

Testing in AngularJS

1

2

3 `expect(unforeseenProblem).toBeDefined()`

4

5



1

2

3

```
expect(unforeseenProblem).toBeDefined()
```

4

5

What's out there

Testing Frameworks



```
1 describe("a functionality", function() {  
2     it("has some behaviours", function() {  
3         expect(expectedBehaviour).toBe(true);  
4     });  
5 });
```

Test Runners



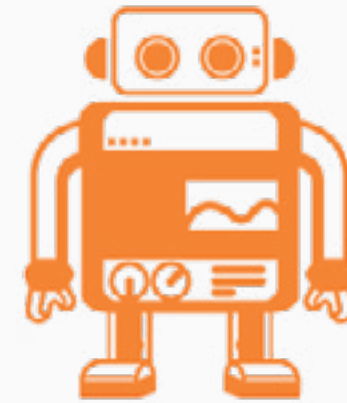
```
1 INFO [karma]: Karma v0.10.9 server started at http://localhost:8080/
2 INFO [Chrome 34.0.1847 (Mac OS X 10.9.2)]: Connected on socket qSa4UuBikfbk9GKFwKVA
3 Chrome 34.0.1847 (Mac OS X 10.9.2): Executed 6 of 6 SUCCESS (3.403 secs / 3.119 secs)
4
5 -----
6
7 Using the selenium server at http://localhost:4444/wd/hub
8 .
9
10 Finished in 0.019 seconds
11 1 test, 1 assertion, 0 failures
```

Test Runners



```
1 1) Sign up allows you to become a new user.  
2   Message:  
3     Expected true to be false.  
4   Stacktrace:  
5     Error: Expected true to be false.  
6     at null.<anonymous> (/.../test/e2e/signup_feature.js:4:18)  
7     at /usr/local/lib/node_modules/protractor/jasminewd/index.js:54:12  
8     at wrapper [as _onTimeout] (timers.js:252:14)  
9     at Timer.listOnTimeout [as ontimeout] (timers.js:110:15)  
10  
11 Finished in 0.02 seconds  
12 1 test, 1 assertion, 1 failure
```


Underlying Technologies



WEBDRIVERJS



Jasmine

```
1 describe("a functionality", function() {  
2     it("has some behaviours", function() {  
3         expect(expectedBehaviour).toBe(true);  
4     });  
5 });
```



```
1 expect(a).toBe(b);
2 expect(a).not.toBe(null);
3 expect({id: 5}).toEqual({id: 5});
4 expect(message).toMatch(/bar/);
5 expect(message).toMatch("bar");
6 expect(a.method).toBeDefined();
7 expect(null).toBeNull();
8 expect(true).toBeTruthy();
9 expect(false).toBeFalsy();
10 expect([1, 2, 3]).toContain(2);
11 expect(1).toBeLessThan(2);
12 expect(2).toBeGreaterThan(2);
13 expect(function).toThrow();
```

```
1 describe('Counter', function () {  
2  
3     beforeEach(function(){  
4         this.instance = new Counter(0);  
5     });  
6  
7     it('does something', function () {  
8         this.instance.set(1);  
9     });  
10  
11    it('does something else', function () {  
12        this.instance.increase();  
13    });  
14  
15    afterEach(function(){  
16        expect(this.instance.get()).toBe(1);  
17    });  
18 });
```

```
1 describe("A user", function() {
2
3     describe("with membership", function(){
4
5         beforeEach(function() {
6             this.user = new User({membership: true});
7         });
8
9         it ("can access the service", function(){
10             expect(this.user.hasAccess()).toBeTruthy();
11         });
12
13     });
14
15     describe("without membership", function(){
16
17         beforeEach(function() {
18             this.user = new User({membership: false});
19         });
20
21         it ("doesn't have access", function(){
22             expect(this.user.hasAccess()).toBeFalsy();
23         });
24
25     });
26
27 });
```


Asynchronous Support in Jasmine 1.3

```
1 runs(function() {  
2     setTimeout(function() {  
3         this.value = 5;  
4     }, 500);  
5 });  
6  
7 waitsFor(function() {  
8     return this.value !== undefined;  
9 }, "Value should be defined", 1000);  
10  
11 runs(function() {  
12     expect(this.value).toBe(5);  
13 });
```

