AngularJS: Do it right

Eugene Zharkov

“You can put down a bad book; you can avoid listening to bad music; but you cannot miss the ugly structure of your project”

–Renzo Piano

1. What is basic structure of any project at the beginning ?

angular-seed or any angular example

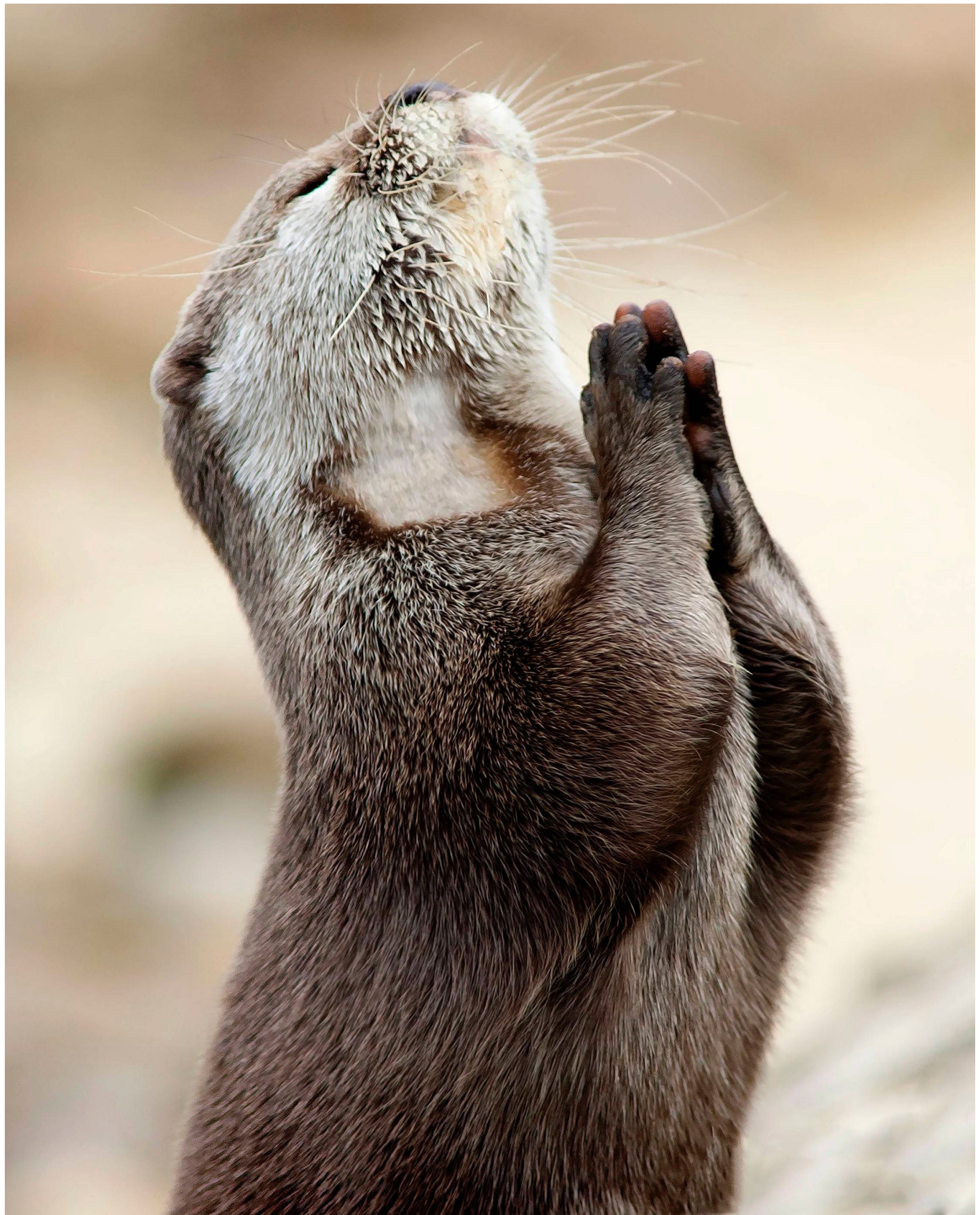
- app/ – all of the files to be used in production
    - css/ – css files
    - img/ – image files
    - index.html – app layout file (the main html template file of the app)
    - js/ – javascript files
        - app.js – application
        - controllers.js – application controllers
        - directives.js – application directives
        - filters.js – custom angular filters
        - services.js – custom angular services
    - lib/ – angular and 3rd party javascript libraries
    - partials/ – angular view partials (partial html templates)
- config/ – config files for running unit tests with Testacular/Karma
- scripts/ – handy shell/js/ruby scripts (run unit tests and dev server)
- test/ – test source files and libraries

The basic shop’s logic

- Login (Twitter, Facebook, Google)
- Registration
- Home
- Widgets
- Shopping Cart
- Search
- Product
- Payment (PayPal, Credit Card..)

