ORDINA

UNIT TESTING IN ANGULARJS

Dennis Jaamann — Frédéric Ghijselinck
@dennisjaamann — @f_ghijselinck

CC Front-end & UX

UNIT TESTING IN ANGULARJS

- Unit testing JavaScript
- Karma
- Unit testing frameworks
  - QUnit
  - Mocha
  - Jasmine
- Unit testing AngularJS
  a. Controller
  b. Service
  c. Directive
- Coverage reports

UNIT TESTING JAVASCRIPT

- Testing JavaScript is hard
- Mixed with HTML
- Inline scripts
- No real classes
- No real modules
- Feels like wrestling a king croc

UNIT TESTING JAVASCRIPT - PITFALLS

- DOM interactions
- AJAX
- Event handlers
- Timeouts / Intervals
- Promises
- Basically everything asynchronous

UNIT TESTING JAVASCRIPT - WHAT DO WE NEED?

- Browser
- Test Runner
- Test framework
- Assertions (matchers)
- Ways to mock
- ...

KARMA

Karma = unit test runner (by angular core team)

- configuration file
- test suites
- headless/real browser

Install karma

```
npm install karma --save-dev
```

Automatically generate a config file

```
karma init
```

KARMA

Config file

`karma.conf.js`

```
module.exports = function(config) {
    config.set({
        basePath: '../..',
        frameworks: ['jasmine'],
        autoWatch : false,
        browsers : [ 'PhantomJS' ]
    });
};
```

Simply run the test suite

```
karma start
```

PHANTOMJS

- Headless browser
- No GUI
- Tests run faster
- Only suited for unit tests, not integration tests

PHANTOMJS KARMA CONFIGURATION

Install PhantomJS launcher

```
npm install karma-phantomjs-launcher --save-dev
```

Enable PhantomJS plugin

```
plugins : ['karma-jasmine', 'karma-phantomjs-launcher']
```

Configure your browser

```
module.exports = function(config) {
    config.set({
        browsers : ['PhantomJS']
    });
};
```

UNIT TESTING FRAMEWORKS

- Several options
    - Qunit
    - Mocha
    - Jasmine
- All similar, choose your own flavor
- Karma supports all

QUNIT

A minimal QUnit test setup:

```
<html>
<head>
    <meta charset="utf-8">
    <title>QUnit Example</title>
    <link rel="stylesheet" href="//code.jquery.com/qunit/qunit-1.18.0.css">
</head>
<body>
    <div id="qunit"></div>
    <div id="qunit-fixture"></div>
    <script src="//code.jquery.com/qunit/qunit-1.18.0.js"></script>
    <script src="tests.js"></script>
</body>
</html>
```

QUNIT

The contents of tests.js

```
QUnit.module( "group a" );
QUnit.test( "a basic test example", function( assert ) {
  assert.ok( true, "true succeeds" );
});
QUnit.test( "notOk test", function( assert ) {
  assert.notOk( false, "false succeeds" );
});

QUnit.module( "group b" );
QUnit.test( "hello test", function( assert ) {
    assert.ok( 1 == "1", "Passed!" );
});
```

MOCHA

An example test

```
var assert = require("assert")
    describe('Array', function(){
        describe('#indexOf()', function(){
            it('should return -1 when the value is not present', function(){
                assert.equal(-1, [1,2,3].indexOf(5));
                assert.equal(-1, [1,2,3].indexOf(0));
            })
        })
})
```

JASMINE

- test suite with specs:

```
describe("A suite", function() {
    it("contains spec with an expectation", function() {
        expect(true).toBe(true);
    });
});
```

```
describe("A suite is just a function", function() {
    var a;
    it("and so is a spec", function() {
        a = true;
        expect(a).toBe(true);
    });
});
```

JASMINE

setup & teardown

global functions:

- beforeEach()
- afterEach()
- beforeAll()
- afterAll()

JASMINE - MATCHERS

- expect().toBe();
- expect().toEqual();
- expect().toMatch();
- expect().toBeDefined();
- expect().toBeUnDefined();
- expect().toBeNull();
- expect().toBeTruthy();
- expect().toBeFalsy();
- expect().toContain();
- expect().toBeLessThan();
- expect().toBeGreaterThan();
- expect().toBeCloseTo();
- expect().toThrow();