Asynchronous Support in Jasmine 2.0

```
1 it('', function(done){  
2   setTimeout(function(){  
3     this.value = 5;  
4     done();  
5   });  
6  
7   it('', function(){  
8     expect(this.value).toBe(5);  
9   });  
10 });
```

Asynchronous Support in Jasmine 2.0

```
1 beforeEach(function(done){  
2     asyncOperation(this.obj).then(done);  
3 });  
4  
5 it('has a certain value', function() {  
6     expect(this.obj.value).toEqual(123);  
7 });
```



```
1 var login = function(){...};  
2  
3 beforeEach(function(done){  
4     this.user = new User();  
5     login(user);  
6 });
```

```
1 jasmine.addMatchers(  
2   {'toBeMultipleOf': function() {  
3     return {  
4       compare: function(actual, expected){  
5         var result = {pass: (actual % expected) === 0};  
6         result.message = result.pass ? '...' : '...';  
7         return result;  
8       }  
9     };  
10  }  
11 });  
12  
13 describe('', function(){  
14   it('', function(){  
15     expect(15).toBeMultipleOf(5);  
16   });  
17 });
```

Karma

- 1 `npm install karma --save-dev`
- 2 `npm install karma-jasmine --save-dev`
- 3 `npm install karma-chrome-launcher --save-dev`
- 4 `npm install -g karma-cli`

```
>karma init karma.conf.js
```

Which testing framework do you want to use ?

Press tab to list possible options. Enter to move to the next question.

```
> jasmine
```

Do you want to use Require.js ?

This will add Require.js plugin.

Press tab to list possible options. Enter to move to the next question.

```
> no
```

Do you want to capture any browsers automatically ?

Press tab to list possible options. Enter empty string to move to the next question.

```
> Chrome
```

```
>
```

What is the location of your source and test files ?

You can use glob patterns, eg. "js/*.js" or "test/**/*.Spec.js".

Enter empty string to move to the next question.

```
>
```

Should any of the files included by the previous patterns be excluded ?

You can use glob patterns, eg. "**/*.swp".

Enter empty string to move to the next question.

```
>
```

Do you want Karma to watch all the files and run the tests on change ?

Press tab to list possible options.

```
> yes
```

Config file generated at "/Users/Sebastian/Work/Eigenes/Testing in AngularJS/Application/karma.conf.js".

```
>karma start
```

```
INFO [karma]: Karma v0.12.14 server started at http://localhost:9876/
```

```
INFO [launcher]: Starting browser Chrome
```

```
INFO [Chrome 34.0.1847 (Mac OS X 10.9.2)]: Connected on socket M2EJ6mpKEQlxjrbeF8dg  
with id 36658312
```

```
Chrome 34.0.1847 (Mac OS X 10.9.2): Executed 1 of 1 SUCCESS (0 secs / 0.001 secsChr
```

```
ome 34.0.1847 (Mac OS X 10.9.2): Executed 1 of 1 SUCCESS (0.008 secs / 0.001 secs)
```


- 1 - Chrome and Chrome Canary (install karma-chrome-launcher)
- 2 - Firefox (install karma-firefox-launcher first)
- 3 - Safari (install karma-safari-launcher first)
- 4 - PhantomJS (install karma-phantomjs-launcher)
- 5 - Opera (install karma-opera-launcher first)
- 6 - IE (install karma-ie-launcher first)
- 7 - SauceLabs (install karma-sauce-launcher)
- 8 - BrowserStack (install karma-browserstack-launcher)
- 9 - many more