- controllers/
    - LoginController.js
    - RegistrationController.js
    - HomeController.js
    - ShoppingCartController.js
    - SearchController.js
    - ProductController.js
    - PaymentController.js

Don’t do that



ng-boilerplate

```
ng-boilerplate/
  |- grunt-tasks/
  |- karma/
  |- src/
  |  |- app/
  |  |  |- <app logic>
  |  |- assets/
  |  |  |- <static files>
  |  |- common/
  |  |  |- <reusable code>
  |  |- less/
  |  |  |- main.less
  |- vendor/
  |  |- angular-bootstrap/
  |  |- bootstrap/
  |  |- placeholders/
  |- .bowerrc
```

FEATURE-BASED MODULARIZATION

- src/
  - app/
    - about/
      - about.js
      - about.html
    - home/
      - **home.js**
      - **home.less**
      - **home.spec.js**
      - **home.html**
  - app.js
  - app.spec.js
- assets/
- components/
- less/
- index.html

Keep going

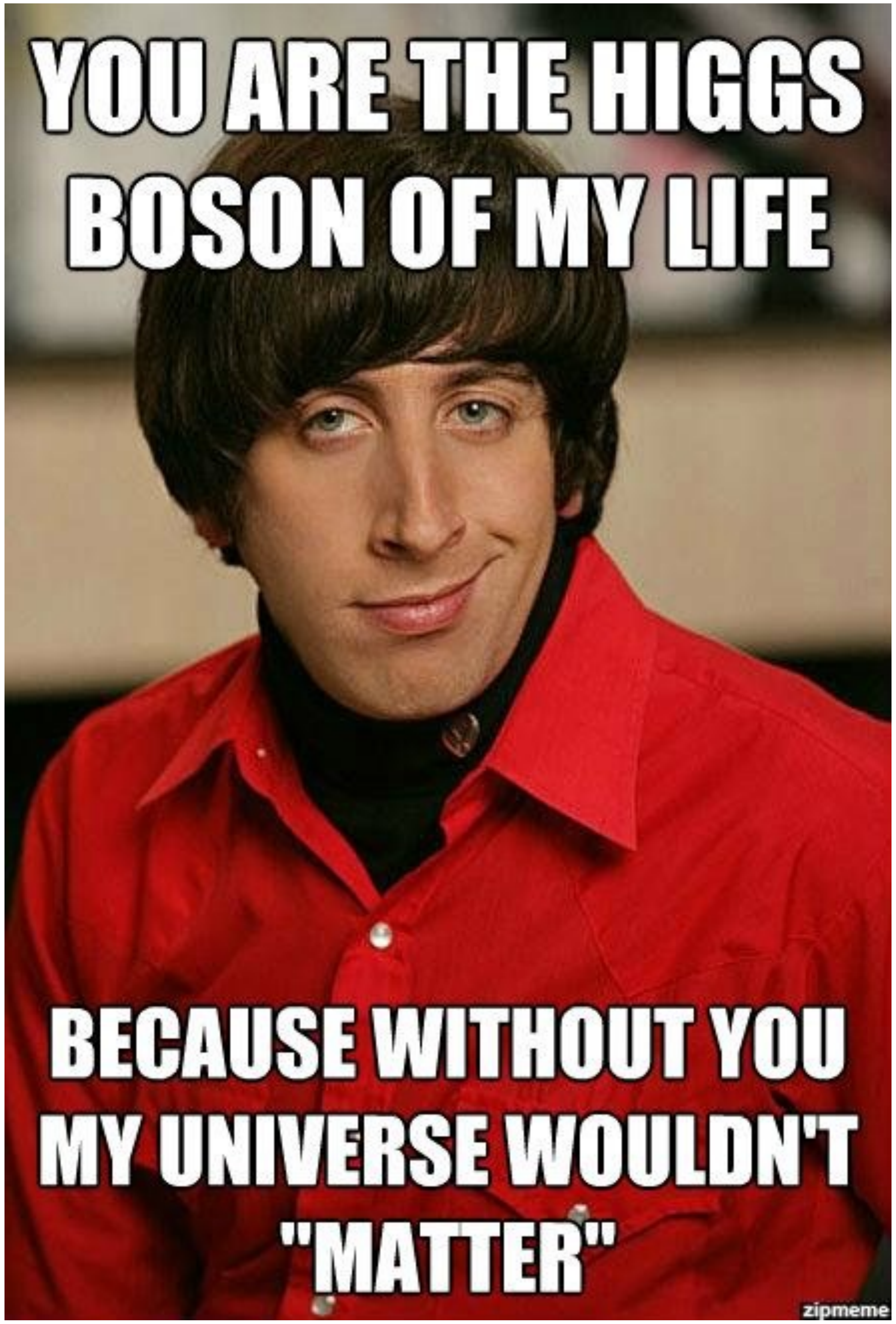
“If you think you can’t make the project better with your work, at least make sure you don’t make it worse.”