JASMINE - SPIES

```
spyOn(foo, 'setBar');
```

```
it("tracks that the spy was called", function() {
    expect(foo.setBar).toHaveBeenCalled();
});
```

- and.callThrough
- and.returnValue
- and.callFake
- and.throwError
- and.stub

JASMINE - SPIES

```
describe("A spy", function() {
    var foo, bar = null;

    beforeEach(function() {
        foo = {
            setBar: function(value) {
                bar = value;
            }
        };

        spyOn(foo, 'setBar');

        foo.setBar(123);
        foo.setBar(456, 'another param');
    });

    it("tracks that the spy was called", function() {
        expect(foo.setBar).toHaveBeenCalled();
    });

    it("tracks all the arguments of its calls", function() {
        expect(foo.setBar).toHaveBeenCalledWith(123);
        expect(foo.setBar).toHaveBeenCalledWith(456, 'another param');
    });
});
```

JASMINE - DISABLE

disabling suites with *xdescribe*

```
xdescribe("A disabled suite", function() {
    it("with a spec", function() {
        expect(true).toBe(true);
    });
});
```

disabling specs with *xit*

```
describe("A suite", function() {
    xit("with a disabled spec", function() {
        expect(true).toBe(true);
    });
});
```

JASMINE - EXCLUSIVE TESTS

run specific suites with *ddescribe*

```
ddescribe("An exclusive run suite", function() {
    it("with a spec", function() {
        expect(true).toBe(true);
    });
});
```

run specific specs with *iit*

```
describe("A suite", function() {
    iit("with a exclusive run spec", function() {
        expect(true).toBe(true);
    });
});
```

UNIT TESTING ANGULARJS

- Angular = separation of concerns
- Create highly cohesive, low coupled pieces of functionality
- Easier to test

UNIT TESTING ANGULARJS

- getting the module

```
beforeEach(module('app'));
```

- injecting the controller

```
beforeEach(inject(function($controller, $rootScope) {
    controller = $controller;
    scope = $rootscope.$new();
}));
```

UNIT TESTING ANGULARJS - CONTROLLER

```
'use strict';

anguler.module('app')
    .controller('FakeController', function($scope, someRecords) {
    $scope.someRecords = someRecords;
});
```

UNIT TESTING ANGULARJS - CONTROLLER

```
describe("FakeController", function() {
    var someRecords;

    beforeEach(module('app'));

    beforeEach(inject(function($rootScope, $controller) {
        $scope = $rootScope.$new();
        givenSomeRecords();
        dependencies = {
            $scope : $scope,
            someRecords : someRecords
        };
        $controller('FakeController', dependencies);
    });

    it('when initialized', function() {
        thenSomeRecordsAreOnScope();
    });

    function givenSomeRecords() {
        someRecords = {
            test : 'test'
        };
    };

    function thenSomeRecordsAreOnScope() {
        expect($scope.someRecords).toEqual(someRecords);
    }
});
```

UNIT TESTING ANGULARJS - SERVICE

```
'use strict';

    anguler.module('app')
        .service('FakeService', function($http) {

    this.getIsFake = function() {
        return $http.get('fakeService/isFake');
    };

});
```

UNIT TESTING ANGULARJS - SERVICE

```
describe('FakeService', function() {
    var resolved;

    beforeEach(module('app'));

    beforeEach(inject(function(_$httpBackend_, _FakeService_) {
        $httpBackend = _$httpBackend_;
        FakeService = _FakeService_;
    }));

    it('Resolver returns resolved promise', function() {
        givenMockIsFake();
        whenGetIsFakeCalled();
        thenPromiseIsResolved();
    });

    function givenMockIsFake() {
        $httpBackend.expectGET('fakeService/isFake').respond(200, 'true');
    }

    function whenGetIsFakeCalled() {
        FakeService.getIsFake().then(function(promise) {
            resolved = true;
        });
        $httpBackend.flush();
    }

    function thenPromiseIsResolved() {
        expect(resolved).toBeTruthy();
    }
});
```