Protractor

```
1 npm install -g protractor
```

```
>webdriver-manager start
seleniumProcess.pid: 14792
Mai 03, 2014 2:53:18 PM org.openqa.grid.selenium.GridLauncher main
INFO: Launching a standalone server
Setting system property webdriver.chrome.driver to /usr/local/lib/node_modules/protractor/selenium/chromedriver
14:53:24.800 INFO - Java: Oracle Corporation 23.25-b01
14:53:24.800 INFO - OS: Mac OS X 10.9.2 x86_64
14:53:24.808 INFO - v2.39.0, with Core v2.39.0. Built from revision ff23eac
14:53:24.869 INFO - Default driver org.openqa.selenium.ie.InternetExplorerDriver registration is skipped: registration capabilities Capabilities [{platform=WINDOWS, ensureCleanSession=true, browserName=internet explorer, version=}] does not match with current platform: MAC
14:53:24.917 INFO - RemoteWebDriver instances should connect to: http://127.0.0.1:4444/wd/hub
14:53:24.918 INFO - Version Jetty/5.1.x
14:53:24.919 INFO - Started HttpContext[/selenium-server/driver,/selenium-server/driver]
14:53:24.919 INFO - Started HttpContext[/selenium-server,/selenium-server]
14:53:24.919 INFO - Started HttpContext[/,/]
14:53:24.970 INFO - Started org.openqa.jetty.jetty.servlet.ServletHandler@3a333bdb
14:53:24.970 INFO - Started HttpContext[/wd,/wd]
14:53:24.974 INFO - Started SocketListener on 0.0.0.0:4444
14:53:24.974 INFO - Started org.openqa.jetty.jetty.Server@a832ce5
```

```
>webdriver-manager update  
selenium standalone is up to date.  
chromedriver is up to date.
```



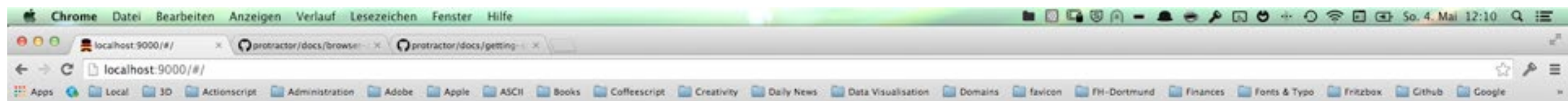
```
1 // An example configuration file.
2 exports.config = {
3   // The address of a running selenium server.
4   seleniumAddress: 'http://localhost:4444/wd/hub',
5
6   // Capabilities to be passed to the webdriver instance.
7   capabilities: {
8     'browserName': 'chrome'
9   },
10
11   // Spec patterns are relative to the location of the spec file. They may
12   // include glob patterns.
13   specs: ['example-spec.js'],
14
15   // Options to be passed to Jasmine-node.
16   jasmineNodeOpts: {
17     showColors: true, // Use colors in the command line report.
18   }
19 };
```

```
>grunt serve  
Running "serve" task
```

```
>protractor protractor.conf.js  
Using the selenium server at http://localhost:4444/wd/hub
```

```
▪  
  
Finished in 0.02 seconds  
1 test, 1 assertion, 0 failures
```

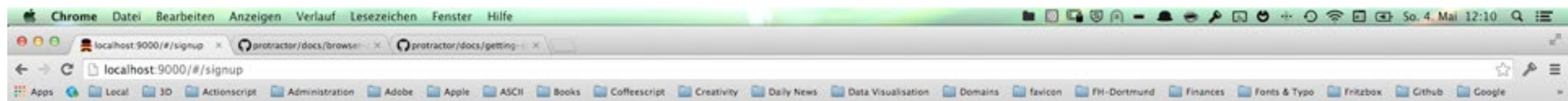
```
1 describe('Sign up', function() {
2
3     it('allows you to become a new user.', function(){
4         browser.get('http://localhost:9000');
5         element(by.css('a[href*="signup"]')).click();
6         var user = userFactory.create();
7         element(by.model('user.name')).sendKeys(user.name);
8         element(by.model('user.email')).sendKeys(user.email);
9         element(by.model('user.password')).sendKeys(user.password);
10        element(by.css('button[type="submit"]')).click();
11        expect(element(by.css('.header .nav')).getText()).toMatch(user.name);
12        expect(element(by.css('body')).getText()).toMatch('Thank you for sign up '+user.name);
13    });
14
15 });
```

Testing in AngularJS

[Home](#)[Signup](#)

Why testing
Whats out there
Jasmine
Karma
Protractor
Mocking
The Cycle of Delivery

