

Tester les applications Angular autrement



mais les tester tout de même!

Les Tests Automatisés, c'est bien

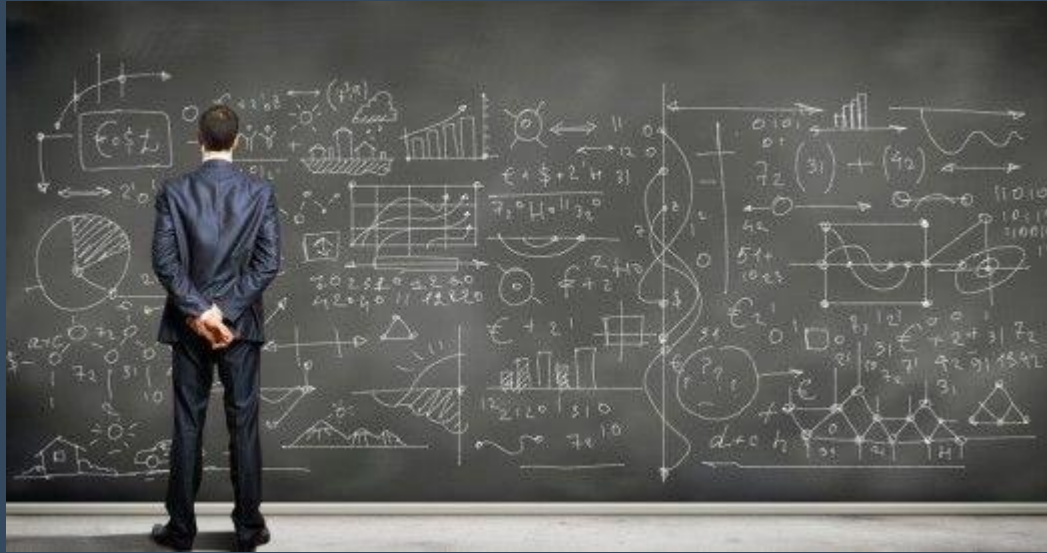
- **Mieux préciser les spécifications**
- **Enrichir la documentation**
- **Offrir des exemples significatifs**
- **Mieux concevoir et structurer le code**

Les Tests Automatisés, c'est bien

- **Contrôler la régression**
- **Détecter et corriger rapidement les bugs**
- **Remettre en cause la qualité la conception**
- **Améliorer la qualité des livrables**

Les Tests Automatisés, c'est bien

Oui, mais à quel prix ?



Tests Smells

- **Difficulté d'implémentation**
- **Tests non isolés**
- **Tests lents**
- **Tests avec des résultats intermittents**
- **Tests fragiles**
- **Tests obscures**

