TESTS ET ANGULARJS

SQLI GROUP

LES TESTS

UNITAIRES ET END TO END

TESTS

LES CATÉGORIES DE TESTS AVEC ANGULARJS

+ La structure modulaire d'une application AngularJS encourage la mise en œuvre des tests.
+ Un test peut être « unitaire » (Unit Test) ou d'intégration (E2E – End To End)

Unit	E2E
Test du code	Test de la fonctionnalité
Interagit avec le code	Interagit avec le navigateur ou l'API du framework
Utiliser le plus possible des « mock »	Utiliser le moins possible des « mock »
Ne requiert pas de serveur web	Requiert un serveur web
Rapide	Lent

TESTS

DE NOMBREUX OUTILS DE TESTS FRONTEND

jQuery's TestSwarm

js-test-driver
Remote javascript console

mocha

buster.js

Chutzpah

Qu

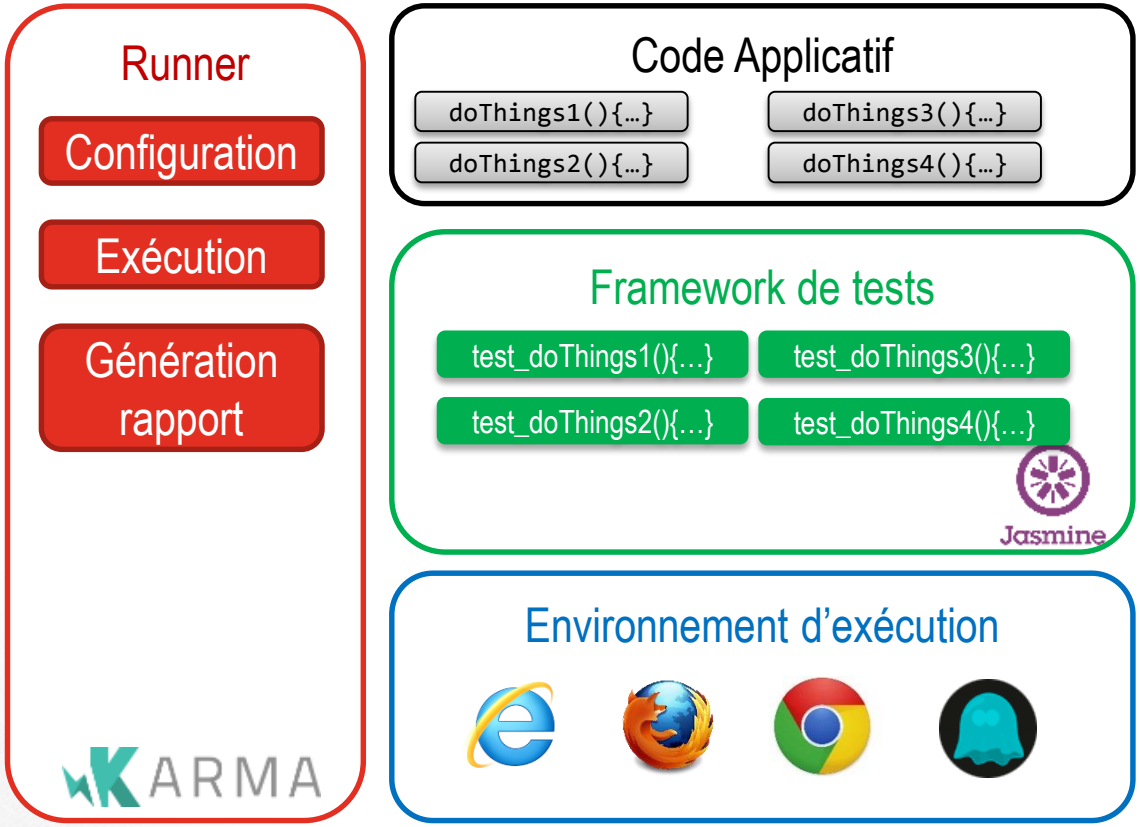
Chai Assertion Library

Sinon.JS

Se

TESTS

LES BRIQUES D'UN ENVIRONNEMENT DE TEST



SQLI GROUP

TESTS UNITAIRES

KARMA + JASMINE

TESTS UNITAIRES

KARMA – PRÉSENTATION

+ Karma est un exécuteur de test JavaScript basé sur Node.js et distribué en tant que package Node.
+ Il intègre de nombreux environnements d'exécution : Chrome, Chrome Canary, Internet Explorer (Windows uniquement), Firefox, Safari (Mac uniquement), PhantomJS, Opéra, etc.
+ Il supporte plusieurs frameworks de tests unitaires : Jasmine, QUnit, Mocha. Il est possible d'étendre ce support via l'écriture d'adapter.
+ Karma s'utilise en ligne de commande ou via une intégration dans d'autres outils (JetBrains WebStorm IDE, Grunt, etc.)

TESTS UNITAIRES

KARMA – COMMANDES CLÉS

+ Installation

```
npm install -g karma-cli

npm install karma
npm install karma-jasmine
npm install karma-chrome-launcher
npm install karma-phantomjs-launcher
```

+ Générer un fichier de configuration Karma

```
karma init <nom_fichier_config.js>
```

+ Lancer Karma

```
karma start <nom_fichier_config.js>
```

TESTS UNITAIRES

JASMINE – PRÉSENTATION

- Un framework de test JavaScript.
- Orienté BDD (Behavior-Driven Development – Développement piloté par le comportement).
- Ecriture de tests que « tout le monde peut comprendre ».

```
describe("Une suite", function() {

 it("décrit ici le premier test", function() {
    // Code du premier test
 });

 it("décrit ici le second test", function() {
    // Code du second test
 });

}
```

TESTS UNITAIRES

JASMINE – SUITE / CAS DE TEST / VÉRIFICATION DES RÉSULTATS