–Herman Hertzberger

2. Build & Automate

- Gulp
- Grunt
- Brunch.io
- make?
- PowerShell ヾ(@°▽°@)ノ

jQuery


YOU ARE THE HIGGS BOSON OF MY LIFE
BECAUSE WITHOUT YOU MY UNIVERSE WOULDN'T "MATTER"
zipmeme

How about to avoid using jQuery…for awhile?

Is your component complex?
OK.
Use jQuery.

Do you want to:

- Show text
- Show/hide blocks
- Replace content
- Handle event

to jQuery

THE KISS PRINCIPLE

KEEP
IT
SIMPLE,
STUPID

- angular.copy
- angular.element
- angular.equals
- angular.extend
- angular.forEach
- angular.fromJson
- angular.identity
- angular.isArray
- angular.isDate
- angular.isDefined
- angular.isElement
- angular.isFunction
- angular.isObject
- angular.isString
- angular.isUndefined
- angular.lowercase
- angular.noop
- angular.toJson
- angular.uppercase

Libraries

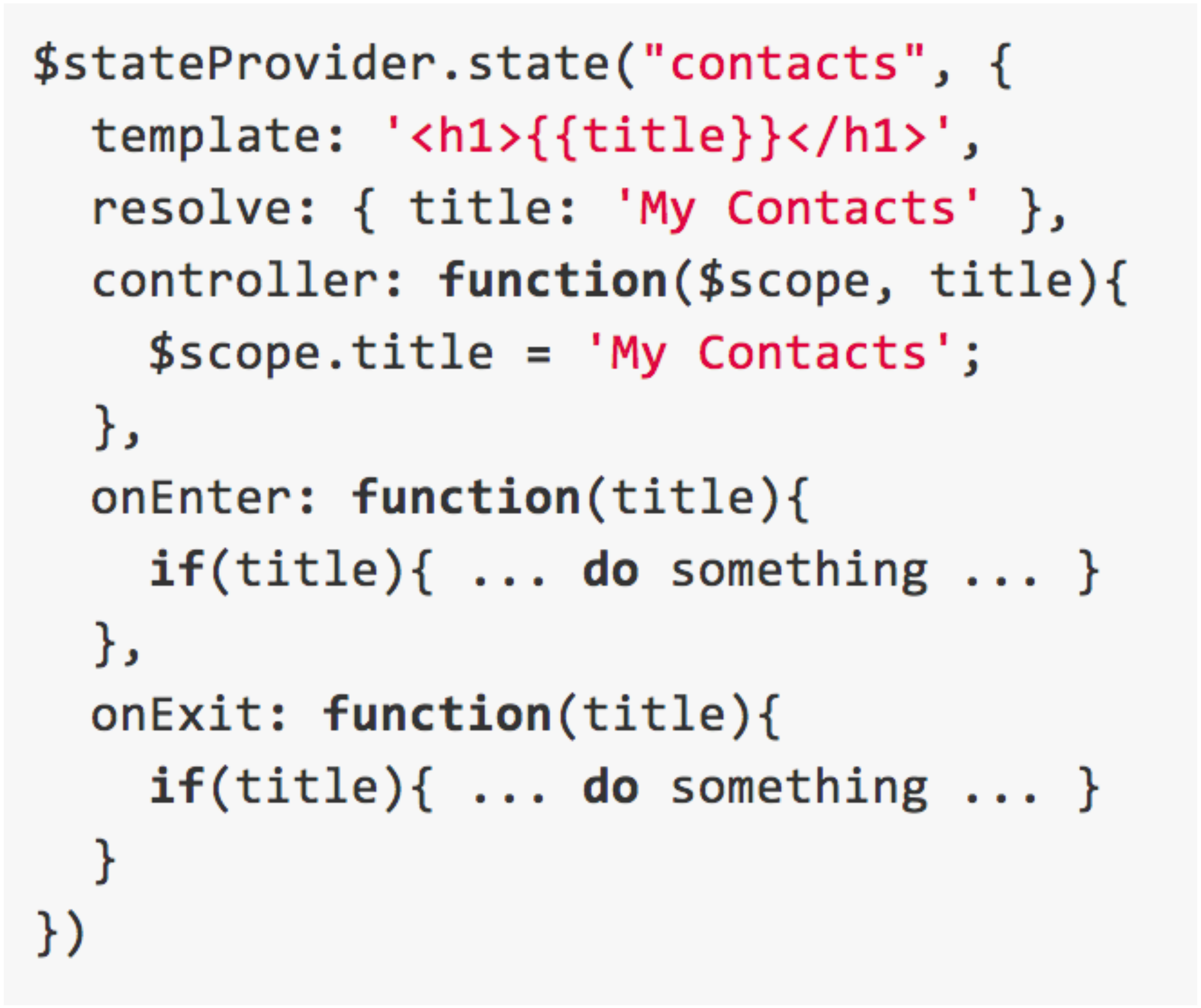
- Twitter Bootstrap
- AngularUI-Bootstrap
- AngularUI
- UI-Router
- Restangular