**Les Tests Automatisés, c'est
difficile**

Mais c'est un mal nécessaire

Les Tests Automatisés, c'est difficile



<https://goo.gl/t3GQW6>

Les Tests Automatisés : trouver la balance



https://youtu.be/_pnW-JjmyXE

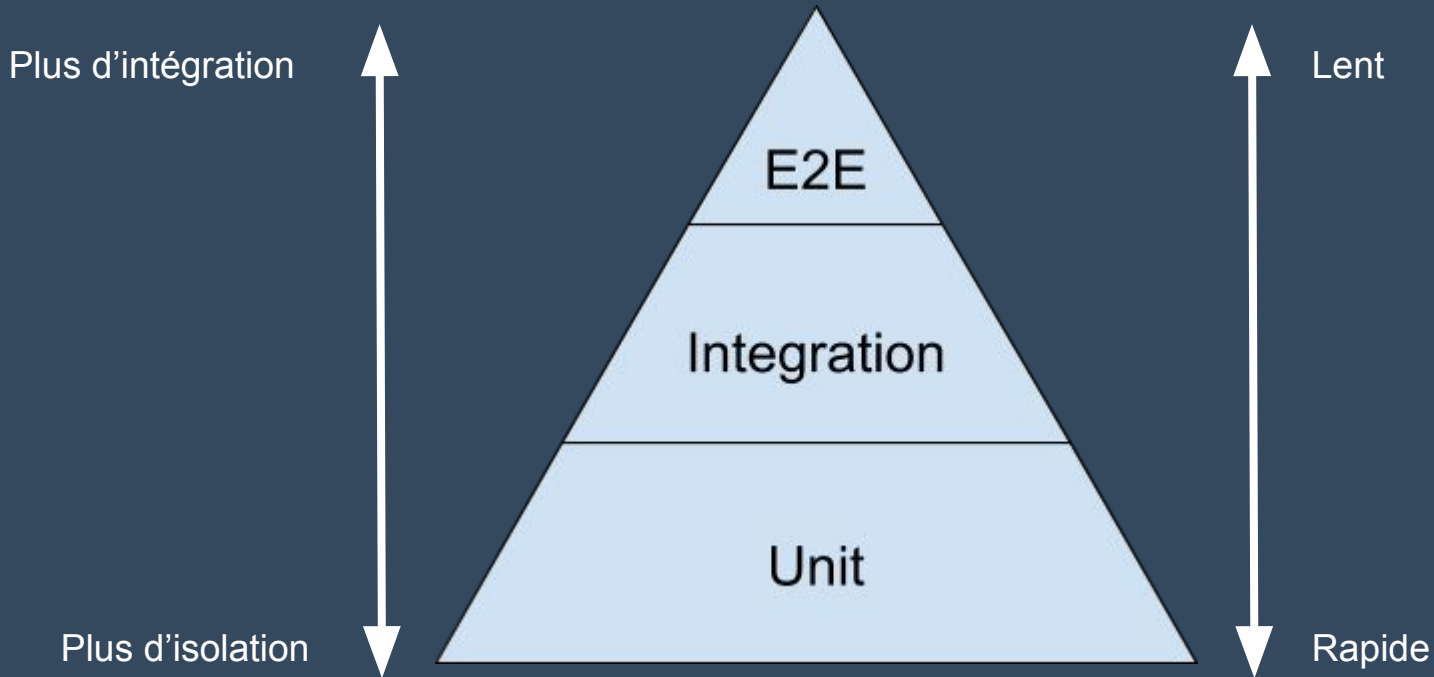
Mission Impossible

Vous faire aimer les tests !

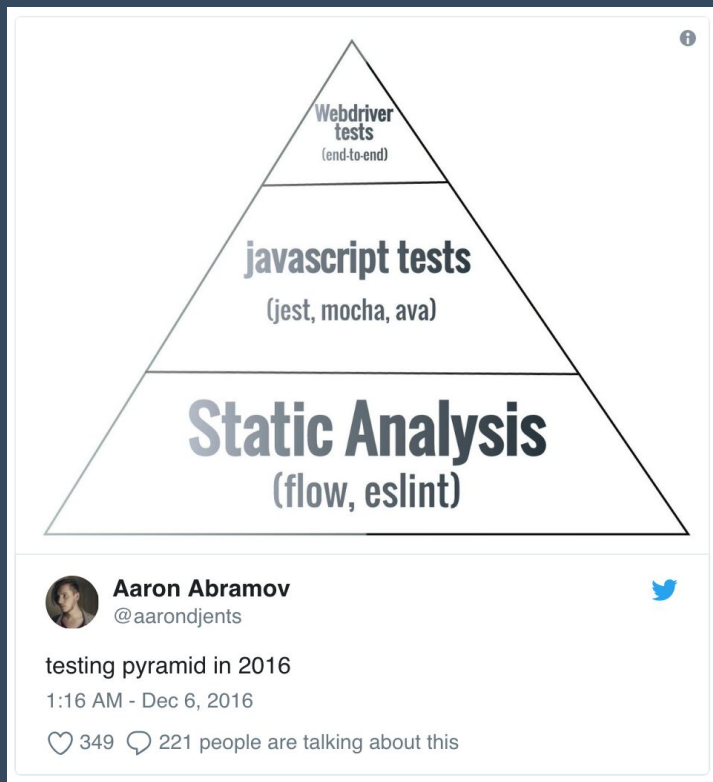
Comment ?

- **Trouver la meilleure stratégie**
- **Trouver les bons compromis**
- **Trouver des outils developer-friendly**

Pyramide des tests



Pyramide des tests (version Front)



Nouvelle tendance Front



Kent C. Dodds [Follow](#)

Making software development more accessible · Husband, Father, Latter-day Saint, Teacher, OSS, GDE, @TC39 · @PayPalEng @eggheadio @FrontendMasters · #JS
Oct 16, 2017 · 6 min read

Write tests. Not too many. Mostly integration.

