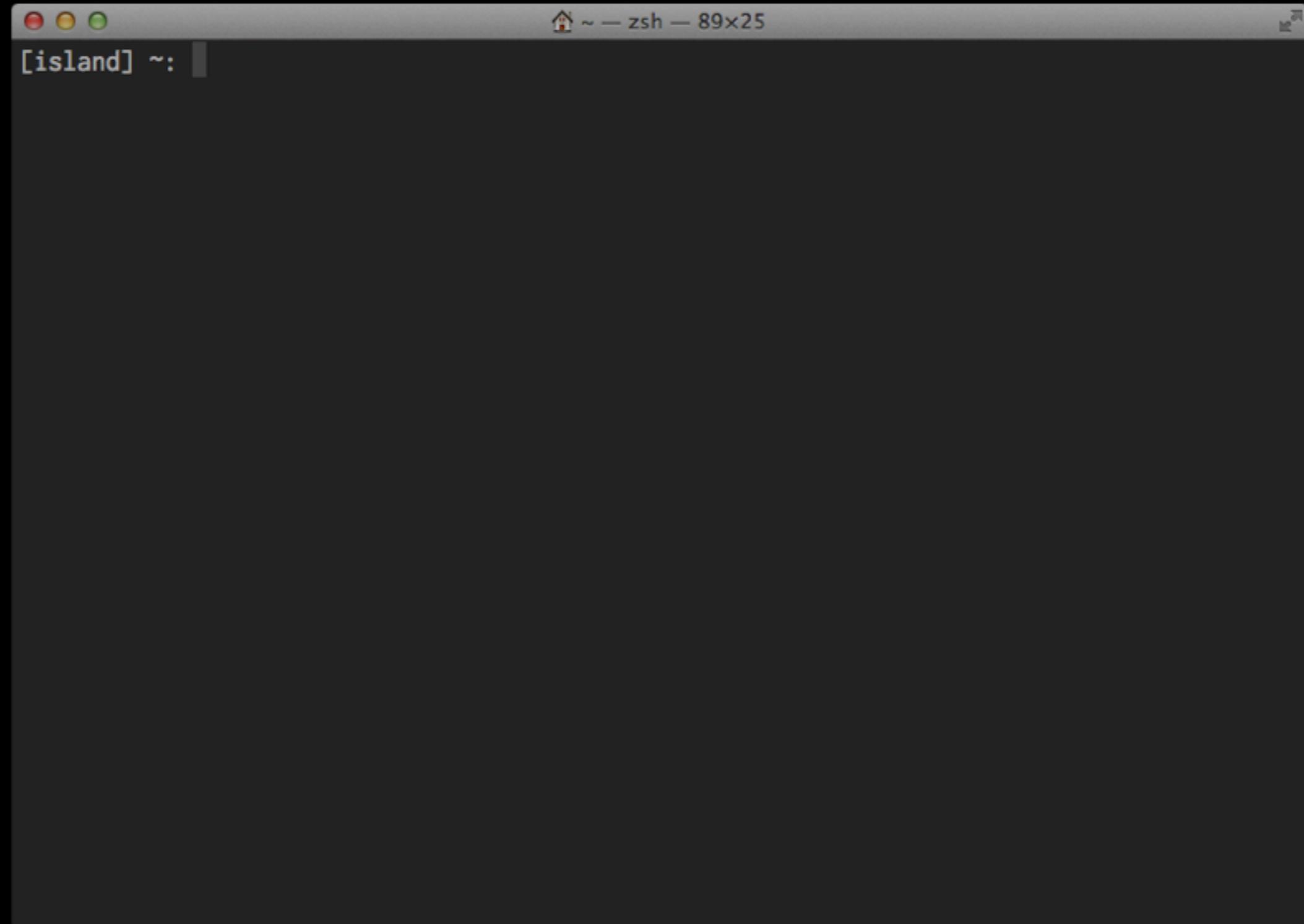
Installation



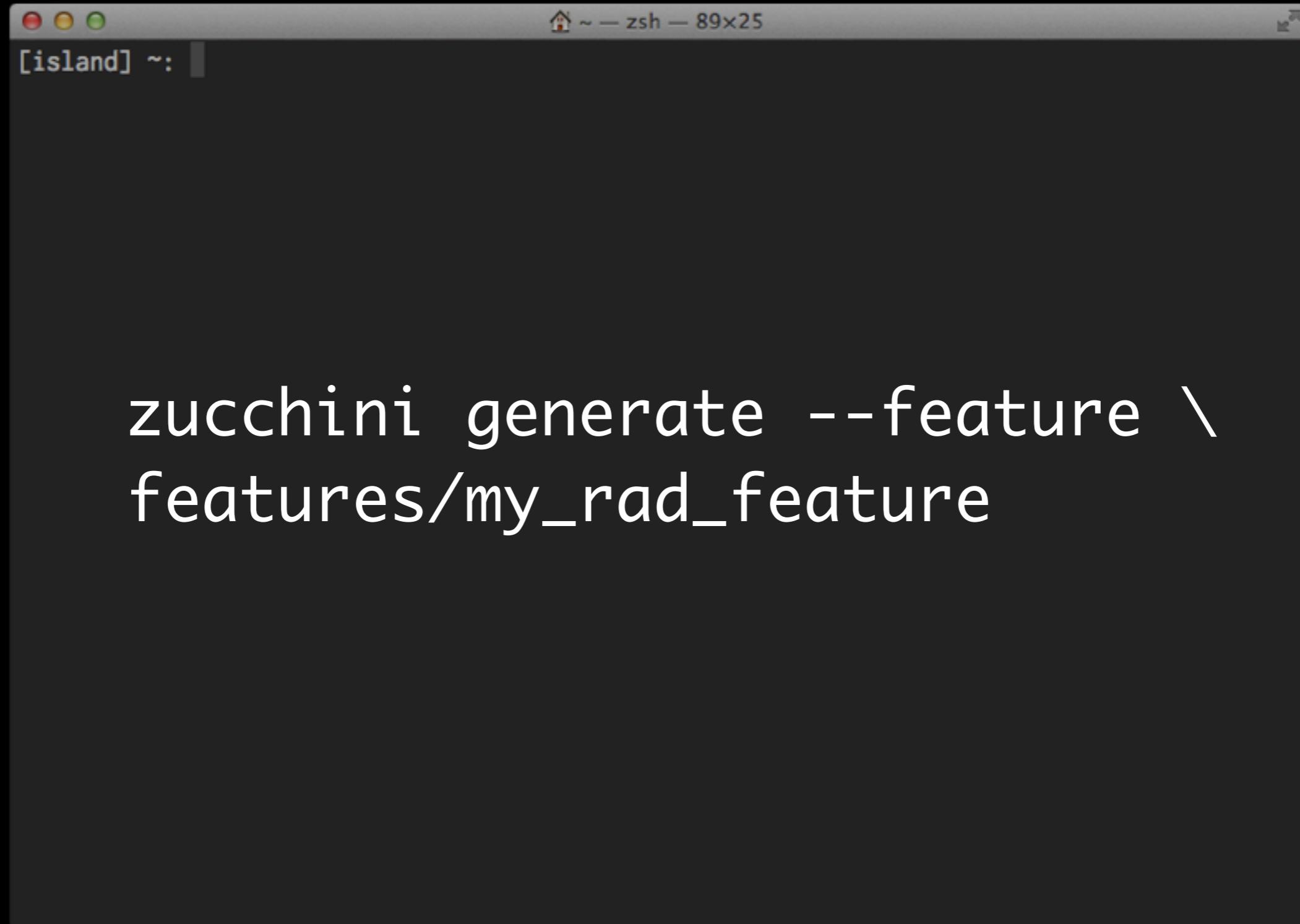
A screenshot of a macOS terminal window. The title bar says "zsh ~ — 89x25". The command line shows the path "[island] ~:" followed by several installation commands:

```
brew update  
brew install imagemagick  
brew install coffee-script  
  
gem install zucchini-ios
```

