

Technical stuff now

Specta

<https://github.com/specta/specta>

Based on XCTest

Minimalistic implementation

Syntax

```
SPEC_BEGIN(Example)
```

```
describe(@"Example specs", ^{
```

```
});
```

```
SPEC_END
```

```
SPEC_BEGIN(Example)
```

```
describe("@Example specs", ^{
```

```
});
```

```
SPEC_END
```

```
SPEC_BEGIN(Example)
```

```
describe("@Example specs", ^{
```

```
});
```

```
SPEC_END
```


SPEC_BEGIN(Example)

```
describe("@Example specs", ^{  
  
});
```

SPEC_END

SPEC_BEGIN(Example)

```
describe(@"Example specs", ^{
```

```
  it(@"should check compiler sanity", ^{  
    expect(YES).to.beTruthy();  
  });
```

```
});
```

SPEC_END

Describe/Context blocks

Used to make tests more
readable.

And isolate behaviour for
different scenarios.

```
describe(@"NSNumber", ^{  
    describe(@"when created with the default constructor", ^{  
        it(@"should have 0 as contained int value", ^{  
            NSNumber *number = [[NSNumber alloc] init];  
            expect([number integerValue]).to.equal(0);  
        });  
    });  
  
    context(@"when constructed with an int", ^{  
        it(@"should have 42 as contained int value", ^{  
            NSNumber *number = [[NSNumber alloc] initWithInt:42];  
            expect([number integerValue]).to.equal(42);  
        });  
    });  
});
```

You can have
as many
nested
describes as
you want.

```
SetupCheckSpecs > iPhone Retina (3.5-inch) Finished running SetupChe
SetupCheck > SetupCheckSpecs > ExampleSpec.m > SPEC_BEGIN()
1 #import "Specs.h"
2
3 SPEC_BEGIN(ExampleSpec)
4
5 describe(@"Example specs on NSString", ^{
6
7     fit(@"lowercaseString returns a new string with everything in lower case", ^{
8         expect([@"FOOBar" lowercaseString]).to.equal(@"foobara");
9     });
10
11     it(@"length returns the number of characters in the string", ^{
12         expect([@"internationalization" length]).to.equal(20);
13     });
14
15     describe(@"isEqualToString:", ^{
16         it(@"should return true if the strings are the same", ^{
17             expect([@"someString" isEqualToString:@"someString"]).to.beTruthy();
18         });
19
20         it(@"should return false if the strings are not the same", ^{
21             expect([@"someString" isEqualToString:@"anotherString"]).to.beFalsy();
22         });
23     });
24 });
25
26 SPEC_END
27
28
```

Before/After each
blocks

```
beforeEach(^{
    appDelegate = [[AppDelegate alloc] init];
});

afterEach(^{
    appDelegate = nil;
});

it(@"should have a window", ^{
    expect(appDelegate.window).to.beKindOfClass([UIWindow class]);
});
```