Guillermo Rauch tweeted this a while back. Let's take a quick dive into what it means.

A while back, [Guillermo Rauch](#) (creator of [Socket.io](#) and founder of [Zeit.co](#) (the company behind a ton of the awesome stuff coming out lately)) [tweeted something profound](#):



Guillermo Rauch ✓

@rauchg

Following



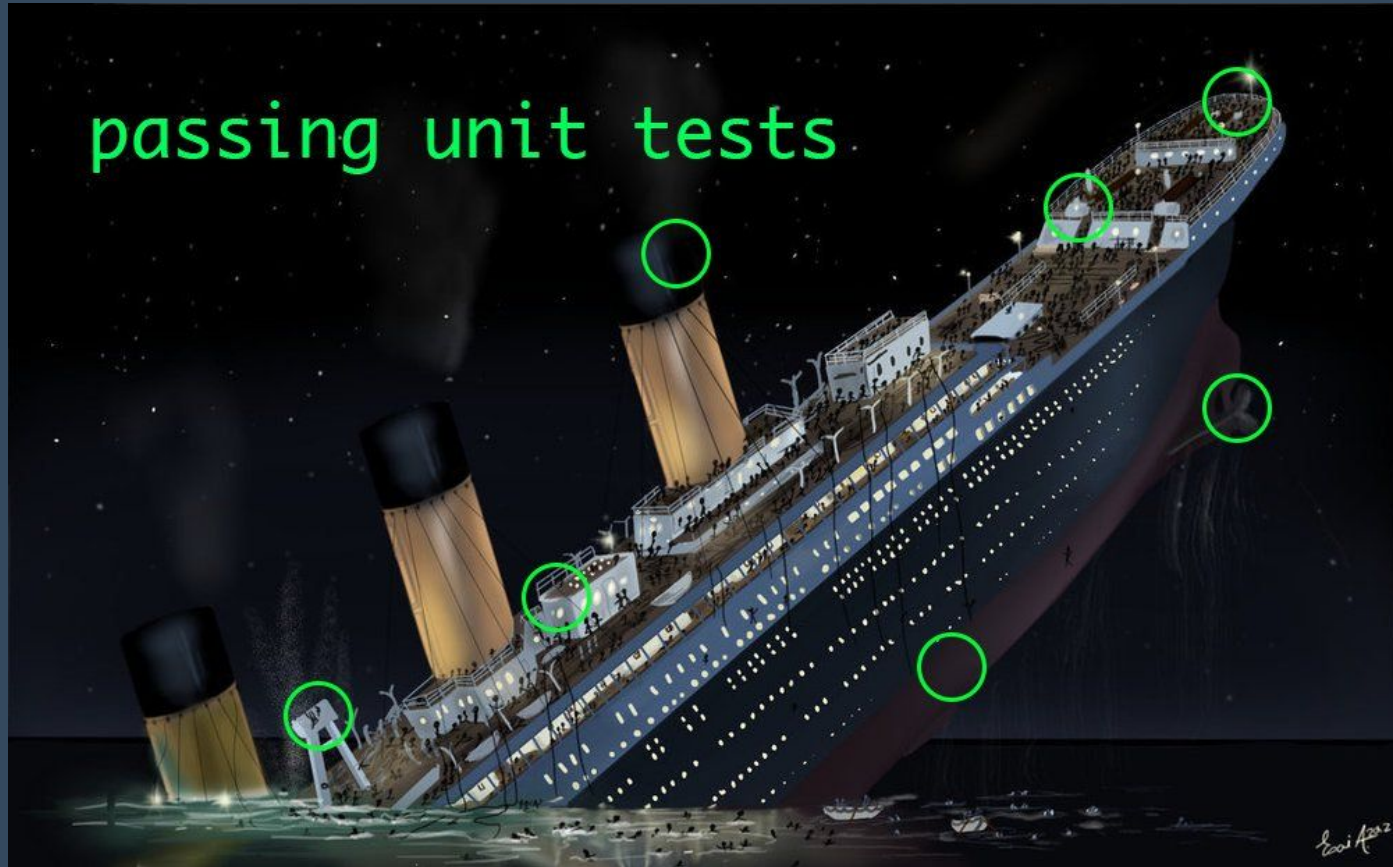
Write tests. Not too many. Mostly integration.