UNIT TESTING ANGULARJS - DIRECTIVE

```
'use strict';

anguler.module('app').directive('fixedPhoneNumberFormat', function() {
    return {
        scope : {
            fixedPhoneNumberFormat : '@'
        },
        link : function(scope, element, attrs) {
            attrs.$observe('fixedPhoneNumberFormat', function(fixedPhoneNumber)
                if (_.isEqual(fixedPhoneNumber.length, 8){
                    fixedPhoneNumber = "0" + fixedPhoneNumber;
                }
                element.text(fixedPhoneNumber.substring(0, 2)
                    + " / " + fixedPhoneNumber.substring(2, 5)
                    + " " + fixedPhoneNumber.substring(5, 7)
                    + " " + fixedPhoneNumber.substring(7, 9));
            });
        }
    };
});
```

UNIT TESTING ANGULARJS - DIRECTIVE

```
describe('FixedPhoneNumberDirective', function() {
    var element = {};
    var formattedFixedPhoneNumber;

    beforeEach(module('app'));

    beforeEach(inject(function($rootScope, _$compile_) {
        $scope = $rootScope.$new();
        $compile = _$compile_;
    }));

    it("formatfixedphonewithcode", function() {
        givenTemplate();
        $scope.fixedPhoneNumber = "025021910"; //givenFixedPhoneNumberWithNineDigits();
        formattedFixedPhoneNumber = "02 / 502 19 10"; //givenFormatFixedPhoneNumber();
        whenFormatFixedPhoneNumber();
        thenFixedPhoneNumberIsFormatted();
    });

    function givenTemplate() {
        var template = '<div data-fixed-phone-number-format="{{fixedPoneNumber}}">{{fixedPhoneNum
        element = $compile(template)($scope);
    }

    function whenFormatFixedPhoneNumber() {
        $scope.$digest();
    }

    function thenFixedPhoneNumberIsFormatted() {
        expect(element.text()).toBe(formattedFixedPhoneNumber);
    }
});
});
```

COVERAGE REPORTS

- Karma can generate coverage reports
- Uses istanbul.js behind the scenes
- Multiple report types
  - HTML
  - LCOV
  - Text
  - Cobertura