1

```
beforeEach(^{  
    appDelegate = [[AppDelegate alloc] init];  
});
```

2

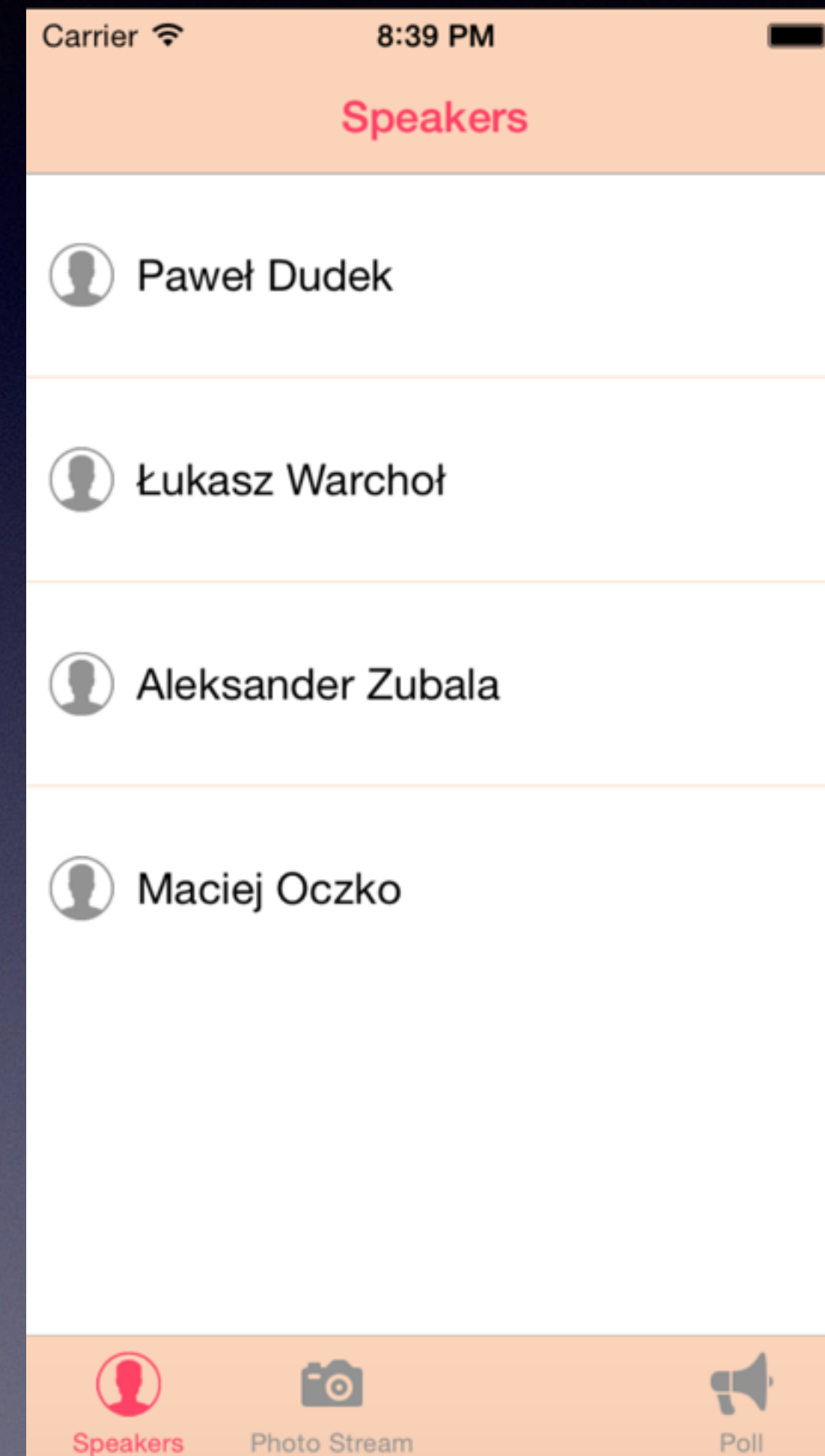
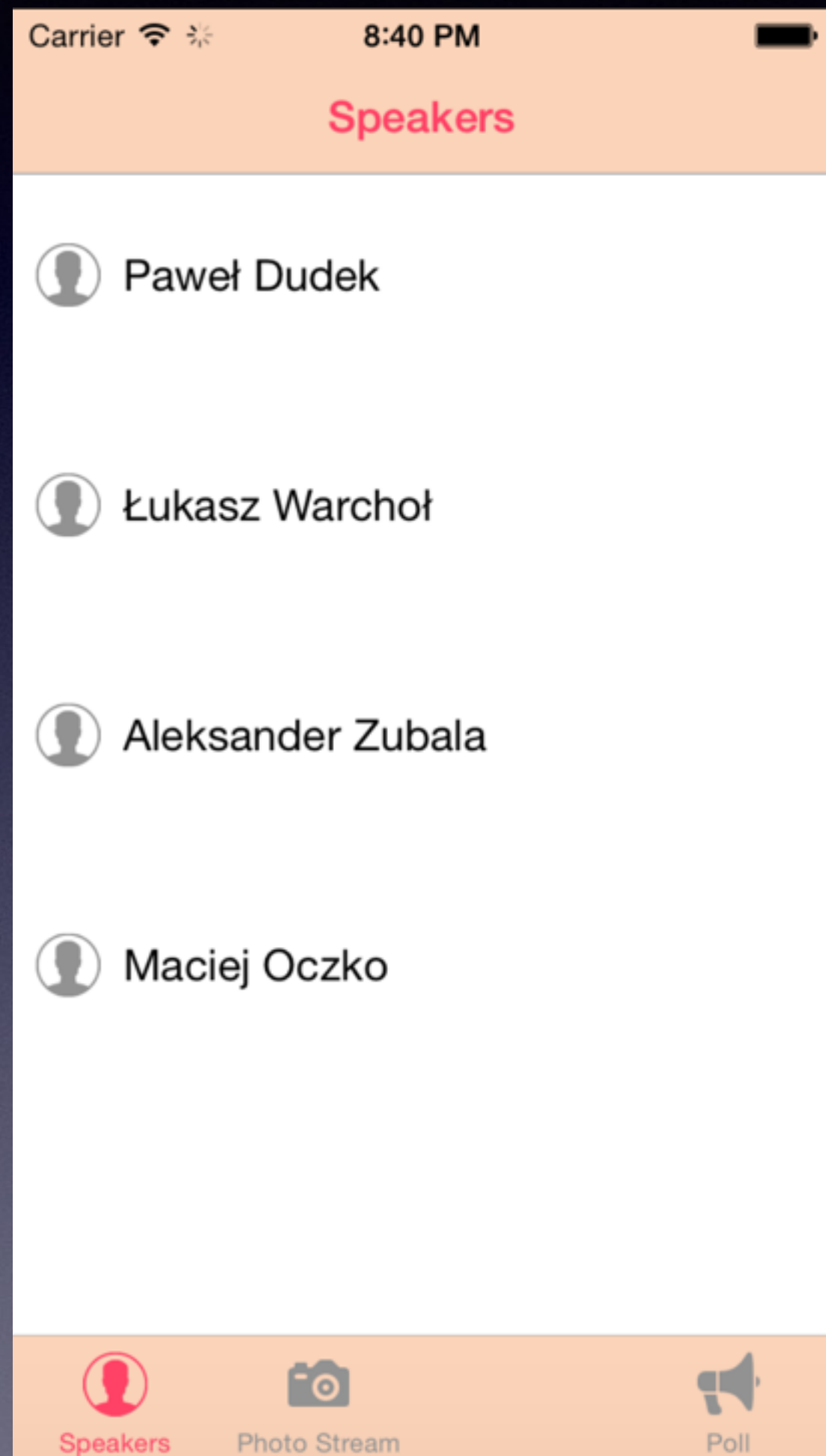
```
it(@"should have a window", ^{  
    expect(appDelegate.window).to.beKindOfClass([UIWindow class]);  
});
```

