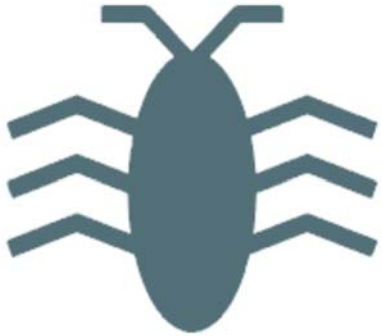
KEEP CALM AND TEST EVERYTHING

WITH KARMA & PROTRACTOR

Johannes Hoppe

Unit Testing



Früher

Jasmine HTML runner

```
<!DOCTYPE html>
<html>
<head>
    <title>Jasmine Spec Runner</title>

    <link rel="stylesheet" href="lib/jasmine-1.3.1/jasmine.css" />
    <script src="lib/jasmine-1.3.1/jasmine.js"></script>
    <script src="lib/jasmine-1.3.1/jasmine-html.js"></script>

    <!-- include source files here... -->
    <script src="src/Player.js"></script>
    <script src="src/Song.js"></script>

    <!-- include spec files here... -->
    <script src="spec/SpecHelper.js"></script>
    <script src="spec/PlayerSpec.js"></script>

    <script>

        (function () {

            var htmlReporter = new jasmine.HtmlReporter();
            var jasmineEnv = jasmine.getEnv();

            jasmineEnv.addReporter(htmlReporter);
            jasmineEnv.specFilter = function (spec) {
                return htmlReporter.specFilter(spec);
            };

            var currentWindowOnload = window.onload;

            window.onload = function () {
```

```
                if (currentWindowOnload) { currentWindowOnload(); }
                jasmineEnv.execute();
            };
        })();
    </script>

</head>

<body>
</body>
</html>
```

Früher

Jasmine 1.3.1 revision 1354556913 finished in 0.086s

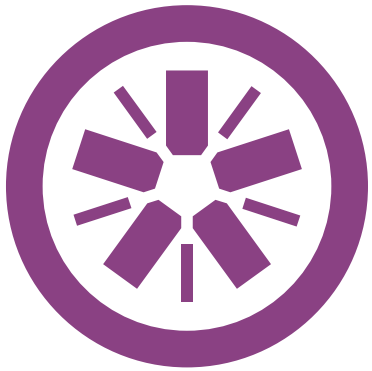
• • • • •

Passing 5 specs No try/catch ☐

```
Player
  should be able to play a Song

  when song has been paused
    should indicate that the song is currently paused
    should be possible to resume
  tells the current song if the user has made it a favorite

  #resume
    should throw an exception if song is already playing
```



Jasmine

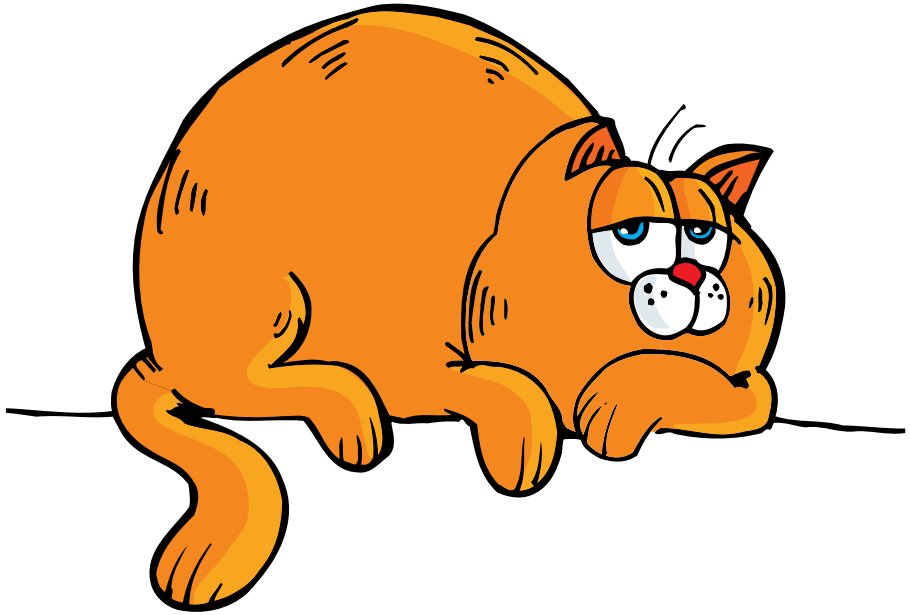
behavior-driven development (BDD) framework

Suites, Specs & Expectations

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


KARMA

> On the AngularJS team, we rely on testing and we always seek better tools to make our life easier. That's why we created Karma - a test runner that fits all our needs.



Demo Hello Garfield: 1 | 2

var gretchenfrage = plain || grunt || gulp;

Heute mal

Plain Node.js

```
$ npm init
$ npm install karma --save-dev
$ npm install -g karma-cli
$ npm install karma-jasmine karma-chrome-launcher --save-dev
$ karma init
$ karma start  #oder: npm test
_
```

Tipp: Karma nicht global installieren - Versionskonflikte zwischen Projekten!