Creating a “feature”



Creating a “feature”



A screenshot of a terminal window titled "zsh" with the status bar showing "~ — zsh — 89x25". The window title bar has three colored buttons (red, yellow, green). The terminal prompt is "[island] ~: |". Below the prompt, the command "zucchini generate --feature \ features/my_rad_feature" is displayed in white text on a dark background.

```
zucchini generate --feature \
features/my_rad_feature
```

Directory Structure

The screenshot shows a Mac OS X Finder window titled "features". The window displays a "Shared Folder" containing 25 items. The folder structure is as follows:

- first_time_use_facebook (Folder)
- first_time_use_twitter (Folder)
 - feature.zucchini (Document)
- masks (Folder)
- pending (Folder)
- reference (Folder)
 - retina_ios5 (Folder)
 - 01_welcome_first_boot.png (Image)
 - 02_twitter_login.png (Image)
 - 03_me_view.png (Image)
- run_data (Folder)
- setup.rb (Ruby Source)
- support (Folder)
 - config.yml (BBEdit Document)
 - masks (Folder)
 - ipad_ios5.png (Image)
 - low_ios4.png (Image)
 - low_ios5.png (Image)
 - retina_ios5.png (Image)
 - screens (Folder)
 - facebook_login.coffee (Document)
 - me_view.coffee (Document)
 - twitter_login.coffee (Document)
 - welcome.coffee (Document)
- first_time_use (Folder)

The table lists the items with their names, last modified dates, sizes, and kinds.

Name	Date Modified	Size	Kind
first_time_use_facebook	Today 9:26 PM	--	Folder
first_time_use_twitter	Today 9:27 PM	--	Folder
feature.zucchini	Today 9:26 PM	286 bytes	Document
masks	Today 9:27 PM	--	Folder
pending	Today 9:26 PM	--	Folder
reference	Today 9:27 PM	--	Folder
retina_ios5	Today 9:26 PM	--	Folder
01_welcome_first_boot.png	Today 9:26 PM	176 KB	Portable image
02_twitter_login.png	Today 9:26 PM	89 KB	Portable image
03_me_view.png	Today 9:26 PM	196 KB	Portable image
run_data	Today 9:26 PM	--	Folder
setup.rb	Today 9:26 PM	112 bytes	Ruby Source
support	Today 9:27 PM	--	Folder
config.yml	Today 9:26 PM	610 bytes	BBEdit document
masks	Today 9:26 PM	--	Folder
ipad_ios5.png	Today 9:26 PM	7 KB	Portable image
low_ios4.png	Today 9:26 PM	4 KB	Portable image
low_ios5.png	Today 9:26 PM	4 KB	Portable image
retina_ios5.png	Today 9:26 PM	7 KB	Portable image
screens	Today 9:26 PM	--	Folder
facebook_login.coffee	Today 9:26 PM	567 bytes	Document
me_view.coffee	Today 9:26 PM	120 bytes	Document
twitter_login.coffee	Today 9:26 PM	283 bytes	Document
welcome.coffee	Today 9:26 PM	370 bytes	Document
first_time_use	Today 9:26 PM	--	Folder

Directory Structure

Shared Folder				
Name	Date Modified	Size	Kind	
► first_time_use_facebook	Today 9:26 PM	--	Folder	
▼ first_time_use_twitter	Today 9:27 PM	--	Folder	
feature.zucchini	Today 9:26 PM	286 bytes	Document	
► masks	Today 9:27 PM	--	Folder	
► pending	Today 9:26 PM	--	Folder	
▼ reference	Today 9:27 PM	--	Folder	
► retina_ios5	Today 9:26 PM	--	Folder	
01_welcome_first_boot.png	Today 9:26 PM	176 KB	Portab...image	
02_twitter_login.png	Today 9:26 PM	89 KB	Portab...image	
03_me_view.png	Today 9:26 PM	196 KB	Portab...image	
► run_data	Today 9:26 PM	--	Folder	
rb setup.rb	Today 9:26 PM	112 bytes	Ruby Source	
▼ support	Today 9:27 PM	--	Folder	
config.yml	Today 9:26 PM	610 bytes	BBEdit...ument	
► masks	Today 9:26 PM	--	Folder	
ipad_ios5.png	Today 9:26 PM	7 KB	Portab...image	
low_ios4.png	Today 9:26 PM	4 KB	Portab...image	
low_ios5.png	Today 9:26 PM	4 KB	Portab...image	
retina_ios5.png	Today 9:26 PM	7 KB	Portab...image	
▼ screens	Today 9:26 PM	--	Folder	
facebook_login.coffee	Today 9:26 PM	567 bytes	Document	
me_view.coffee	Today 9:26 PM	120 bytes	Document	
twitter_login.coffee	Today 9:26 PM	283 bytes	Document	
welcome.coffee	Today 9:26 PM	370 bytes	Document	
► first_time_use	Today 9:26 PM	--	Folder	