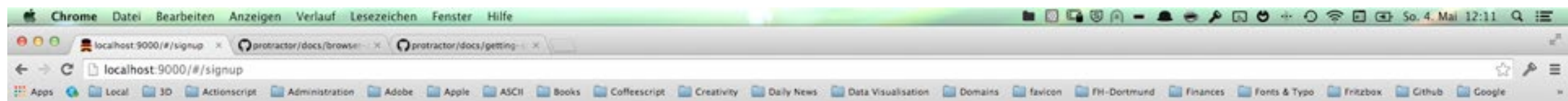


Testing in AngularJS

[Home](#)[Signup](#)

<input type="text"/>	
<input type="text"/>	
<input type="password"/>	

[Signup now](#)

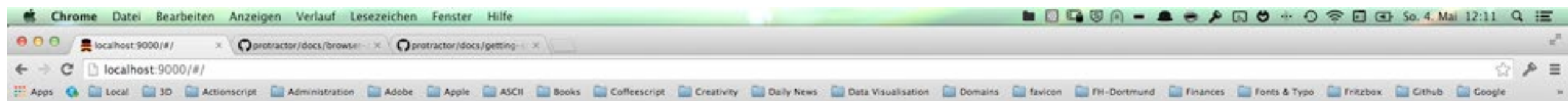


Testing in AngularJS

[Home](#)[Signup](#)

spape	
sebastian.pape@local.ch	

[Signup now](#)



Testing in AngularJS

Home spape

Thank you for sign up spape


```
1 |by.css( '.person' );
2 |by.cssContainingText( '#inner ul .pet', 'dog' );
3 |by.binding( 'person.phone' );
4 |by.model( 'person' );
5 |by.id( 'person' );
6 |by.textarea( 'aboutbox' );
7 |by.select( 'dayColor.color' );
8 |by.selectedOption( 'fruit' );
9 |by.partialButtonText( 'text' );
10 |by.buttonText( 'Exact text' );
11 |by.repeater( 'day in days' );
```

```
1 by.css( '.person' );
2 by.cssContainingText( '#inner ul .pet', 'dog' );
3 by.css( '*[ng-bind="person.phone"]' );
4 by.css( '*[ng-model="person"]' );
5 by.css( '*[id="person"]' );
6 by.css( 'textarea[name="aboutbox"]' );
7 by.css( 'select[ng-select][ng-model="dayColor.color"]' );
8 by.css( 'select[name="fruit"] option:selected' );
9 by.cssContainingText( 'button', 'text' );
10 by.buttonText( 'Exact text' );
11 by.css( '*[ng-repeat="day in days"]' );
```

```
1 element(by.css( '.class' ));
```

```
2
```

```
3 $( '.class' );
```

```
1 describe('Sign up', function() {  
2  
3     it('allows you to become a new user.', function(){  
4         browser.get('http://localhost:9000');  
5         element(by.css('a[href*="signup"]')).click();  
6         var user = userFactory.create();  
7         element(by.model('user.name')).sendKeys(user.name);  
8         element(by.model('user.email')).sendKeys(user.email);  
9         element(by.model('user.password')).sendKeys(user.password);  
10        element(by.css('button[type="submit"]')).click();  
11        expect(element(by.css('.header .nav')).getText()).toMatch(user.name);  
12        expect(element(by.css('body')).getText()).toMatch('Thank you for sign up '+user.name);  
13    });  
14  
15 });
```



```
1 describe('Sign up', function() {  
2  
3     it('allows you to become a new user.', function(){  
4         browser.get('http://localhost:9000');  
5         $('a[href*="signup"]').click();  
6         var user = userFactory.create();  
7         $('*[ng-model="user.name"]').sendKeys(user.name);  
8         $('*[ng-model="user.email"]').sendKeys(user.email);  
9         $('*[ng-model="user.password"]').sendKeys(user.password);  
10        $('button[type="submit"]').click();  
11        expect($('.header .nav').getText()).toMatch(user.name);  
12        expect($('body').getText()).toMatch('Thank you for sign up '+user.name);  
13    });  
14  
15 });
```

```
1 var SignUpPage = function(){
2   this.get = function(){
3     $('a[href*="signup"]').click();
4     this.nameInput = $('*[ng-model="user.name"]');
5     this.emailInput = $('*[ng-model="user.email"]');
6     this.passwordInput = $('*[ng-model="user.password"]');
7     this.submitButton = $('button[type="submit"]');
8   };
9   this.getHeader = function(){
10    return $('.header .nav');
11  };
12  this.setName = function(val){
13    this.nameInput.sendKeys(val);
14  };
15  this.setEmail = function(val){
16    this.emailInput.sendKeys(val);
17  };
18  this.setPassword = function(val){
19    this.passwordInput.sendKeys(val);
20  };
21  this.submit = function(){
22    this.submitButton.click();
23  };
24  this.navigationContains = function(text){
25    expect(this.getHeader().getText()).toMatch(text);
26  };
27  this.contains = function(text){
28    expect($('body').getText()).toMatch(text);
29  };
30 };
31
```

```
32▼ describe('Sign up', function() {  
33  
34▼   it('allows you to become a new user.', function(){  
35     browser.get('http://localhost:9000');  
36     var user = userFactory.create();  
37     var page = new SignUpPage();  
38     page.get();  
39     page.setName(user.name);  
40     page.setEmail(user.email);  
41     page.setPassword(user.password);  
42     page.submit();  
43     page.navigationContains(user.name);  
44     page.contains('Thank you for sign up '+user.name);  
45   });  
46  
47 });
```



```
27▼ this.contains = function(text){  
28     browser.debugger();  
29     expect($('body').getText()).toMatch(text);  
30 };
```



```
>protractor debug protractor.conf
< debugger listening on port 5858
connecting... ok
break in /usr/local/lib/node_modules/protractor/lib/cli.js:7
  5  * Values from command line options override values from the config.
  6  */
  7  'use strict';
  8
  9  // Coffee is required here to enable config files written in coffee-script.
debug> c
< Using the selenium server at http://localhost:4444/wd/hub
break in /usr/local/lib/node_modules/protractor/lib/protractor.js:735
 733  var flow = webdriver.promise.controlFlow();
 734  flow.execute(function() {
 735      debugger;
 736  });
 737 };
debug> █
```