UI-Router

```
$stateProvider
  .state('state1', {
    url: "/state1",
    templateUrl: "partials/state1.html"
  })
  .state('state1.list', {
    url: "/list",
    templateUrl: "partials/state1.list.html",
    controller: function($scope) {
      $scope.items = ["A", "List", "Of", "Items"];
    }
  })
  .state('state2', {
    url: "/state2",
    templateUrl: "partials/state2.html"
  })
  .state('state2.list', {
    url: "/list",
      templateUrl: "partials/state2.list.html",
      controller: function($scope) {
        $scope.things = ["A", "Set", "Of", "Things"];
      }
    })
  });
```

```
myApp.config(function($stateProvider) {
  $stateProvider
    .state('index', {
      url: "",
      views: {
        "viewA": { template: "index.viewA" },
        "viewB": { template: "index.viewB" }
      }
    })
    .state('route1', {
      url: "/route1",
      views: {
        "viewA": { template: "route1.viewA" },
        "viewB": { template: "route1.viewB" }
      }
    })
    .state('route2', {
      url: "/route2",
      views: {
        "viewA": { template: "route2.viewA" },
        "viewB": { template: "route2.viewB" }
      }
    })
});
```

UI-Router



```
$stateProvider.state("contacts", {
  template: '<h1>{{title}}</h1>',
  resolve: { title: 'My Contacts' },
  controller: function($scope, title){
    $scope.title = 'My Contacts';
  },
  onEnter: function(title){
    if(title){ ... do something ... }
  },
  onExit: function(title){
    if(title){ ... do something ... }
  }
})
```