3

```
afterEach(^{  
    appDelegate = nil;  
});
```

Let's write our very first
unit test!

Hands on!



Configuring tests

Focusing tests

Focusing tests

```
fdescribe(@"Example specs on NSString", ^{  
  
  fit(@"lowercaseString returns a new string with  
  everything in lower case", ^{  
  
    fcontext(@"init with damping", ^{
```

PENDING

PENDING

```
it(@"lowercaseString returns a new string with  
everything in lower case", PENDING);
```


x'ing tests

x'ing tests

```
xdescribe(@"Example specs on NSString", ^{
```

```
xit(@"lowercaseString returns a new string with  
everything in lower case", ^{
```

```
xcontext(@"init with damping", ^{
```

Unit tests results

Unit tests results

How to understand the output?

Xcode, AppCode, Command Line

All give the same results. Devil is in the details

(...)

-[SpecSuiteName passing_spec_name]

Test Case '-[SpecSuiteName passing_spec_name]' started.

Test Case '-[SpecSuiteName passing_spec_name]' passed

(0.271 seconds).

-[SpecSuiteName failing_spec_name]

Test Case '-[SpecSuiteName failing_spec_name]' started.

Test Case '-[SpecSuiteName failing_spec_name]' failed

(0.002 seconds).

(...)

Executed 2 tests, with 1 failure (1 unexpected) in 0.273
(0.278) seconds

2 tests; 0 skipped; 1 failure; 1 exception; 0 pending

(...)

–[SpecSuiteName passing_spec_name]

Test Case '–[SpecSuiteName passing_spec_name]' started.

Test Case '–[SpecSuiteName passing_spec_name]' passed

(0.271 seconds).

–[SpecSuiteName failing_spec_name]

Test Case '–[SpecSuiteName failing_spec_name]' started.

Test Case '–[SpecSuiteName failing_spec_name]' failed

(0.002 seconds).

(...)

Executed 2 tests, with 1 failure (1 unexpected) in 0.273
(0.278) seconds

2 tests; 0 skipped; 1 failure; 1 exception; 0 pending

(...)

-[SpecSuiteName passing_spec_name]

Test Case '-[SpecSuiteName passing_spec_name]' started.

Test Case '-[SpecSuiteName passing_spec_name]' passed

(0.271 seconds).

-[SpecSuiteName failing_spec_name]

Test Case '-[SpecSuiteName failing_spec_name]' started.