```
describe("Une suite avec 2 tests", function() {

 it("12 + 5 doit être égal à 17", function() {
    var a = 12;
    var b = 5;
    var resultat = a + b;

    expect(resultat).toBeDefined();
    expect(resultat).toBe(17);
    expect(resultat).not.toBe(a);
 });

 it("L'égalité de deux objets est bien gérée", function() {
    var a = { prop1 :"val1", prop2 : "val2" };
    var b = { prop1 :"val1", prop2 : "val2" };

    expect(a).toEqual(b);
 });
}
```

TESTS UNITAIRES

JASMINE – SETUP & TEARDOWN / MOT CLÉ THIS

```
describe("Une suite avec 2 tests", function() {

 beforeEach(function() {
    this.valeur = 5;
 });

 afterEach(function() {
    this.count = 1;
 });

 it("test 1", function() {
    expect(this.valeur).toEqual(5);
    expect(this.count).not.toBeDefined();
 });

 it("test 2", function() {
     expect(this.count).not.toBeDefined();
 });
}
```

TESTS UNITAIRES

JASMINE – SUITE IMBRIQUÉE

```
describe("Une suite", function() {

   it("test 1", function() {
      // code du test
   });

   describe("Une suite imbriquée", function() {

      it("test suite imbriquée 1", function() {
         // code du test
      });

      it("test suite imbriquée 2", function() {
         // code du test
      });
   });
});
```

MODULE NGMOCK

NGMOCK – PRÉSENTATION

+ ngMock est un module qui fournit un support pour injecter et « mocker » des services pendant des tests unitaires.
+ Il s'installe via l'inclusion du fichier angular-mock.js

ANGULAR MOCK

NGMOCK – LES MOCKS FOURNIS

Provider

$exceptionHandleProvider

Service

$exceptionHandler
$log
$interval
$httpBackend
$timeout

Type

angular.mock.TzDate

Function

angular.mock.dump
angular.mock.module
angular.mock.inject

MODULE NGMOCK

NGMOCK – ANGULAR.MOCK.MODULE()

+ Collecte les informations de configuration d'un module donné.

```
describe("Une suite", function() {

 beforeEach(module('monModule')); // angular.mock.module()

}
```

MODULE NGMOCK

NGMOCK – ANGULAR.MOCK.INJECT() 1/2

- Permet d'injecter des dépendances.
- Créé une nouvelle instance de $injector par test.

```
describe("Une suite", function() {

 beforeEach(module('monModule'));

 it("test 1",inject(function($rootScope, $controller,…){// angular.mock.inject()
    var scope = $rootScope.$new();
 }));

}
```