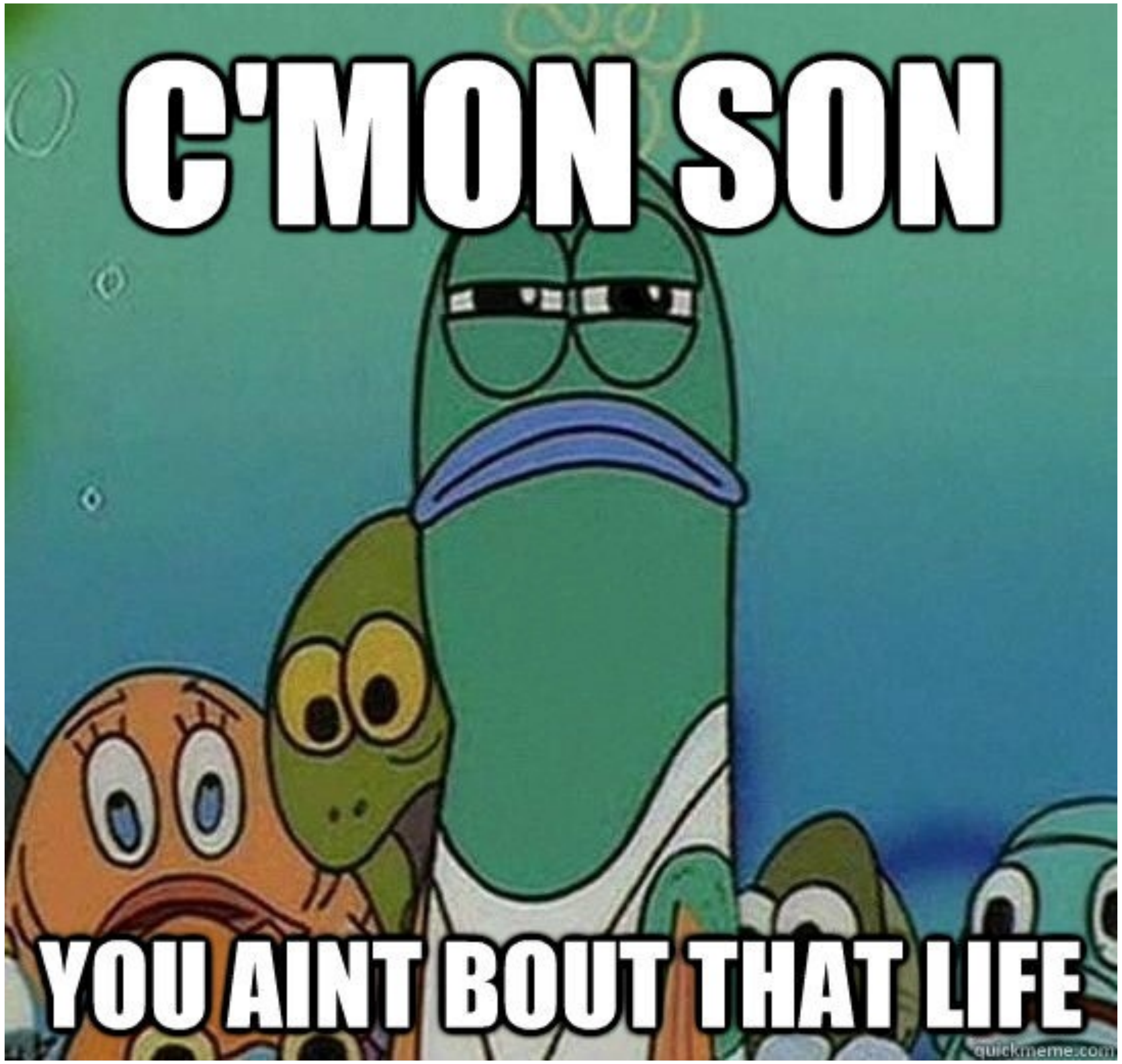
Restangular

```
var restangualrSpaces = Restangular.one("accounts",123).one("buildings", 456).all("spaces");

// This will do ONE get to /accounts/123/buildings/456/spaces
restangularSpaces.getList()

// This will do ONE get to /accounts/123/buildings/456/spaces/789
Restangular.one("accounts", 123).one("buildings", 456).one("spaces", 789).get()

// POST /accounts/123/buildings/456/spaces
Restangular.one("accounts", 123).one("buildings", 456).all("spaces").post({name: "New Space"});

// DELETE /accounts/123/buildings/456
Restangular.one("accounts", 123).one("buildings", 456).remove();
```

Directives, it’s easy

```
var myModule = angular.module(...);
myModule.directive('directiveName', function (injectables) {
  return {
    restrict: 'A',
    template: '<div></div>',
    templateUrl: 'directive.html',
    replace: false,
    priority: 0,
    transclude: false,
    scope: false,
    terminal: false,
    require: false,
    controller: function($scope, $element, $attrs, $transclude, otherInjectables) { ... },
    compile: function compile(tElement, tAttrs, transclude) {
      return {
        pre: function preLink(scope, iElement, iAttrs, controller) { ... },
        post: function postLink(scope, iElement, iAttrs, controller) { ... }
      }
    },
    link: function postLink(scope, iElement, iAttrs) { ... }
  };
});
```