Windows?

```
SET CHROME_BIN = "C:\Users\Me\AppData\Local\Google\Chrome\Appl
_
```

package.json

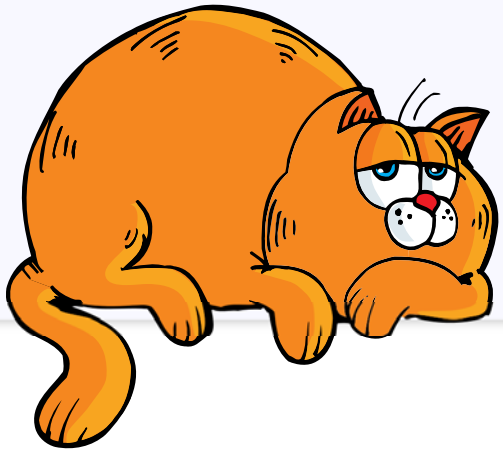
```
{
  "name": "Tests-Karma-Protractor",
  "version": "1.0.0",
  "description": "Example package",
  "main": "index.js",
  "directories": {
    "test": "test"
  },
  "scripts": {
    "test": "karma start"
  },
  "author": "Johannes Hoppe",
  "license": "MIT",
  "devDependencies": {
    "jasmine-core": "^2.3.4",
    "karma": "^0.12.36",
    "karma-chrome-launcher": "^0.1.12",
    "karma-jasmine": "^0.3.5"
  }
}
```

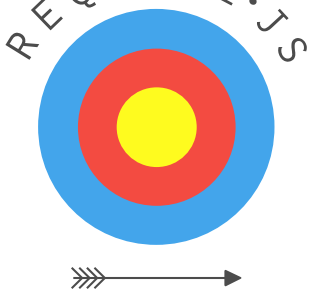
karma.conf.js

```
module.exports = function(config) {
    config.set({
        basePath: 'scripts',
        frameworks: ['jasmine'],
        files: ['**/*.js'],
        exclude: [],
        autoWatch: true,
        browsers: ['Chrome']
    });
};
```

karma.conf.js

```
files: [ /* Reihenfolge? */ ]
```





var gretchenfrage2 = <script> || RequireJS || CommonJS;

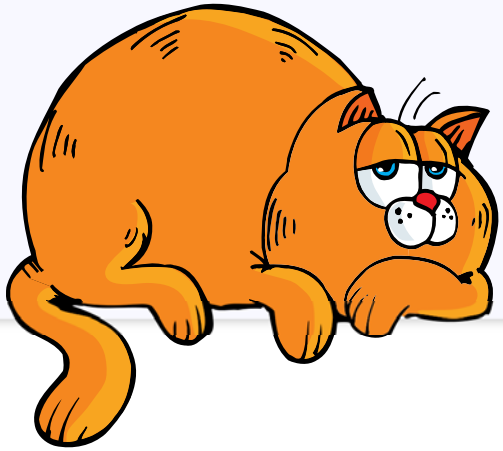
```
define('Garfield', [], function() {

    var Garfield = function () {
        this.getName = function () {
            return "Garfield";
        };
    };

    return Garfield;
});

require(['Garfield'], function(Garfield) {

    var garfield = new Garfield();
    garfield.getName();
});
```



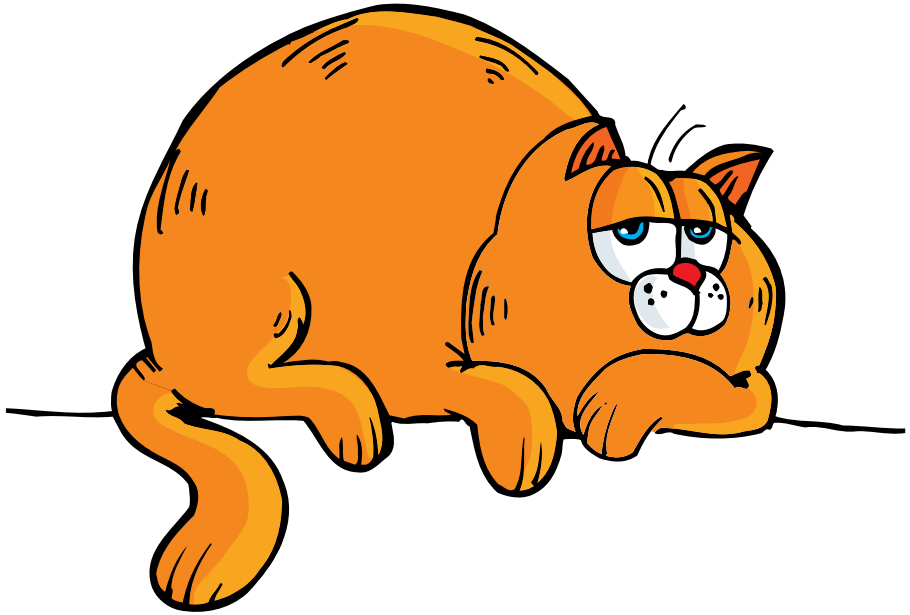
Finaler Unit-Test

```
define(['Garfield'], function(Garfield) {

    describe('Garfield', function() {

        it('should have a name', function() {

            var garfield = new Garfield();
            expect(garfield.getName()).toEqual("Garfield");
        });
    });
});
```