MODULE NGMOCK

NGMOCK – ANGULAR.MOCK.INJECT() 2/2

+ Résolution des références.

```
describe("Une suite", function() {

 var monService;

 beforeEach(inject(function(_monService_) {
    monService = _monService_;
 }));

 it("test 1", function() {
    monService.lancerTraitement();
 });

}
```

MODULE NGMOCK

NGMOCK – ANGULAR.MOCK.DUMP()

- Sérialise un objet AngularJS (scope, element, etc.) en chaine de caractères.
- Utile pour le debug.

```
angular.mock.dump(scope);
```

MODULE NGMOCK

NGMOCK – $EXCEPTIONHANDLERPROVIDER 1/2

- Un mock du service $exceptionHandler, le gestionnaire des exceptions non gérées d'une application AngularJS.
- Il permet de spécifier deux modes (via sa méthode « mode(UN_MODE)») : « rethrow » et « log ».
- « rethrow » est le mode par défaut. Il permet de propager une exception non gérée. Un test en cours d'exécution échouera en cas d'exception non gérée.
- « log » permet de tester les cas où une erreur est souhaitée dans le cadre du test.

MODULE NGMOCK

NGMOCK – $EXCEPTIONHANDLERPROVIDER 2/2

```
describe("Une suite", function() {

 it("test 1", function() {

    module(function($exceptionHandlerProvider){
         $exceptionHandlerProvider.mode('log');
    });

    inject(function($log, $exceptionHandler) {
         expect($exceptionHandler.errors).toEqual(['Erreur inattendue']);
    });
 });
}
```

MODULE NGMOCK

NGMOCK – $HTTPBACKEND

+ Un backend « bouchon » du service $http

```
describe("Une suite", function() {

 it("test 1",inject(function($httpBackend) {

    $httpBackend.when('GET', '/list').respond([{id : 15}]); // specs

    $httpBackend.expectGET('/list'); // expects

    $httpBackend.flush(); // déclenche les réponses en suspens

    $httpBackend.verifyNoOutstandingExpectation(); // verification toutes les
                                                   // requetes ont été expected
    $httpBackend.verifyNoOutstandingRequest(); // verification toutes les
                                               // requetes ont été flushées

 }));
}
```

MODULE NGMOCK

NGMOCK – AUTRES

- $interval - Mock du service $interval.
- $log – Mock sur service $log
- TzDate – Mock du type Date
- $timeout – Mock sur service $timeout

SQLI GROUP

EXERCICES PRATIQUES

TP 17 – TESTS UNITAIRES

TESTER LE DASHBOARDCONTROLLER. VÉRIFIER QU'APRÈS L'INITIALISATION, LES TABLEAUX DU DASHBOARD (AUTHORS, EDITORS, CATEGORIES) SONT BIEN REMPLIS

- ✓ charger le module applicatif
- ✓ injecter $controller et récupérer le contrôleur
- ✓ injecter $httpBackend et mocker l'appel au backend
- ✓ flush
- ✓ vérifier…

SQLI GROUP

TESTS END TO END

PROTRACTOR

NGMOCKE2E

$HTTPBACKEND

Ce n'est pas le même $httpBackend que pour les tests unitaires !