8:43 AM - 10 Dec 2016 from San Francisco, CA

111 Retweets 357 Likes

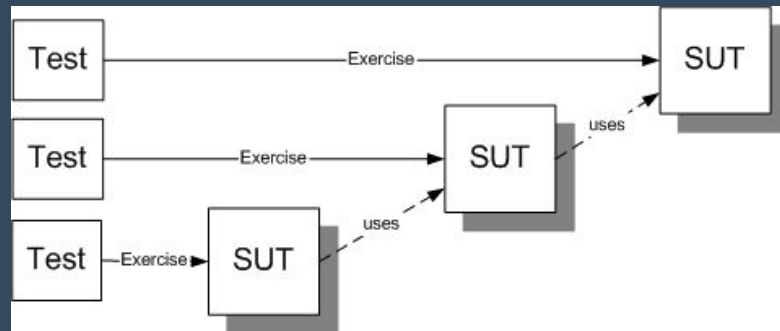
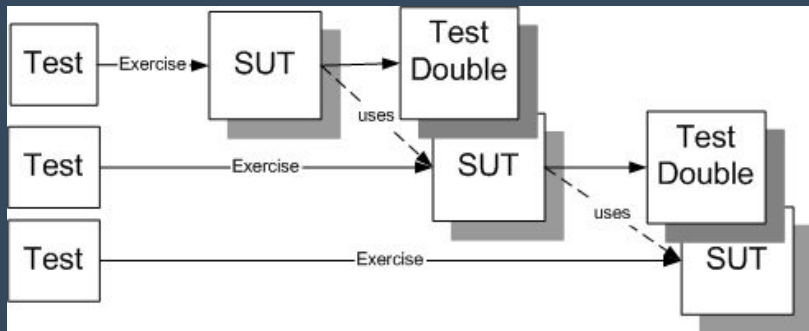


Quand on manque de tests d'intégration



Stratégies des tests

Outside-In Vs Inside-Out



<http://xunitpatterns.com/Philosophy%20Of%20Test%20Automation.html>

Composant Angular



```
@Component({
  selector: 'app-login-form',
  templateUrl: './login-form.component.html'
})
export class LoginFormComponent {
  form: FormGroup;
  constructor(private loginService: LoginService, private fb: FormBuilder) {
    this.form = this.fb.group({
      username: '',
      password: ''
    })
  }
  submit() {
    this.loginService.authenticate(this.form.value);
  }
}
```

Composant Angular



```
<app-logo></app-logo>
```

```
<h1>Authentication</h1>
```

```
<form [formGroup]="form" (ngSubmit)="submit()">  
  <input formControlName="username" name="username" type="text">  
  <input formControlName="password" name="password" type="password">  
  <button [disabled]="!form.valid" type="submit" mat-raised-button>Se connecter</button>  
</form>
```


Module Angular



```
@NgModule({  
  imports: [  
    CommonModule, FormsModule, ReactiveFormsModule,  
  ],  
  declarations: [  
    LoginFormComponent,  
    AppLogoComponent,  
  ],  
  providers: [  
    LoginService  
  ]  
})  
export class LoginModule { }
```

TestBed Angular



```
let component: LoginFormComponent;
let fixture: ComponentFixture<LoginFormComponent>;

beforeEach(async(() => {
  TestBed.configureTestingModule({
    declarations: [ LoginFormComponent ]
  })
  .compileComponents();
}));

beforeEach(() => {
  fixture = TestBed.createComponent(LoginFormComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should create', () => {
  expect(component).toBeTruthy();
});
```

TestBed Angular



```
<app-logo></app-logo>
```

```
<h1>Authentication</h1>
```

```
<form [formGroup]="form" (ngSubmit)="submit()">
  <input formControlName="username" name="username" type="text">
  <input formControlName="password" name="password" type="password">
  <button [disabled]="!form.valid" type="submit" mat-raised-button>Se connecter</button>
</form>
```

Tests failed: 1 of 1 test - 676 ms

Test Results 676 ms
login-form.component.spec. 676 ms
LoginFormComponent 676 ms
should create 676 ms

Error: Template parse errors:

'app-logo' is not a known element:

1. If 'app-logo' is an Angular component, then verify that it is part of this module.
2. If 'app-logo' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@NgModule.schemas'.