Demo Hello Garfield (require.js): 1 | 2

E2E Testing


Protractor
end to end testing for AngularJS

Terminologie

Selenium	OS-project to automate web browsers
Selenium 2.0	concept of WebDrivers
WebDrivers	drivers control a browser (e.g. ChromeDriver) and implement the WebDriver wire protocol
WebDriverJS	JavaScript bindings for **WebDriver** (over wire protocol)
Protractor	end-to-end test framework (not only) for AngularJS on top of **WebDriverJS**

mehr

Installation

```
$ npm install -g protractor # 💾 protractor 💾 webdriver-manager
$ webdriver-manager update  # 💾 ChromeDriver
$ webdriver-manager start
$ protractor protractor.conf.js
```

protractor.conf.js

```
exports.config = {
    seleniumAddress: 'http://localhost:4444/wd/hub',
    specs: ['e2e/*Spec.js']
};
```

AngularJS Spec

```
describe('Protractor Demo Spec', function() {

    it('should have a title', function() {
        browser.get('http://johanneshoppe.github.io/angular_calc/');
        expect(browser.getTitle()).toEqual('AngularJS Calculator');
    });
});
```

Expect

das normale Jasmine expect wurde ge-patched

```
browser.getTitle().then(function (title) {
    expect(title).toBe('AngularJS Calculator');
});
```

```
expect(browser.getTitle()).toBe('AngularJS Calculator');
```

Ohne Angular

```
describe('Johannes Talk at DWX', function() {

    beforeEach(function () {
        // no waiting on angular to load and finish its tasks
        browser.ignoreSynchronization = true;
    });

    it('should have the expected title', function() {

        browser.get('http://www.developer-week.de/Programm/Veranstaltung/(event)/18498');
        var heading = element(by.tagName('h2'));
        expect(heading.getText()).toEqual('JS Unit- und Oberflächentests mit Karma & Protractor');
    });

    afterEach(function() {
        return browser.ignoreSynchronization = false;
    });
});
```

Locator Spickzettel

by.irgendwas() + asynchronous actions

`<span>{{yourName}}</span>`	`element(by.binding('yourName'));`
`<input ng-model="password">`	`element(by.input('password'));`
`<tr ng-repeat="foo in foos">`	`element(by.repeater('foo in foos')).row(1)`
	`element.by(repeater('foo in foos')).column('foo.bar');`

Debugging

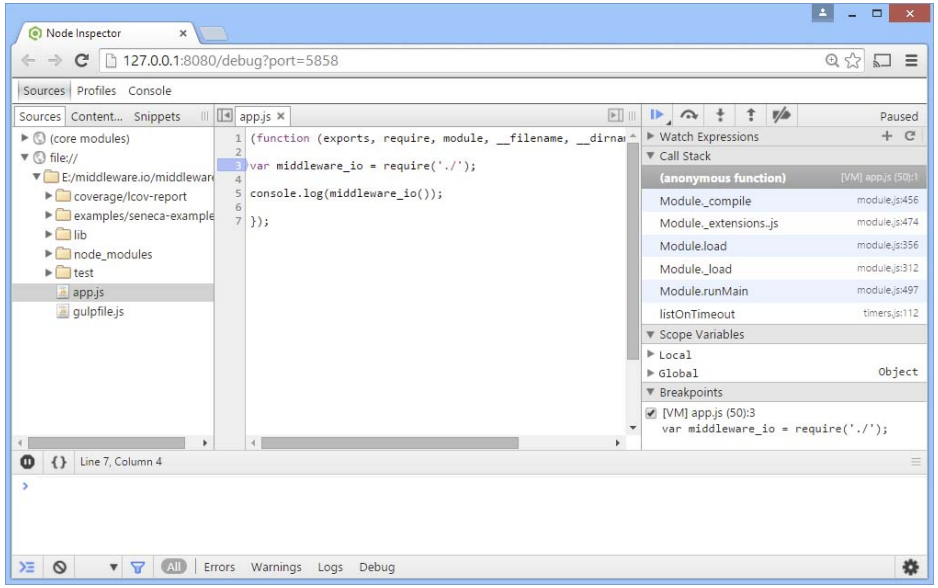
wd-debug	`browser.pause();`
	`// Enter "c" to move the test forward by one task`
	`// Enter "repl" to enter interactive mode`

node debugger	`cmd: protractor debug protractor.conf.js`
	`browser.debugger()`
	`// Enter "c" to continue after breakpoint`

node-debug	`cmd: node-debug node_modules/protractor/bin/protractor protractor.conf.js` `browser.debugger()`

Schamlose Werbung

:-)



gulp-node-debug

Downloads

github.com/JohannesHoppe/Presentations2015

Danke!

 johanneshoppe.de
 @JohannesHoppe
 Johannes.Hoppe