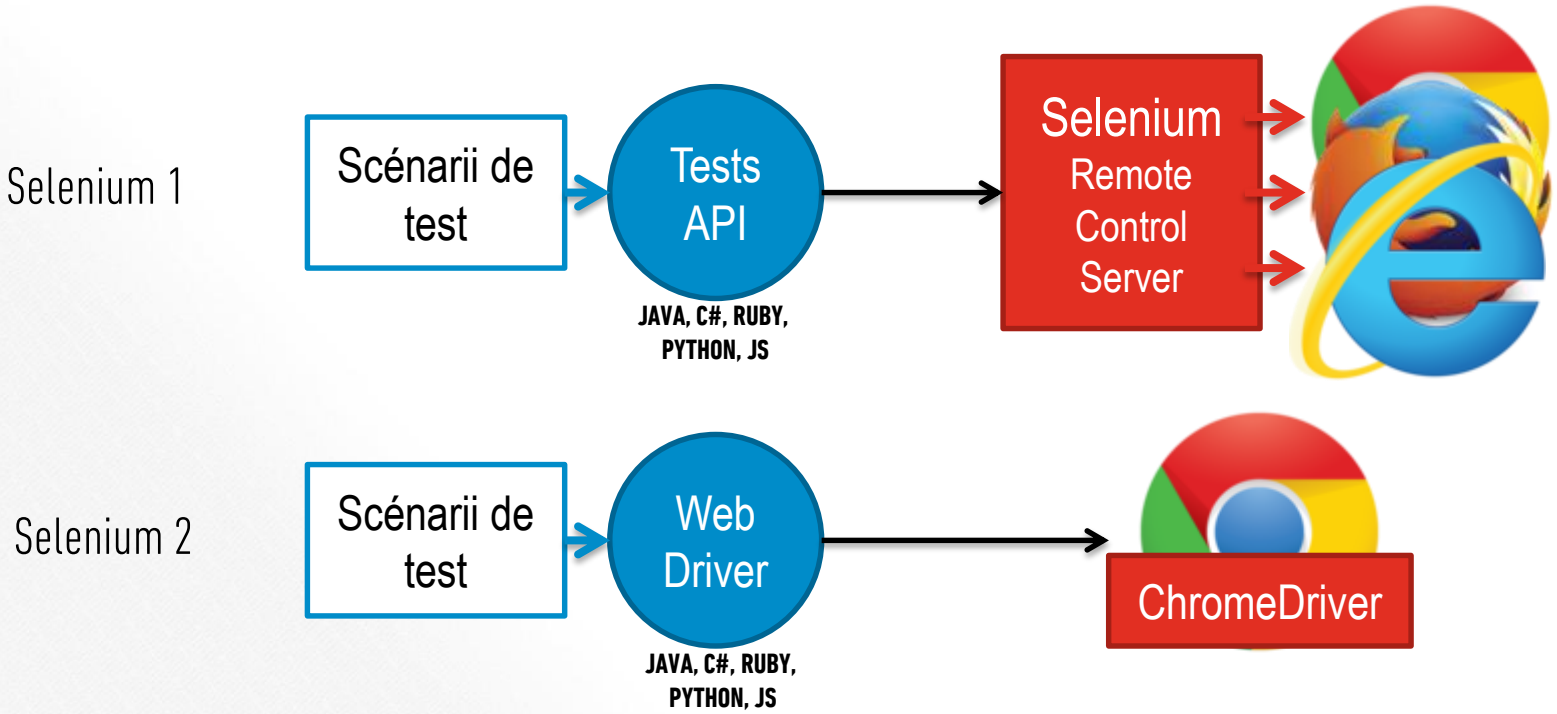
- Permet de simuler un backend HTTP pour notre application.
- L'index.html testé fait référence à cette application mockée

```
var myMockedApp = angular.module('myMockedApp', ['myApp', 'ngMockE2E']);
myMockedApp.run(function($httpBackend) {
    var trips = [{ ... }, ...];

    $httpBackend.whenGET('/trips').respond(trips);

    $httpBackend.whenPOST('/trips').respond(function(method, url, data) {
        trips.push(angular.fromJson(data));
    });

    // ...

    $httpBackend.whenGET(/^\/templates\/$/).passThrough();
});
```