Test Case '-[SpecSuiteName failing_spec_name]' failed

(0.002 seconds).

(...)

Executed 2 tests, with 1 failure (1 unexpected) in 0.273
(0.278) seconds

2 tests; 0 skipped; 1 failure; 1 exception; 0 pending

(...)

-[SpecSuiteName passing_spec_name]

Test Case '-[SpecSuiteName passing_spec_name]' started.

Test Case '-[SpecSuiteName passing_spec_name]' passed
(0.271 seconds).

-[SpecSuiteName failing_spec_name]

Test Case '-[SpecSuiteName failing_spec_name]' started.

Test Case '-[SpecSuiteName failing_spec_name]' failed
(0.002 seconds).

(...)

**Executed 2 tests, with 1 failure (1 unexpected) in 0.273
(0.278) seconds**

2 tests; 0 skipped; 1 failure; 1 exception; 0 pending

(...)

-[SpecSuiteName passing_spec_name]

Test Case '-[SpecSuiteName passing_spec_name]' started.

Test Case '-[SpecSuiteName passing_spec_name]' passed

(0.271 seconds).

-[SpecSuiteName failing_spec_name]

Test Case '-[SpecSuiteName failing_spec_name]' started.

Test Case '-[SpecSuiteName failing_spec_name]' failed

(0.002 seconds).

(...)

Executed 2 tests, with 1 failure (1 unexpected) in 0.273
(0.278) seconds

2 tests; 0 skipped; 1 failure; 1 exception; 0 pending

Run your tests from
command line.

“Perfect” setup:
Have your tests run each
time you change something
in a file.

Enhance your tests output.

(...)

-[SpecSuiteName passing_spec_name]

Test Case '-[SpecSuiteName passing_spec_name]' started.

Test Case '-[SpecSuiteName passing_spec_name]' passed

(0.271 seconds).

-[SpecSuiteName failing_spec_name]

Test Case '-[SpecSuiteName failing_spec_name]' started.

Test Case '-[SpecSuiteName failing_spec_name]' failed

(0.002 seconds).

(...)

Executed 2 tests, with 1 failure (1 unexpected) in 0.273
(0.278) seconds

2 tests; 0 skipped; 1 failure; 1 exception; 0 pending

```
Project — bash — 155x40
eldudi@Syfiasz-Air (master *) ~/Workspace/taptera/showcase-ios/Project: rake
Executing xcodebuild test -workspace Showcase.xcworkspace -scheme Showcase -sdk iphonesimulator -destination "platform=iOS Simulator,name=iPad Retina,OS=7.1" | xcpretty -c --no-utf --test

[!] ld: warning: directory not found for option '-F/Applications/Xcode 5.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/Library/Frameworks'

.....FF
.....F.....

BoxEventSpec
BoxEvent_initialization_should_have_proper_created_at_date, expected: 2014-07-03 06:11:57 +0000, got: 2014-07-03 07:11:57 +0000
/Users/eldudi/Workspace/taptera/showcase-ios/Project/Showcase/Classes/Box/Events/SDK/BoxEventSpec.m:69

BoxEvent_initialization_should_have_proper_recorded_at_date, expected: 2014-07-03 06:11:57 +0000, got: 2014-07-03 07:11:57 +0000
/Users/eldudi/Workspace/taptera/showcase-ios/Project/Showcase/Classes/Box/Events/SDK/BoxEventSpec.m:74

BoxFolderSpec
BoxFolder_theme_asset_version_should_use_both_date_and_size_as_theme_asset_version, expected: 42-43, got: -3558-43
/Users/eldudi/Workspace/taptera/showcase-ios/Project/Showcase/Classes/Box/Theme/Assets/BoxFolder+ThemeAssetSpec.m:26

Executed 2504 tests, with 3 failures (3 unexpected) in 55.300 (56.009) seconds
** TEST FAILED **
```

Test Output

xctool vs xcpretty

<https://github.com/facebook/xctool>

<https://github.com/supermarin/xcpretty>

AppCode

“AppCode definitely empowers TDD.
What I didn't get until I saw someone's
screencast is to really lean on Extract
Variable to reduce typing.”

Jon Reid

reduce typing

Resources & Contact

Code Examples

github.com/mobile-academy/ios-tdd-lodz

Contact

[@eldudi](https://twitter.com/eldudi)

[@alekzubala](https://twitter.com/alekzubala)

pawel@dudek.mobi

alek@zubala.com