TestBed Angular



```
beforeEach(async(() => {  
  TestBed.configureTestingModule({  
    imports: [ FormsModule, ReactiveFormsModule ],  
    declarations: [  
      LoginFormComponent,  
      AppLogoComponent  
    ],  
    providers: [  
      LoginService,  
    ]  
  })  
  .compileComponents();  
}));
```

Module/TestBed Angular



```
@NgModule({
  imports: [
    CommonModule, FormsModule, ReactiveFormsModule,
  ],
  declarations: [
    LoginFormComponent,
    AppLogoComponent,
  ],
  providers: [
    LoginService
  ]
})
export class LoginModule { }
```



```
TestBed.configureTestingModule({
  imports: [ FormsModule, ReactiveFormsModule ],
  declarations: [
    LoginFormComponent,
    AppLogoComponent
  ],
  providers: [
    LoginService,
  ]
})
```

TestBed : Isolation avec NO_ERRORS_SCHEMA



```
beforeEach(async(() => {  
  TestBed.configureTestingModule({  
    imports: [ FormsModule, ReactiveFormsModule ],  
    declarations: [  
      LoginFormComponent  
    ],  
    schemas: [ NO_ERRORS_SCHEMA ]  
  })  
  .compileComponents();  
}));
```

TestBed : Injection de dépendances



```
TestBed.configureTestingModule({  
  imports: [ FormsModule, ReactiveFormsModule ],  
  declarations: [  
    LoginFormComponent,  
    AppLogoComponent  
  ],  
  providers: [  
    {  
      provide: LoginService,  
      useValue: {  
        authenticate: mock()  
      }  
    }  
  ]  
})
```

Tester le composant et le rendu HTML



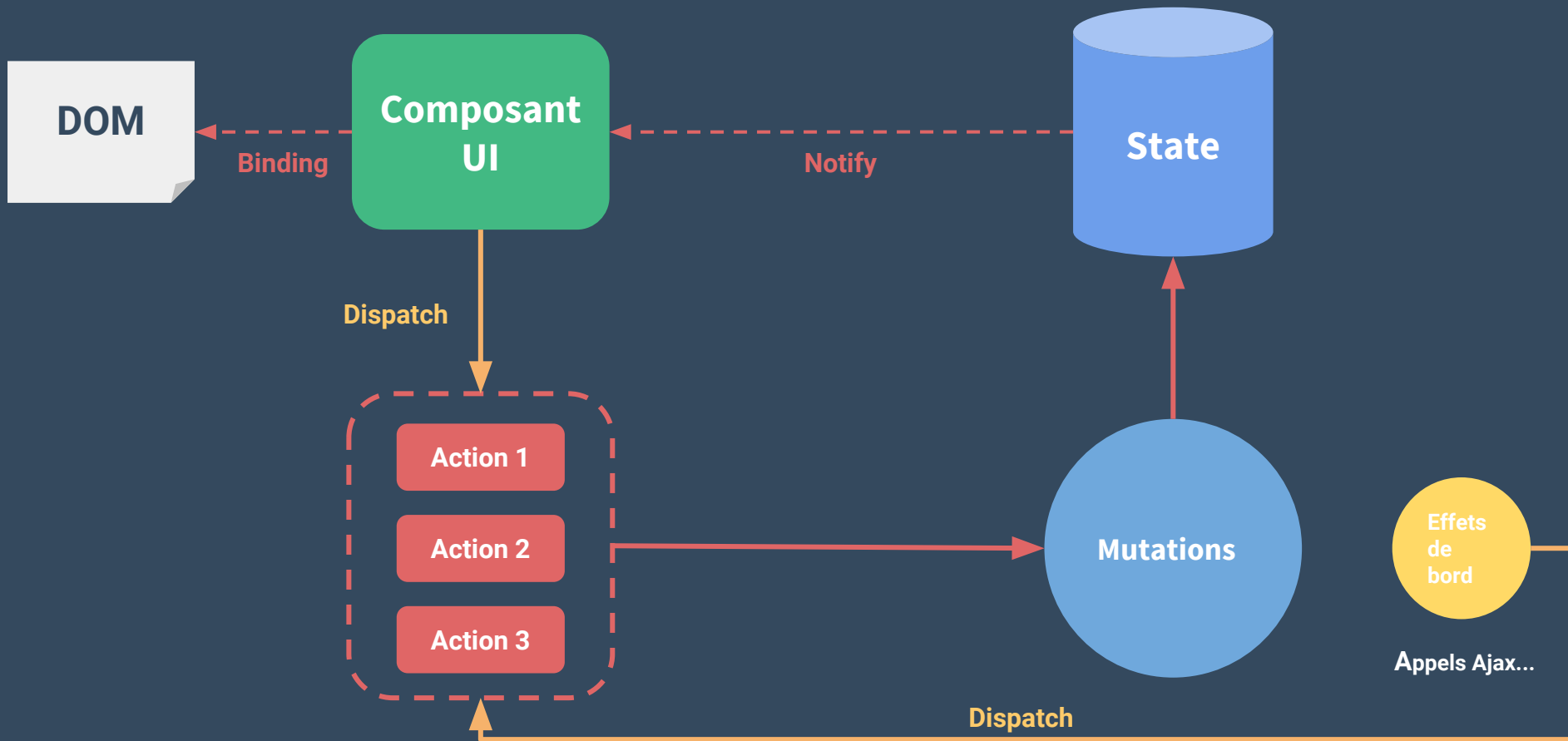
```
beforeEach(() => {  
  fixture = TestBed.createComponent(LoginFormComponent);  
  component = fixture.componentInstance;  
  fixture.detectChanges();  
  
  loginService = TestBed.get(LoginService);  
});  
  
it('should create', () => {  
  expect(component).toBeTruthy();  
});  
  
it('should render', () => {  
  const title: HTMLElement = fixture.nativeElement.querySelector(`h1`);  
  expect(title.textContent).toContain('Authentication');  
});
```