Testing



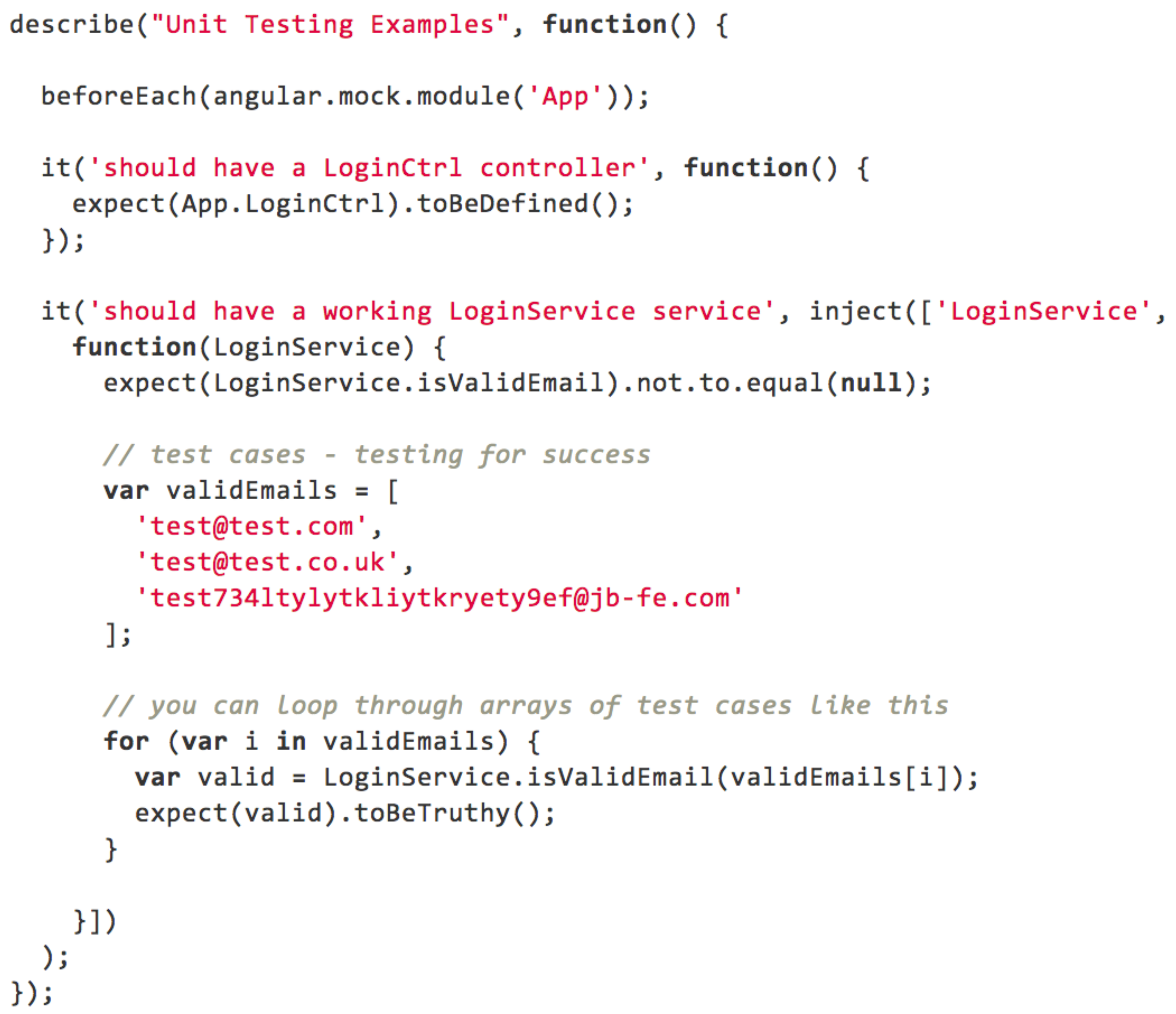
Unit Testing

- Jasmine
- Mocha
- QUniT

Integration Testing

- Karma
- CasperJS

Unit Tests



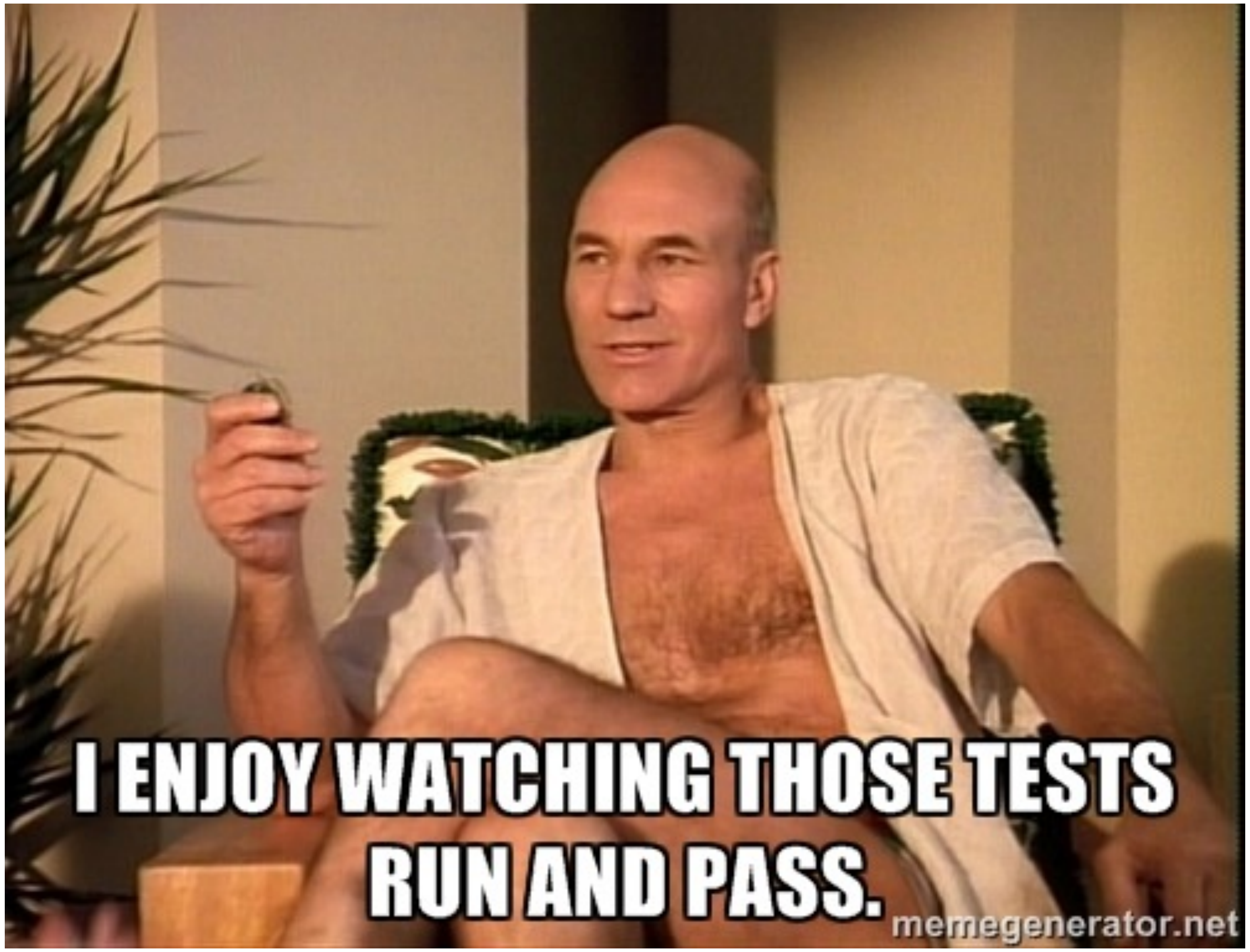
```
describe("Unit Testing Examples", function() {

  beforeEach(angular.mock.module('App'));

  it('should have a LoginCtrl controller', function() {
    expect(App.LoginCtrl).toBeDefined();
  });

  it('should have a working LoginService service', inject(['LoginService',
    function(LoginService) {
      expect(LoginService.isValidEmail).not.to.equal(null);

      // test cases - testing for success
      var validEmails = [
        'test@test.com',
        'test@test.co.uk',
        'test734ltylytkliytkryety9ef@jb-fe.com'
      ];

      // you can loop through arrays of test cases like this
      for (var i in validEmails) {
        var valid = LoginService.isValidEmail(validEmails[i]);
        expect(valid).toBeTruthy();
      }

    }])
  );
});
```

Integration Tests

```
describe("Integration/E2E Testing", function() {

  // start at root before every test is run
  beforeEach(function() {
    browser().navigateTo('/');
  });

  // test default route
  it('should jump to the /home path when / is accessed', function() {
    browser().navigateTo('#/');
    expect(browser().location().path()).toBe("/login");
  });

  it('ensures user can log in', function() {
    browser().navigateTo('#/login');
    expect(browser().location().path()).toBe("/login");

    // assuming inputs have ng-model specified, and this conbination will successfully login
    input('email').enter('test@test.com');
    input('password').enter('password');
    element('submit').click();

    // logged in route
    expect(browser().location().path()).toBe("/dashboard");

    // my dashboard page has a label for the email address of the logged in user
    expect(element('#email').html()).toContain('test@test.com');
  });
```