Directory Structure

Shared Folder				
Name	Date Modified	Size	Kind	
► first_time_use_facebook	Today 9:26 PM	--	Folder	
▼ first_time_use_twitter	Today 9:27 PM	--	Folder	
feature.zucchini	Today 9:26 PM	286 bytes	Document	
► masks	Today 9:27 PM	--	Folder	
► pending	Today 9:26 PM	--	Folder	
▼ reference	Today 9:27 PM	--	Folder	
► retina_ios5	Today 9:26 PM	--	Folder	
01_welcome_first_boot.png	Today 9:26 PM	176 KB	Portab...image	
02_twitter_login.png	Today 9:26 PM	89 KB	Portab...image	
03_me_view.png	Today 9:26 PM	196 KB	Portab...image	
► run_data	Today 9:26 PM	--	Folder	
rb setup.rb	Today 9:26 PM	112 bytes	Ruby Source	
▼ support	Today 9:27 PM	--	Folder	
config.yml	Today 9:26 PM	610 bytes	BBEdit...ument	
► masks	Today 9:26 PM	--	Folder	
ipad_ios5.png	Today 9:26 PM	7 KB	Portab...image	
low_ios4.png	Today 9:26 PM	4 KB	Portab...image	
low_ios5.png	Today 9:26 PM	4 KB	Portab...image	
retina_ios5.png	Today 9:26 PM	7 KB	Portab...image	
▼ screens	Today 9:26 PM	--	Folder	
facebook_login.coffee	Today 9:26 PM	567 bytes	Document	
me_view.coffee	Today 9:26 PM	120 bytes	Document	
twitter_login.coffee	Today 9:26 PM	283 bytes	Document	
welcome.coffee	Today 9:26 PM	370 bytes	Document	
► first_time_use	Today 9:26 PM	--	Folder	

Directory Structure

Shared Folder				
Name	Date Modified	Size	Kind	
► first_time_use_facebook	Today 9:26 PM	--	Folder	
▼ first_time_use_twitter	Today 9:27 PM	--	Folder	
feature.zucchini	Today 9:26 PM	286 bytes	Document	
► masks	Today 9:27 PM	--	Folder	
► pending	Today 9:26 PM	--	Folder	
▼ reference	Today 9:27 PM	--	Folder	
► retina_ios5	Today 9:26 PM	--	Folder	
01_welcome_first_boot.png	Today 9:26 PM	176 KB	Portab...image	
02_twitter_login.png	Today 9:26 PM	89 KB	Portab...image	
03_me_view.png	Today 9:26 PM	196 KB	Portab...image	
► run_data	Today 9:26 PM	--	Folder	
rb setup.rb	Today 9:26 PM	112 bytes	Ruby Source	
▼ support	Today 9:27 PM	--	Folder	
config.yml	Today 9:26 PM	610 bytes	BBEdit...ument	
► masks	Today 9:26 PM	--	Folder	
ipad_ios5.png	Today 9:26 PM	7 KB	Portab...image	
low_ios4.png	Today 9:26 PM	4 KB	Portab...image	
low_ios5.png	Today 9:26 PM	4 KB	Portab...image	
retina_ios5.png	Today 9:26 PM	7 KB	Portab...image	
▼ screens	Today 9:26 PM	--	Folder	
facebook_login.coffee	Today 9:26 PM	567 bytes	Document	
me_view.coffee	Today 9:26 PM	120 bytes	Document	
twitter_login.coffee	Today 9:26 PM	283 bytes	Document	
welcome.coffee	Today 9:26 PM	370 bytes	Document	
► first_time_use	Today 9:26 PM	--	Folder	

feature.zucchini

feature.zucchini

Start on the "Welcome" screen:

Wait for "3" seconds

Take a screenshot named "welcome_first_boot"

Tap "Twitter"

Then on the "Twitter Login" screen:

Take a screenshot named "twitter_login"

Login as "meechuapp"

Then on the "Me View" screen:

Take a screenshot named "me_view"

Screen Definition

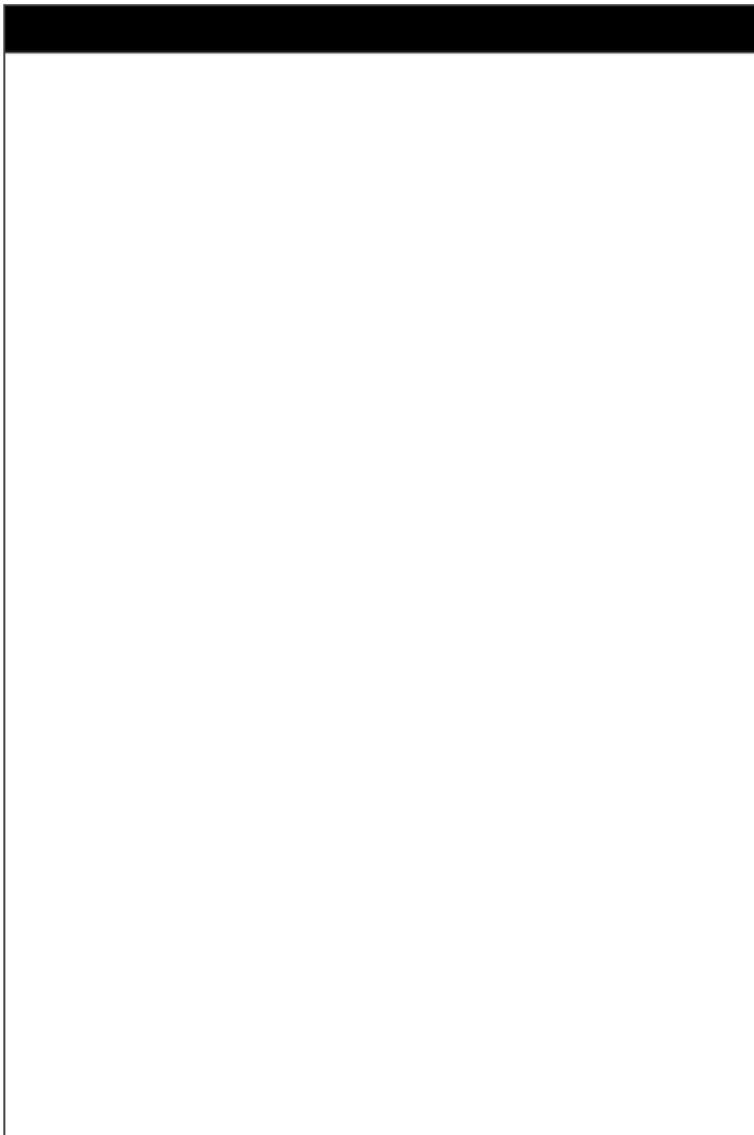
```
class TwitterLoginScreen extends Screen
  anchor: -> view.navigationBars()["Twitter"]

  constructor: ->
    super 'twitter_login'

  extend @actions,
    'Login as "([^\"]*)"': (loginName) ->
      tableView = view.tableViews()[0]
      tableView.cells()[loginName].tap();
```