SELENIUM 2.0

= SELENIUM 1 + WEBDRIVER

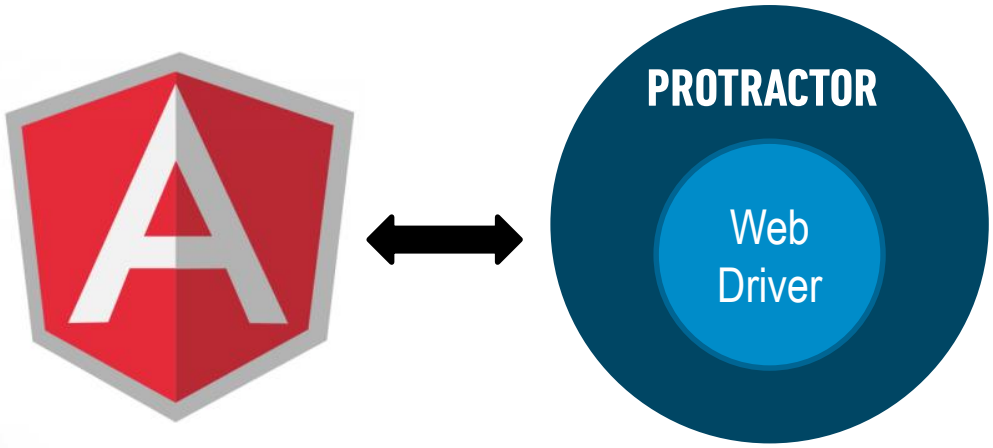
+ Selenium-Webdriver est un framework d'automatisation de navigateur. Les tests sont écrits avec l'API WebDriver, qui communique avec un serveur Selenium pour contrôler les navigateurs cibles.



PROTRACTOR

LE FRAMEWORK DE TEST END TO END D'ANGULARJS

- Protractor est un « wrapper » autour de WebDriverJS (module Node.js).
- Il permet une intégration naturelle avec AngularJS
- S'utilise avec Jasmine (Mocha possible)



PROTRACTOR

INSTALLATION ET LANCEMENT

+ Installation de Protractor

```
npm install -g protractor
```

+ Installation de selenium server (standalone) et du chromedriver

```
webdriver-manager update
```

+ Lancement du standalone selenium server (nécessite un JDK)

```
webdriver-manager start
```

+ Lancement du test avec Protractor

```
protractor conf.js
```

PROTRACTOR

CONFIGURATION

```
exports.config = {

 seleniumAddress: 'http://localhost:4444/wd/hub',

  multiCapabilities: [{
    'browserName': 'chrome'
  }, {
    'browserName': 'firefox'
  }],

  baseUrl: 'http://localhost:8080',

  specs: [
    'spec/*_spec.js'
  ],

};
```

PROTRACTOR

INSTALLATION ET LANCEMENT

```
protractor conf.js
```

```
[launcher] Running using config.multiCapabilities - config.capabilities will be ignored
[launcher] Running 2 instances of WebDriver.........
-----------------------------------
PID: 3284 (capability: chrome #1)
-----------------------------------

Using the selenium server at http://localhost:4444/wd/hub
...

Finished in 18.017 seconds
3 tests, 5 assertions, 0 failures


[launcher] 1 instance(s) of WebDriver still running...
-----------------------------------
PID: 7212 (capability: firefox #1)
-----------------------------------

Using the selenium server at http://localhost:4444/wd/hub
...

Finished in 19.352 seconds
3 tests, 5 assertions, 0 failures
```

PROTRACTOR

EXEMPLE DE TEST

```
describe('angularjs homepage todo list', function() {
    it('should add a todo', function() {
        browser.get('http://www.angularjs.org');

        element(by.model('todoText')).sendKeys('write a protractor test');
        element(by.css('[value="add"]')).click();

        var todoList = element.all(by.repeater('todo in todos'));
        expect(todoList.count()).toEqual(3);
        expect(todoList.get(2).getText()).toEqual('write a protractor test');
    });
});
```

PROTRACTOR

API

+ **browser** : wrapper autour de l'instance WebDriver. Permet de naviguer.
+ **element** : Permet d'interagir avec des élements du DOM
+ **by** : Permet de sélectionner des élements (par css, modèle , repeater...)
+ Ainsi que l'API de Jasmine (expect...)

https://github.com/angular/protractor/blob/master/docs/api.md

SQLI GROUP

EXERCICES PRATIQUES

TP 18 – TESTS E2E

TESTER LE DASHBOARDCONTROLLER. VÉRIFIER QUE SUR LA PAGE DASHBOARD, LES DONNÉES SONT BIEN PRÉSENTES

- ✓ aller sur l'url du dashboard
- ✓ sélectionner les repeat
- ✓ vérifier…

FIN