> `window.clientSideScripts`

▼ `Object {waitForAngular: function, findBindings: function, findRepeaterElement: function...}` ⓘ

- ▶ `evaluate: function () {`
- ▶ `findAllRepeaterRows: function () {`
- ▶ `findBindings: function () {`
- ▶ `findByModel: function () {`
- ▶ `findInputs: function () {`
- ▶ `findRepeaterColumn: function () {`
- ▶ `findRepeaterElement: function () {`
- ▶ `findRepeaterRows: function () {`
- ▶ `findSelectedOptions: function () {`
- ▶ `findSelects: function () {`
- ▶ `findTextareas: function () {`
- ▶ `getLocationAbsUrl: function () {`
- ▶ `testForAngular: function () {`
- ▶ `waitForAngular: function () {`
- ▶ `__proto__: Object`

```
> window.clientSideScripts.evaluate($(' [ng-view] '), 'currentUser()')  
Object {name: "Sebastian", email: "sebastian@local.ch", password: "aPa
```

```
>node ./node_modules/protractor/bin/elementexplorer.js http://localhost:9000
Type <tab> to see a list of locator strategies.
Use the `list` helper function to find elements by strategy:
  e.g., list(by.binding('')) gets all bindings.
```

Getting page at: http://localhost:9000

element(by.id(''))	element(by.css(''))
element(by.name(''))	element(by.binding(''))
element(by.input(''))	
element(by.select(''))	element(by.textarea(''))
element(by.xpath(''))	element(by.tagName(''))
element(by.className(''))	


```
> element(by.css('body'))
{ click: [Function],
  sendKeys: [Function],
  getTagName: [Function],
  getCssValue: [Function],
  getAttribute: [Function],
  getText: [Function],
  getSize: [Function],
  getLocation: [Function],
  isEnabled: [Function],
  isSelected: [Function],
  submit: [Function],
  clear: [Function],
  isDisplayed: [Function],
  getOuterHtml: [Function],
  getInnerHtml: [Function],
  toWireValue: [Function],
  findElements: [Function],
  isElementPresent: [Function],
  evaluate: [Function],
  findElement: [Function],
  find: [Function],
  isPresent: [Function],
  element: { [Function] all: [Function] },
  '$': [Function],
  '$$': [Function] }
```

```
> list(by.css('.nav'))  
[ 'Home\nSignup' ]
```

```
43 | browser.wait(function() {  
44 |     return browser.findElement(protractor.By.css('.nav')).isDisplayed();  
45 | });
```

```
1 browser.driver.get('http://localhost:9000');
2 browser.driver.findElement(by.css('[type="submit"]')).click();
3 browser.driver.isElementPresent(by.css('.nav'))
4     .then(function(isPresent){
5         expect(isPresent).toBe(true);
6     });
```