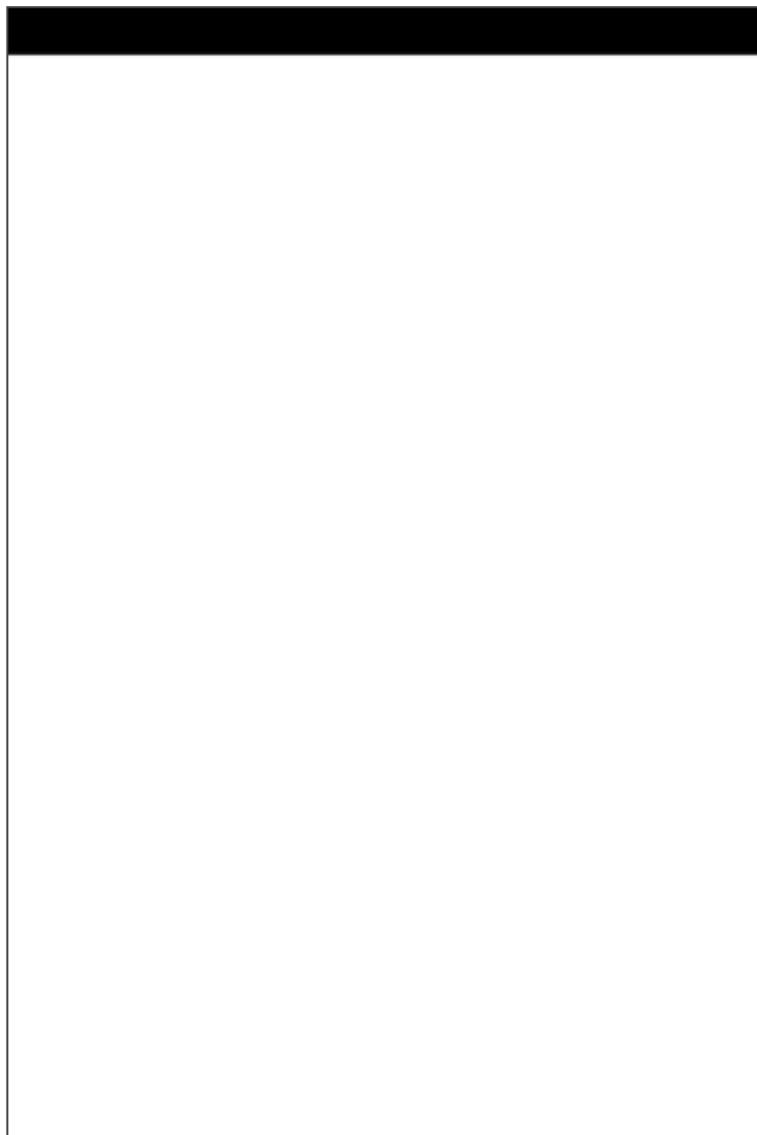
Image Masking

Image Masking

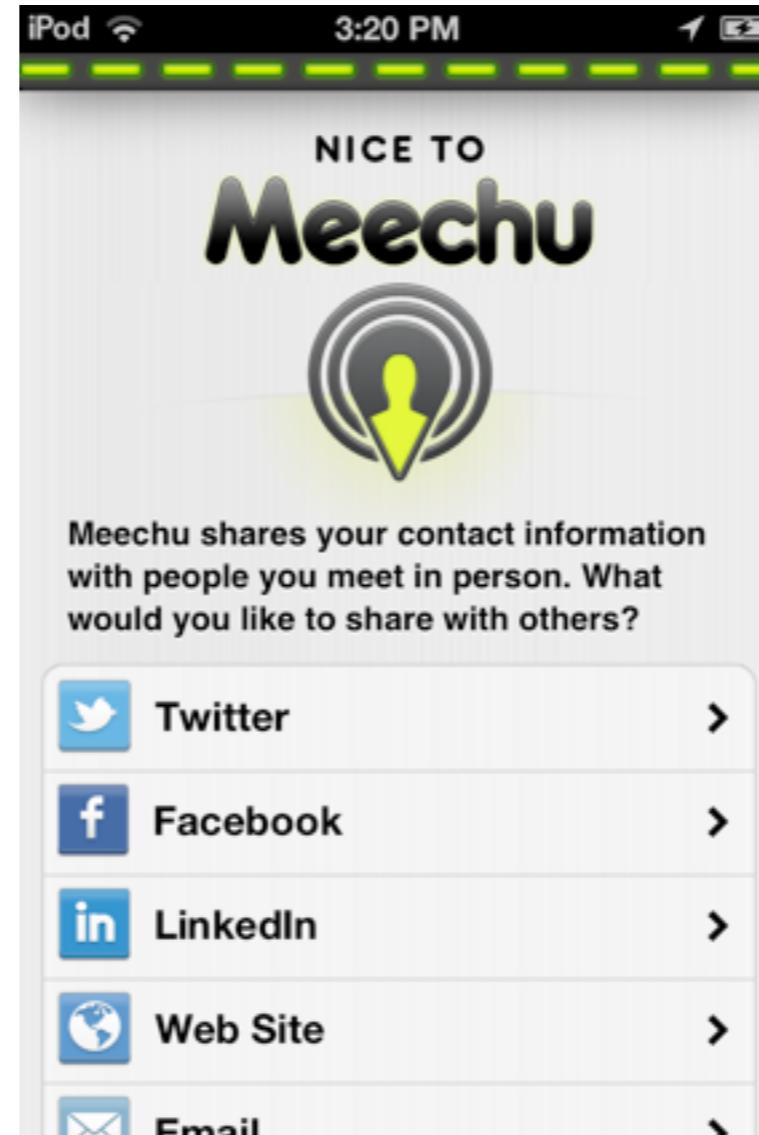


Mask

Image Masking



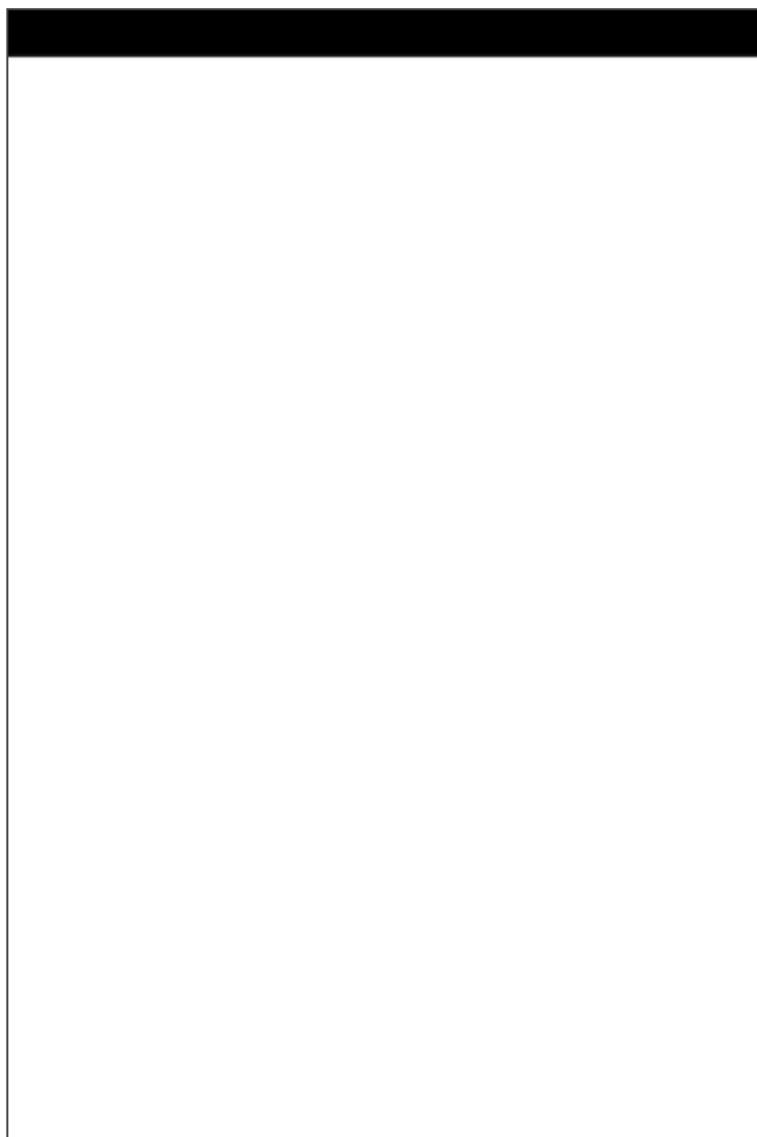
+



Mask

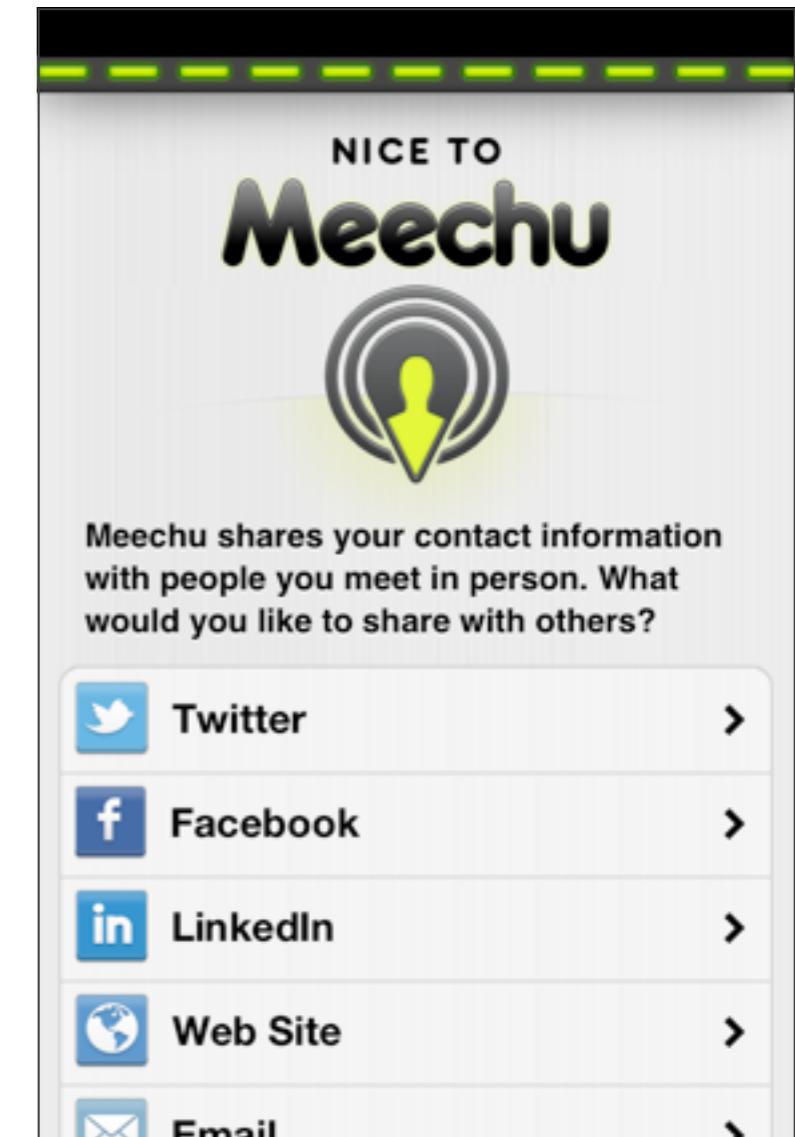
Reference Image

Image Masking



Mask

Reference Image



Comparison Image

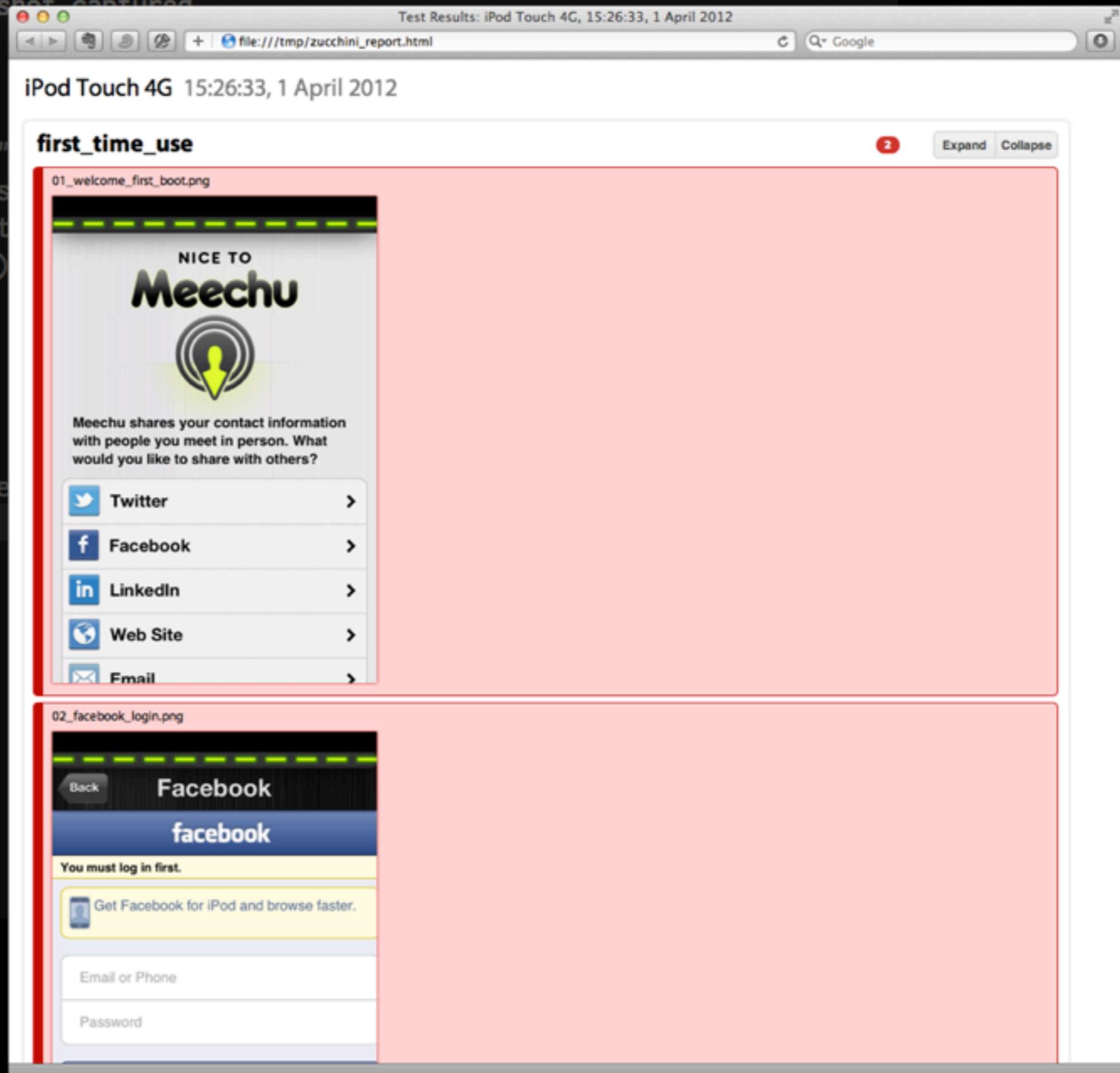
```
[imac] Development/MeechuApp: ZUCCHINI_DEVICE="iPod Touch 4G" zucchini run features  
2012-04-01 22:27:48 +0000 Debug: target.captureRectWithName("{origin:{x:0.00,y:0.00}, size:  
{height:480.00,width:320.00}}", "01_welcome_first_boot")  
2012-04-01 22:27:49 +0000 Screenshot captured.  
