



Tests Unitaires Front-end



Jasmine // Karma



Introduction



Tests Unitaire ? Pourquoi ? Qui ? Comment ?

Définitions importantes

Personnages notables





Kent Beck



Gang Of Four



Erich Gamma





La configuration Karma

```
// Karma configuration
module.exports = function(config) {
  config.set({
    // base path, that will be used to resolve files and exclude
    basePath: '',

    // testing framework to use (jasmine/mocha/qunit/...)
    frameworks: ['jasmine'],

    // list of files / patterns to load in the browser
    files: [
      'app/bower_components/angular/angular.js',
      'app/bower_components/angular-mocks/angular-mocks.js',
      'app/bower_components/angular-resource/angular-resource.js',
      'app/bower_components/angular-cookies/angular-cookies.js',
      'app/bower_components/angular-sanitize/angular-sanitize.js',
      'app/bower_components/angular-route/angular-route.js',
      'app/scripts/*.js',
      'app/scripts/**/*.js',
      'test/mock/**/*.js',
      'test/spec/**/*.js'
    ],

    // list of files / patterns to exclude
    exclude: [],

    // web server port
    port: 8080,
```

```
    // level of logging
    // possible values: LOG_DISABLE || LOG_ERROR || LOG_WARN || LOG_INFO || LOG_DEBUG
    logLevel: config.LOG_INFO,

    // enable / disable watching file and executing tests whenever any file changes
    autoWatch: false,

    // Start these browsers, currently available:
    // - Chrome
    // - ChromeCanary
    // - Firefox
    // - Opera
    // - Safari (only Mac)
    // - PhantomJS
    // - IE (only Windows)
    browsers: ['Chrome'],

    // Continuous Integration mode
    // if true, it capture browsers, run tests and exit
    singleRun: true,

    plugins: [
      'karma-chrome-launcher',
      'karma-jasmine'
    ]
  });
};
```

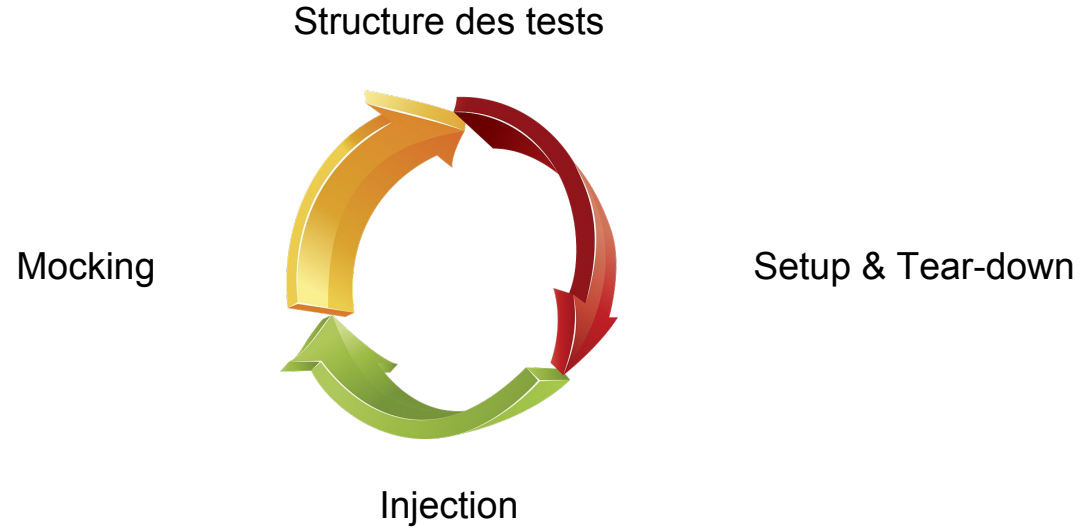


Plus d'informations :

<http://karma-runner.github.io/0.13/config/configuration-file.html>



Notions Générales



```
beforeEach(inject(function($httpBackend, $rootScope, $controller) {  
  httpBackend = $httpBackend;  
  httpBackend.expectGET('phones/phones.json').  
    respond([{name: 'Nexus S'}, {name: 'Motorola DROID'}]);  
  
  scope = $rootScope.$new();  
  ctrl = $controller(PHONE_LIST_CTRL, {$scope: scope});  
}));  
  
it('should create "phones" model with 2 phones fetched from xhr', function() {  
  expect(scope.phones).toBeUndefined();  
  httpBackend.flush();  
  
  expect(scope.phones).toEqual([{name: 'Nexus S'},  
                                {name: 'Motorola DROID'}]);  
});
```



Les tests



Les Matchers



Les Spies



Les Matchers

toBe()	toEqual()	toBeTruthy()	toBeFalsy()
toBeDefined()	toBeUndefined()	toBeNull()	toMatch()
toBeCloseTo()	toBeGreaterThan()	toBeLessThan()	toContain()
toThrow()	toThrowError()		



Les Spies

toHaveBeenCalled()	toHaveBeenCalledWith()	callThrough()*
returnValue()*	callFake()*	throwError()*

.calls

any()	count()	argsFor()	allArgs()
all()	mostRecent()	first()	reset()



Plus d'informations :

<http://jasmine.github.io/2.4/introduction.html>





MISE EN PRATIQUE



Ressource :

<https://github.com/meitneshi/unit-testing-workshop-v2>

RTFM (et si ça ne fonctionne toujours pas, RTFMA)