**IT'S
A
MOCK!**



```
1▼ module.exports = function(config) {
2▼   config.set({
3     basePath: '',
4     frameworks: ['jasmine'],
5▼   files: [
6     'app/bower_components/angular-mocks/angular-mocks.js',
7     '...'
8   ],
9   exclude: [],
10  port: 8080,
11  logLevel: config.LOG_INFO,
12  autoWatch: true,
13  browsers: ['Chrome'],
14  singleRun: false
15  });
16 };
17
```

```

3 describe('lclTextEditor', function() {
4
5     beforeEach(module('lclTextEditor'));
6
7     var setup;
8     beforeEach(inject(function($rootScope, $compile){
9         setup = function(description, allowedContent, callback){
10             var scope = $rootScope;
11             allowedContent = allowedContent || '';
12             scope.description = description || '';
13             var element = $compile('<textarea lcl-text-editor="lcl-text-
14             scope.$digest();
15             waitsFor(function(){
16                 return element.data('ck') !== undefined;
17             }, "CKEditor to setup up properly", 3000);
18             runs(function(){
19                 callback({
20                     element: element,
21                     scope: scope,
22                     ck: element.data('ck')
23                 });
24             });
25         };
26     }));
27

```

```
1 'use strict';
2
3 angular.module('applicationApp')
4 .controller('SignUpCtrl', function ($scope, $location, User) {
5     $scope.user = new User();
6     $scope.submit = function(isValid){
7         $scope.formSubmitted = true;
8         if(isValid){
9             User.create($scope.user).success(function(){
10                 $location.path('/#/');
11             });
12         }
13     };
14 });
15
```