2012-04-01 22:27:49 +0000 Debug: target.frontMostApp().mainWindow().tableViews()[0].cells()  
["Facebook"].tap()  
2012-04-01 22:28:00 +0000 Debug: target.captureRectWithName("{origin:{x:0.00,y:0.00}, size:  
{height:480.00,width:320.00}}", "02_facebook_login")  
2012-04-01 22:28:01 +0000 Screenshot captured.  
Instruments Trace Complete (Duration : 32.846035s; Output : /Volumes/Users/alex/Development  
/MeechuApp/instrumentscli0.trace)  
first_time_use:  
0 passed, 2 failed, 0 pending  
  
Failed:  
01_welcome_first_boot.png: no reference or pending screenshot for retina_ios5  
02_facebook_login.png: no reference or pending screenshot for retina_ios5  
[imac] Development/MeechuApp: [git]-[av/zucchini]-
```

```
[imac] Development/MeechuApp: ZUCCHINI_DEVICE="iPod Touch 4G" zucchini run features
2012-04-01 22:27:48 +0000 Debug: target.captureRectWithName("{origin:{x:0.00,y:0.00}, size:
{height:480.00,width:320.00}}", "01_welcome_first_boot")
2012-04-01 22:27:49 +0000 Screenshot captured
2012-04-01 22:27:49 +0000 Debug:
["Facebook"].tap()
2012-04-01 22:28:00 +0000 Debug:
{height:480.00,width:320.00}}", "
2012-04-01 22:28:01 +0000 Screens
Instruments Trace Complete (Durat
/MeechuApp/instrumentscli0.trace)
first_time_use:
0 passed, 2 failed, 0 pending
```