Tester le composant et le rendu HTML

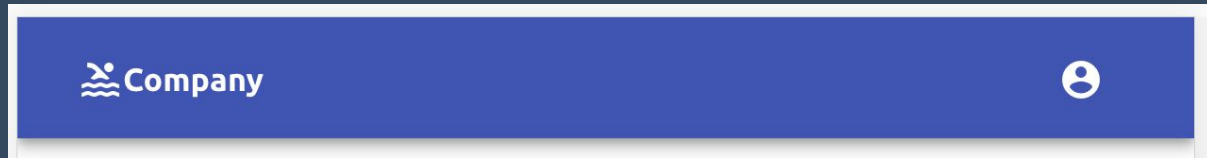


```
it('should call authenticate service', () => {  
  const username = fixture.nativeElement.querySelector('input[name=username]');  
  const password = fixture.nativeElement.querySelector('input[name=password]');  
  const submit = fixture.nativeElement.querySelector('button[type=submit]');  
  
  username.value = 'user';  
  username.dispatchEvent(new Event('input'));  
  password.value = 'pass';  
  username.dispatchEvent(new Event('input'));  
  fixture.detectChanges();  
  
  const errorMessage = fixture.nativeElement.querySelector('.requiredErrorMessage');  
  expect(errorMessage).toBeFalsy();  
  
  submit.click();  
  expect(loginService.authenticate).toHaveBeenCalledWith({ username: 'user', password: 'pass' });  
})
```

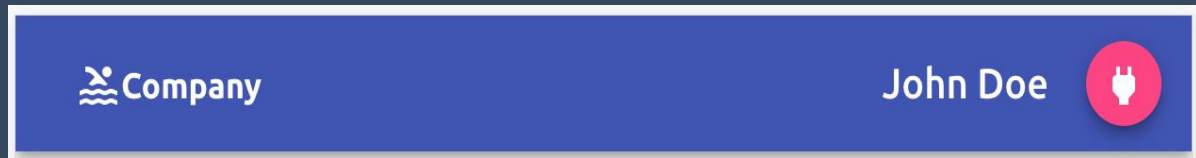
Utilisation d'un State Manager (Redux-like)



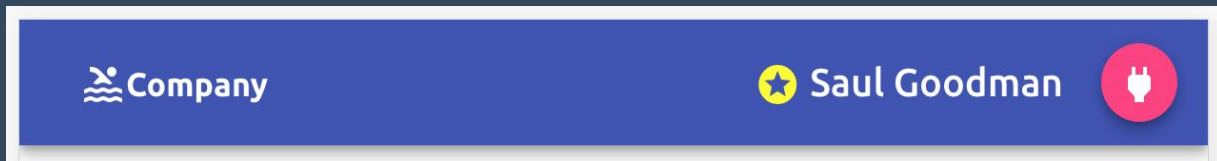
Utilisation d'un State Manager



```
{  
  auth: false,  
  user: null  
}
```



```
{  
  auth: true,  
  user: {  
    login: 'John Doe',  
    isVIP: false  
  }  
}
```