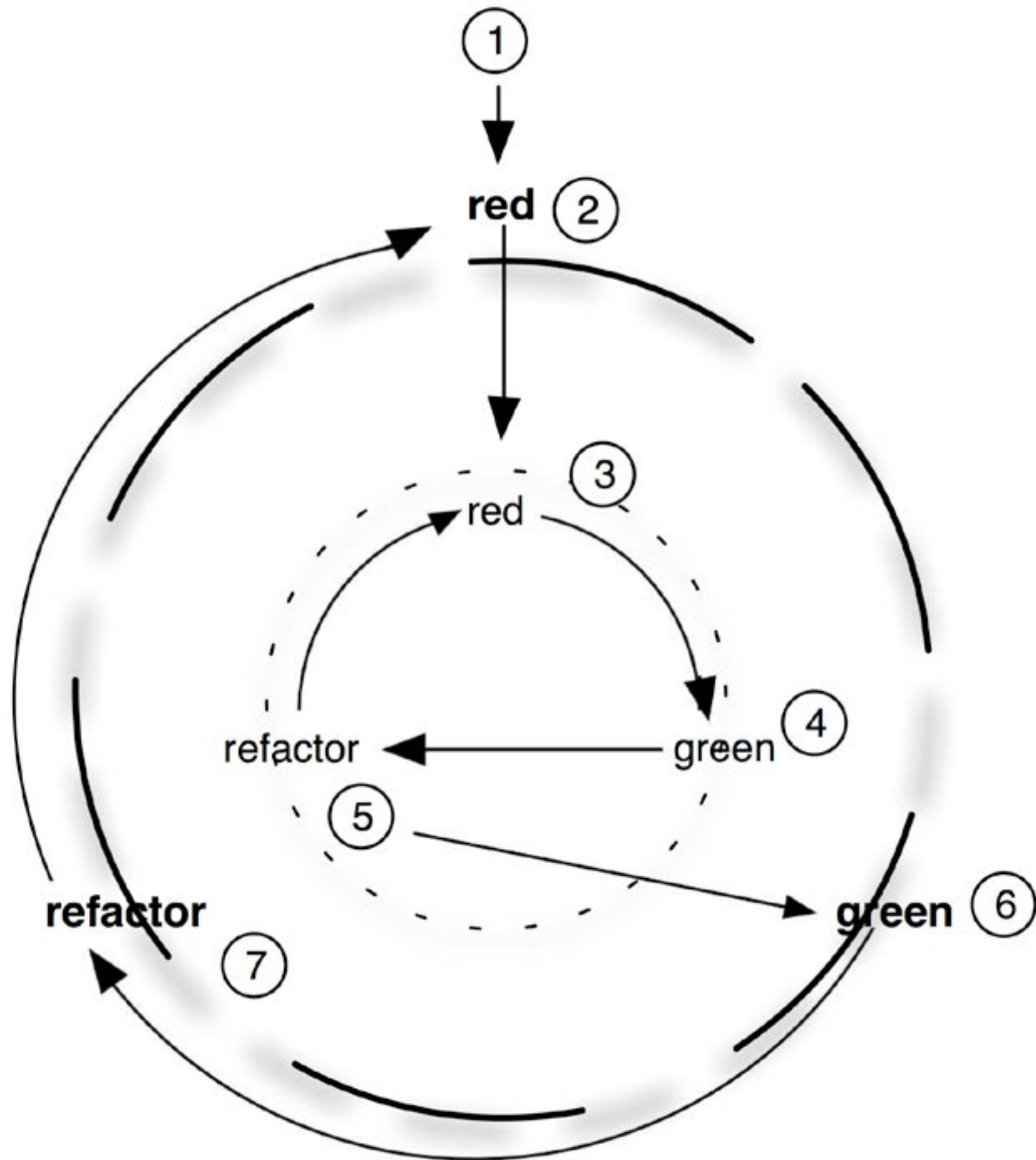
```
1
2 'use strict';
3
4 angular.module('applicationApp')
5 .service('User', function($http) {
6
7     function User(){
8         this.name = null;
9         this.email = null;
10        this.password = null;
11    }
12
13    User.create = function(user){
14        return $http.post('http://localhost:3000/users', user).success(function(){
15            localStorage.user = JSON.stringify(user);
16        });
17    };
18
19    return User;
20 });
21
```

```
3  var $httpBackend;
4
5  beforeEach(module('applicationApp'));
6
7  beforeEach(inject(function($injector){
8    $httpBackend = $injector.get('$httpBackend');
9  }));
10
11 it('creates a new user', inject(function($controller, $rootScope, $location){
12   var $scope = $rootScope;
13   $controller('SignUpCtrl', {$scope: $scope});
14   $scope.$apply(function(){
15     $scope.user = {
16       name: 'Sebastian Pape',
17       email: 'sebastian.pape@local.ch',
18       password: 'ngPassword'
19     };
20   });
21   $httpBackend.expectPOST('http://localhost:3000/users').respond(201, '');
22   $scope.submit(true);
23   $httpBackend.flush();
24   expect($location.path()).toBe('/#/');
25 });
26
27 afterEach(function() {
28   $httpBackend.verifyNoOutstandingExpectation();
29   $httpBackend.verifyNoOutstandingRequest();
30 });
```



```
5 it('creates a new user', function(){
6   inject(function($controller, $rootScope){
7     var $scope = $rootScope;
8     var UserMock = function(){};
9     UserMock.create = function(){
10      return {
11        success: function(c){
12          c();
13        }
14      };
15    };
16    spyOn(UserMock, 'create').and.callThrough();
17    $controller('SignUpCtrl', {$scope: $scope, User: UserMock});
18    $scope.$apply(function(){
19      $scope.user = {
20        name: 'Sebastian Pape',
21        email: 'sebastian.pape@local.ch',
22        password: 'ngPassword'
23      };
24    });
25    $scope.submit(true);
26    expect(UserMock.create).toHaveBeenCalled();
27  });
28 });
```

The Cycle of Delivery



Thank you!