Mocking Services

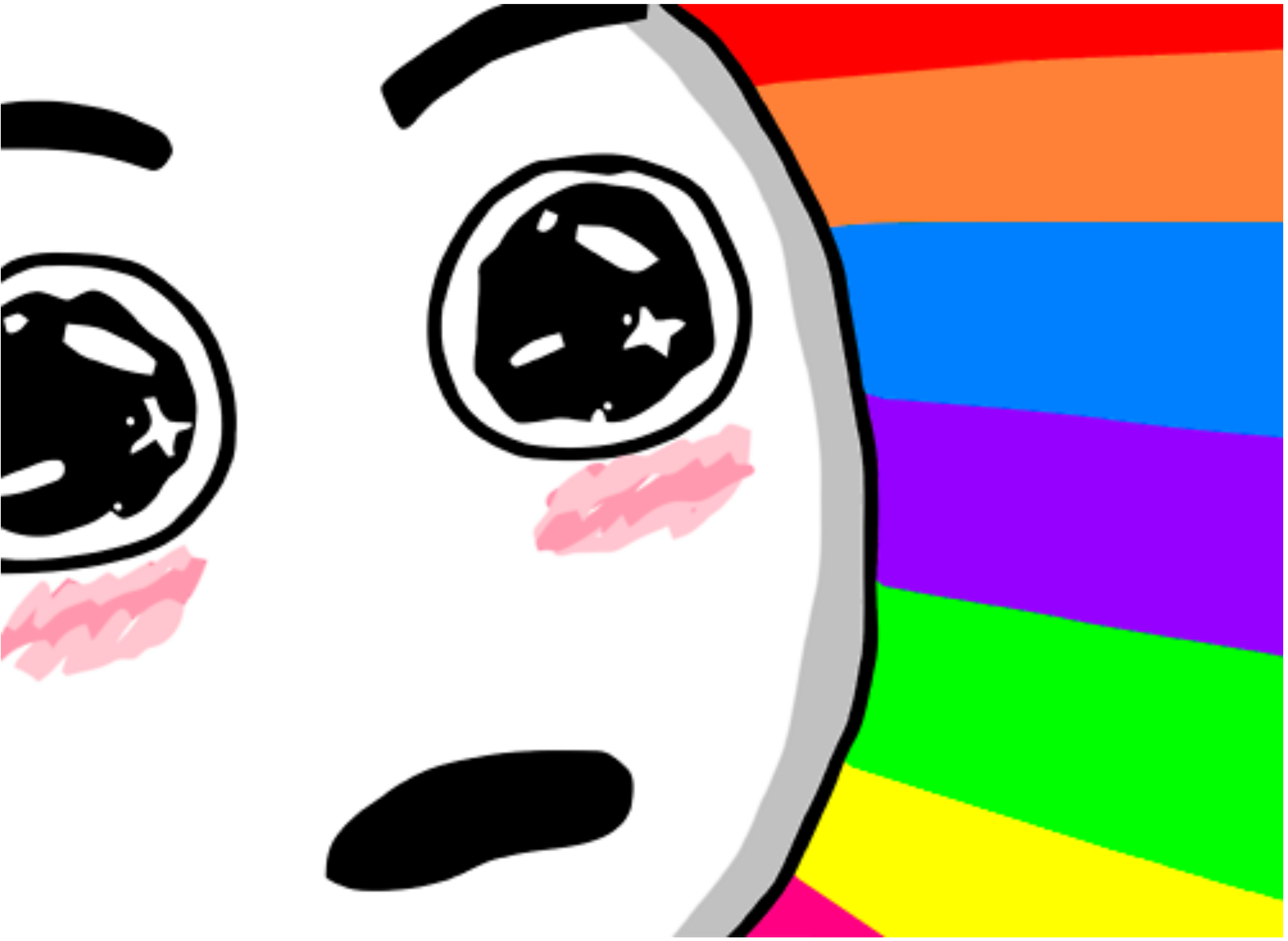
```
it('should get login success',
  inject(function(LoginService, $httpBackend) {

    $httpBackend.expect('POST', 'https://api.mydomain.com/login')
      .respond(200, "[{ success : 'true', id : 123 }]");

    LoginService.login('test@test.com', 'password')
      .then(function(data) {
        expect(data.success).toBeTruthy();
    });

  $httpBackend.flush();
});
```


I ENJOY WATCHING THOSE TESTS RUN AND PASS.
memegenerator.net

When a developer is asked what his best project is, he usually answers, “The next one.”

–Emilio Ambasz

Your application <u>now</u>



It’s…

…beautiful

Questions?
Need a help with your project?
Ping me.

@2j2e

@2j2e

eu.zharkov@gmail.com