KARMA COVERAGE CONFIGURATION

- Add a preprocessor

```
preprocessors = {'**/lib/*.js': 'coverage'};
```

- Add a reporter

```
reporters = ['coverage'];
```

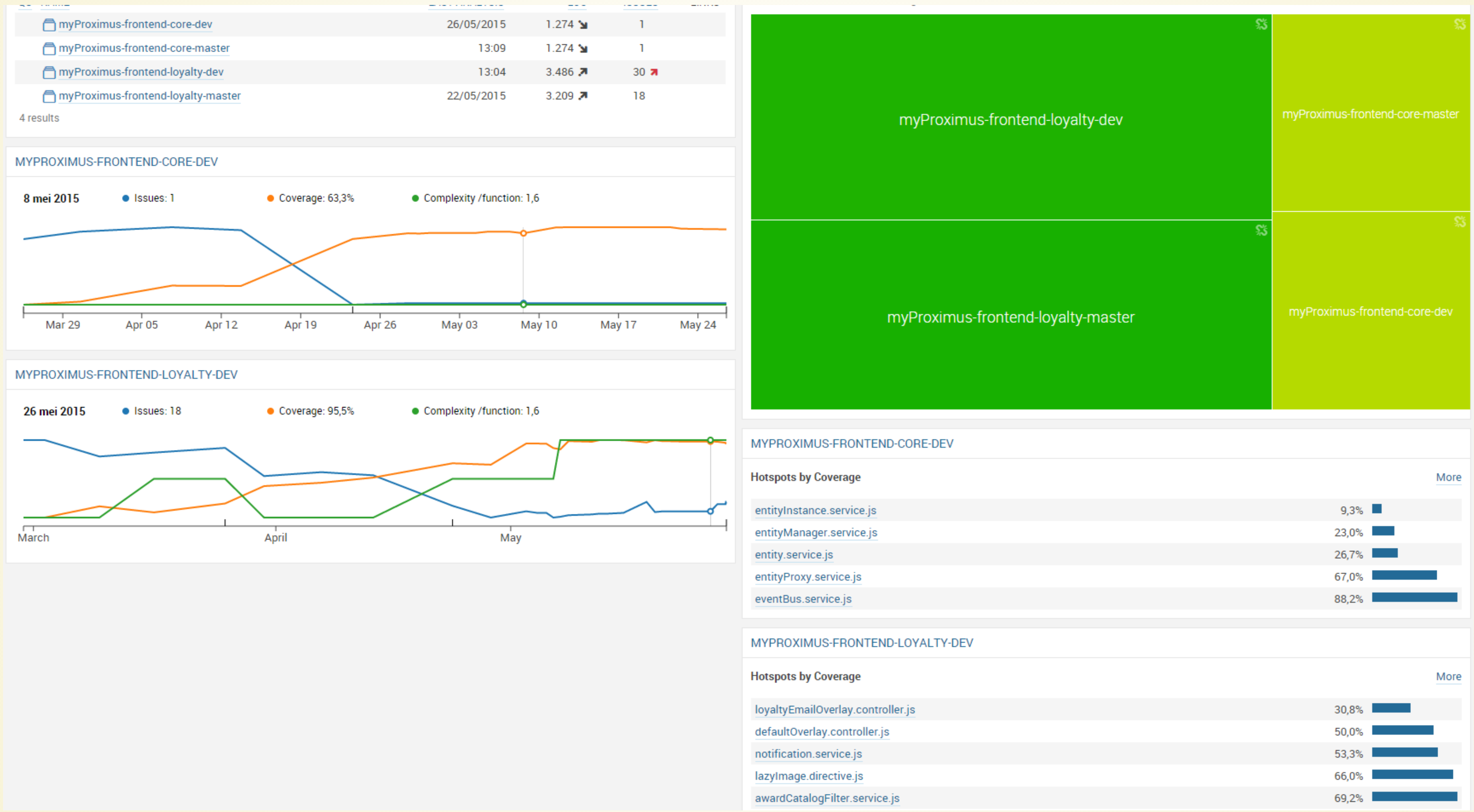
- Configure the reporter

```
coverageReporter = {type : 'lcovonly',dir : 'coverage/'}
```

LCOV FILE EXAMPLE

```
TN:
SF:../app/scripts/app.js
FN:20,(anonymous_1)
FNF:1
FNH:1
FNDA:3,(anonymous_1)
DA:11,1
DA:21,3
LF:2
LH:2
BRF:0
BRH:0
...
```

LCOV SONAR INTEGRATION



myProximus-frontend-core-dev	26/05/2015	1.274 ↘	1
myProximus-frontend-core-master	13:09	1.274 ↘	1
myProximus-frontend-loyalty-dev	13:04	3.486 ↗	30 ↗
myProximus-frontend-loyalty-master	22/05/2015	3.209 ↗	18

4 results

MYPROXIMUS-FRONTEND-CORE-DEV

8 mei 2015 ● Issues: 1 ● Coverage: 63,3% ● Complexity /function: 1,6

MYPROXIMUS-FRONTEND-LOYALTY-DEV

26 mei 2015 ● Issues: 18 ● Coverage: 95,5% ● Complexity /function: 1,6

March
April
May
myProximus-frontend-loyalty-dev
myProximus-frontend-core-master
myProximus-frontend-loyalty-master
myProximus-frontend-core-dev

MYPROXIMUS-FRONTEND-CORE-DEV

Hotspots by Coverage More

entityInstance.service.js	9,3%
entityManager.service.js	23,0%
entity.service.js	26,7%
entityProxy.service.js	67,0%
eventBus.service.js	88,2%

MYPROXIMUS-FRONTEND-LOYALTY-DEV

Hotspots by Coverage More

loyaltyEmailOverlay.controller.js	30,8%
defaultOverlay.controller.js	50,0%
notification.service.js	53,3%
lazyImage.directive.js	66,0%
awardCatalogFilter.service.js	69,2%

RESOURCES

- AngularJS website
- QUnit website
- Mocha website
- Jasmine website
- Karma website
- Istanbul website
- SinonJS website
- Sonar website

THAAAAAAAANKS!

Dennis Jaamann — Frédéric Ghijselinck

@dennisjaamann — @f_ghijselinck

CC Front-end & UX