Failed:

01_welcome_first_boot.png: no
02_facebook_login.png: no refe

```
[imac] Development/MeechuApp:
```



```
[imac] Development/MeechuApp: ZUCCHINI_DEVICE="iPod Touch 4G" zucchini run features  
2012-04-01 22:29:05 +0000 Debug: target.captureRectWithName("{origin:{x:0.00,y:0.00}, size:  
{height:480.00,width:320.00}}", "01_welcome_first_boot")  
2012-04-01 22:29:06 +0000 Screenshot captured.  
2012-04-01 22:29:06 +0000 Debug: target.frontMostApp().mainWindow().tableViews()[0].cells()  
["Facebook"].tap()  
2012-04-01 22:29:18 +0000 Debug: target.captureRectWithName("{origin:{x:0.00,y:0.00}, size:  
{height:480.00,width:320.00}}", "02_facebook_login")  
2012-04-01 22:29:18 +0000 Screenshot captured.  
Instruments Trace Complete (Duration : 32.844032s; Output : /Volumes/Users/alex/Development  
/MeechuApp/instrumentscli0.trace)  
first_time_use:  
2 passed, 0 failed, 0 pending  
[imac] Development/MeechuApp: (git)-[av/zucchini]-
```

```
[imac] Development/MeechuApp: ZUCCHINI_DEVICE="iPod Touch 4G" zucchini run features  
2012-04-01 22:29:05 +0000 Debug: target.captureRectWithName("{origin:{x:0.00,y:0.00}, size:  
{height:480.00,width:320.00}}", "01_welcome first boot")  
2012-04-01 22:29:06 +0000 Screen  
2012-04-01 22:29:06 +0000 Debug:  
["Facebook"].tap()  
2012-04-01 22:29:18 +0000 Debug:  
{height:480.00,width:320.00}}",  
2012-04-01 22:29:18 +0000 Screen  
Instruments Trace Complete (Duration: 0.000s)  
/MeechuApp/instrumentscli0.trace  
first_time_use:  
2 passed, 0 failed, 0 pending  
[imac] Development/MeechuApp:
```

Test Results: iPod Touch 4G, 15:23:31, 1 April 2012

iPod Touch 4G 15:23:31, 1 April 2012

first_time_use

01_welcome_first_boot.png

NICE TO
Meechu

Meechu shares your contact information with people you meet in person. What would you like to share with others?

- Twitter
- Facebook
- LinkedIn
- Web Site
- Email

02_facebook_login.png

Back Facebook

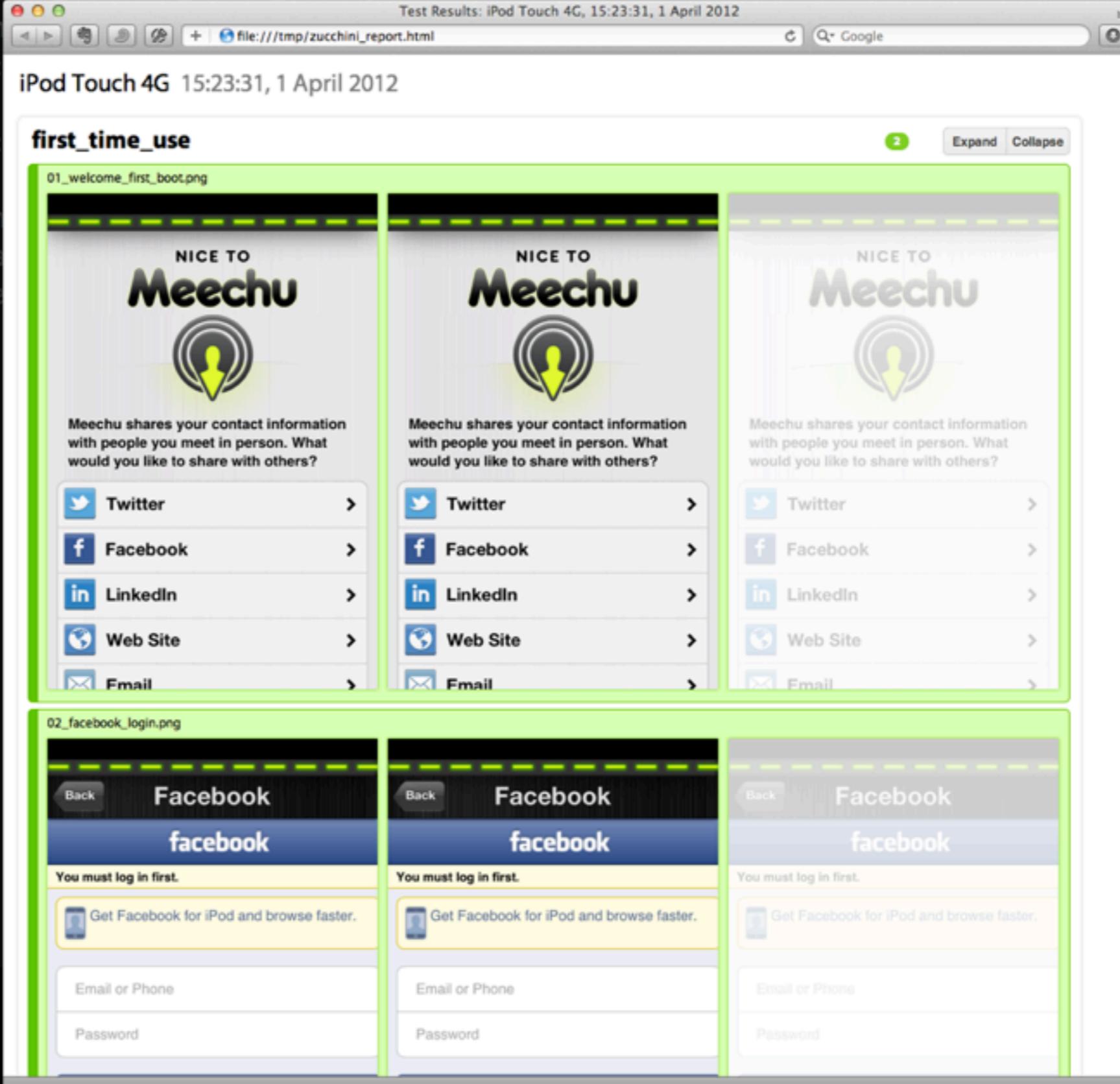
You must log in first.

Get Facebook for iPod and browse faster.

Email or Phone

Password

Expand Collapse



One Man's Opinion

One Man's Opinion

- Addresses setup and teardown externally

One Man's Opinion

- Addresses setup and teardown externally
- A picture is worth a 1000 words

One Man's Opinion

- Addresses setup and teardown externally
- A picture is worth a 1000 words
- Too many languages

One Man's Opinion

- Addresses setup and teardown externally
- A picture is worth a 1000 words
- Too many languages
- Difficult to debug

One Man's Opinion

- Addresses setup and teardown externally
- A picture is worth a 1000 words
- Too many languages
- Difficult to debug
- Poorly documented

UIAutomation Revisited

UIAutomation Revisited

- Still not rock-solid and reliable

UIAutomation Revisited

- Still not rock-solid and reliable
- Timing

UIAutomation Revisited

- Still not rock-solid and reliable
 - Timing
 - False positives

UIAutomation Revisited

- Still not rock-solid and reliable
 - Timing
 - False positives
- S I O O O W W W W

UIAutomation Revisited

UIAutomation Revisited

- Major shortcomings fixed in 5.0

UIAutomation Revisited

- Major shortcomings fixed in 5.0
- Each release gets more stable

UIAutomation Revisited

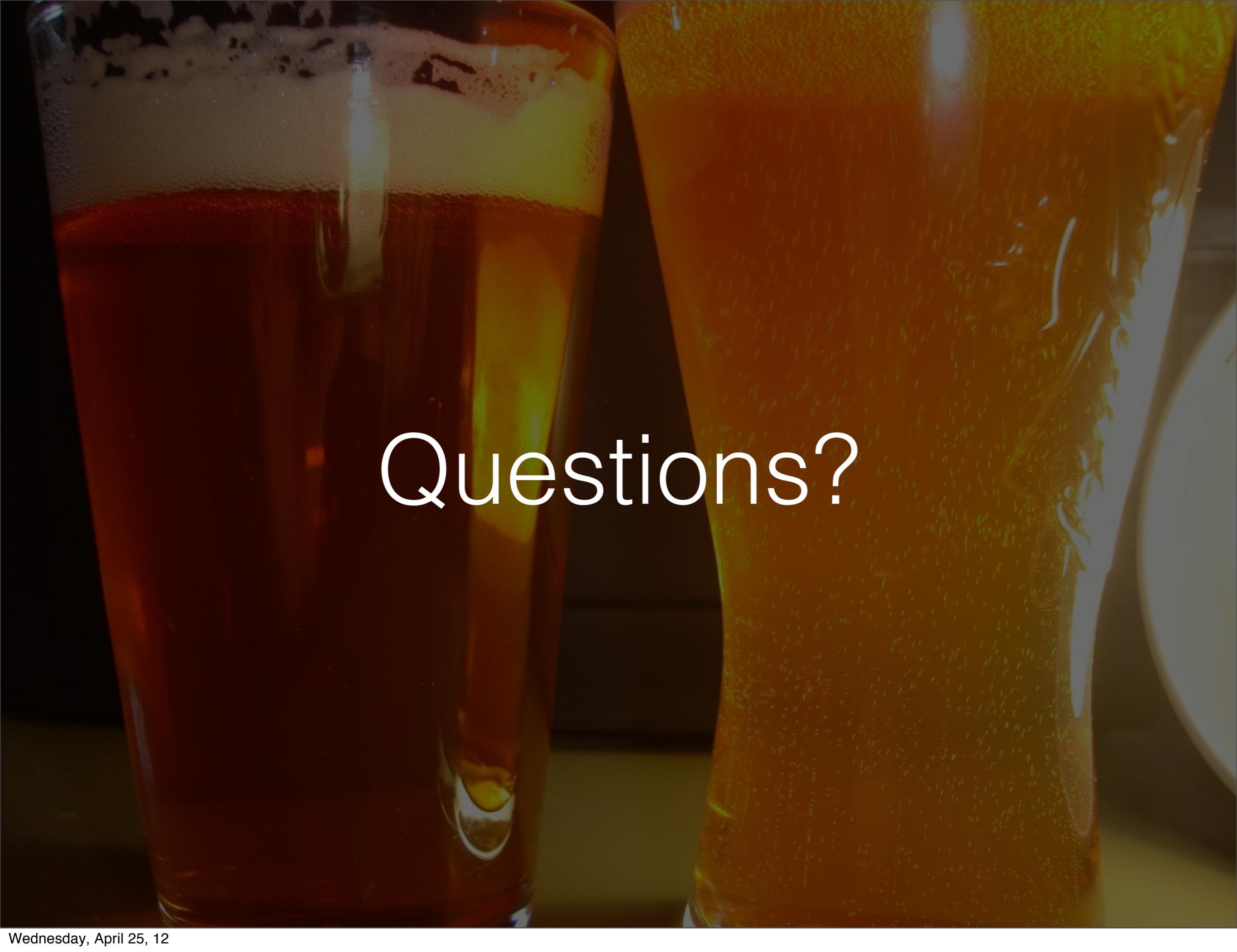
- Major shortcomings fixed in 5.0
- Each release gets more stable
- Apple seems interested in improving it

More Info

- “UIAutomation” group “Core Services Layer”
- “Instruments User Guide” – look for UIAutomation in “Built-in Instruments”
- <http://alexvollmer.com/tags/uiautomation>



Wednesday, April 25, 12

A close-up photograph of two glasses filled with beer. The beer has a thick, white head of foam. The glass on the left is dark brown, while the one on the right is clear. The background is dark and out of focus.

Questions?

Thank You.