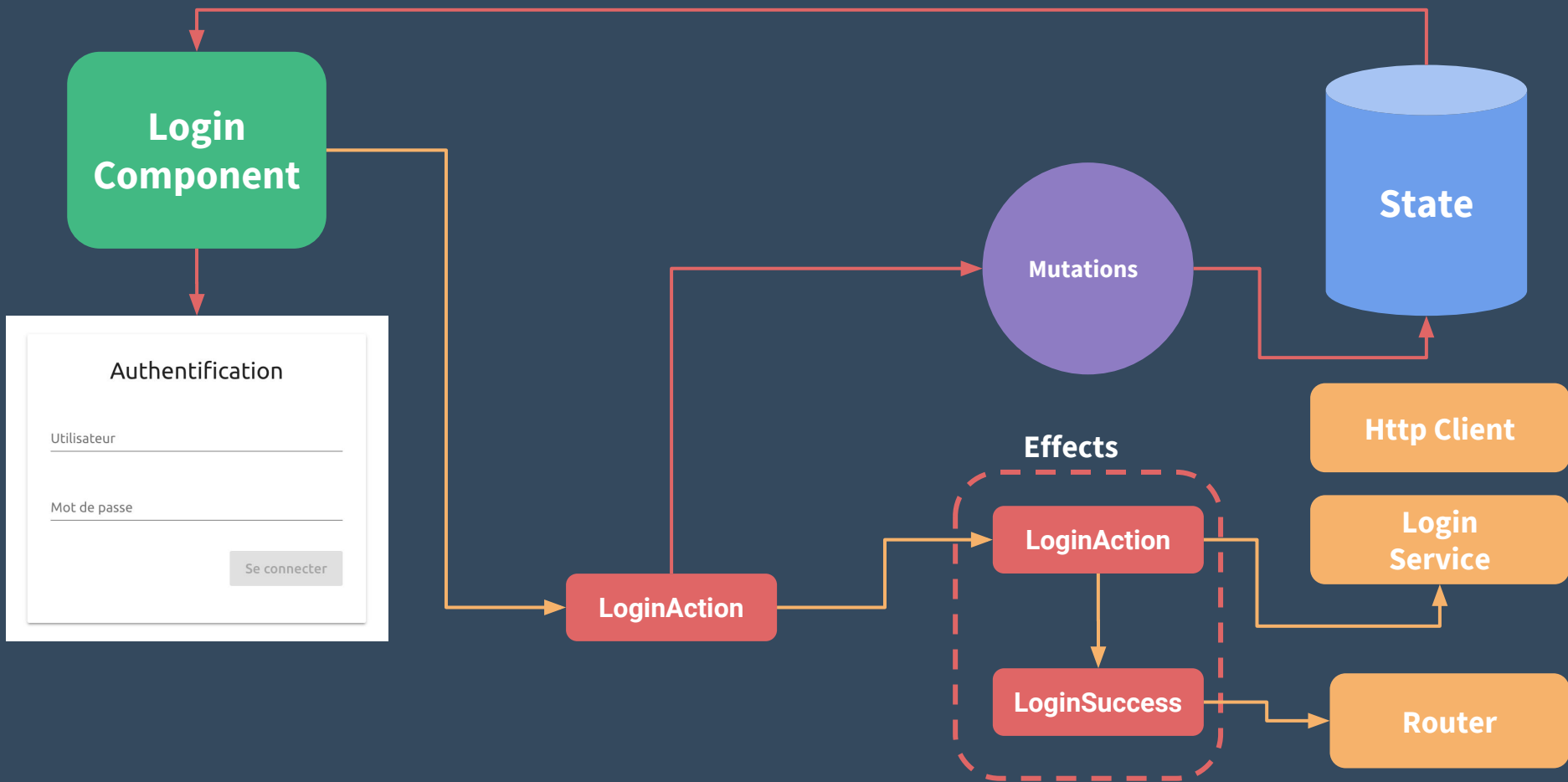
```
{  
  auth: true,  
  user: {  
    login: 'Saul Goodman',  
    isVIP: true  
  }  
}
```

Utilisation d'un State Manager

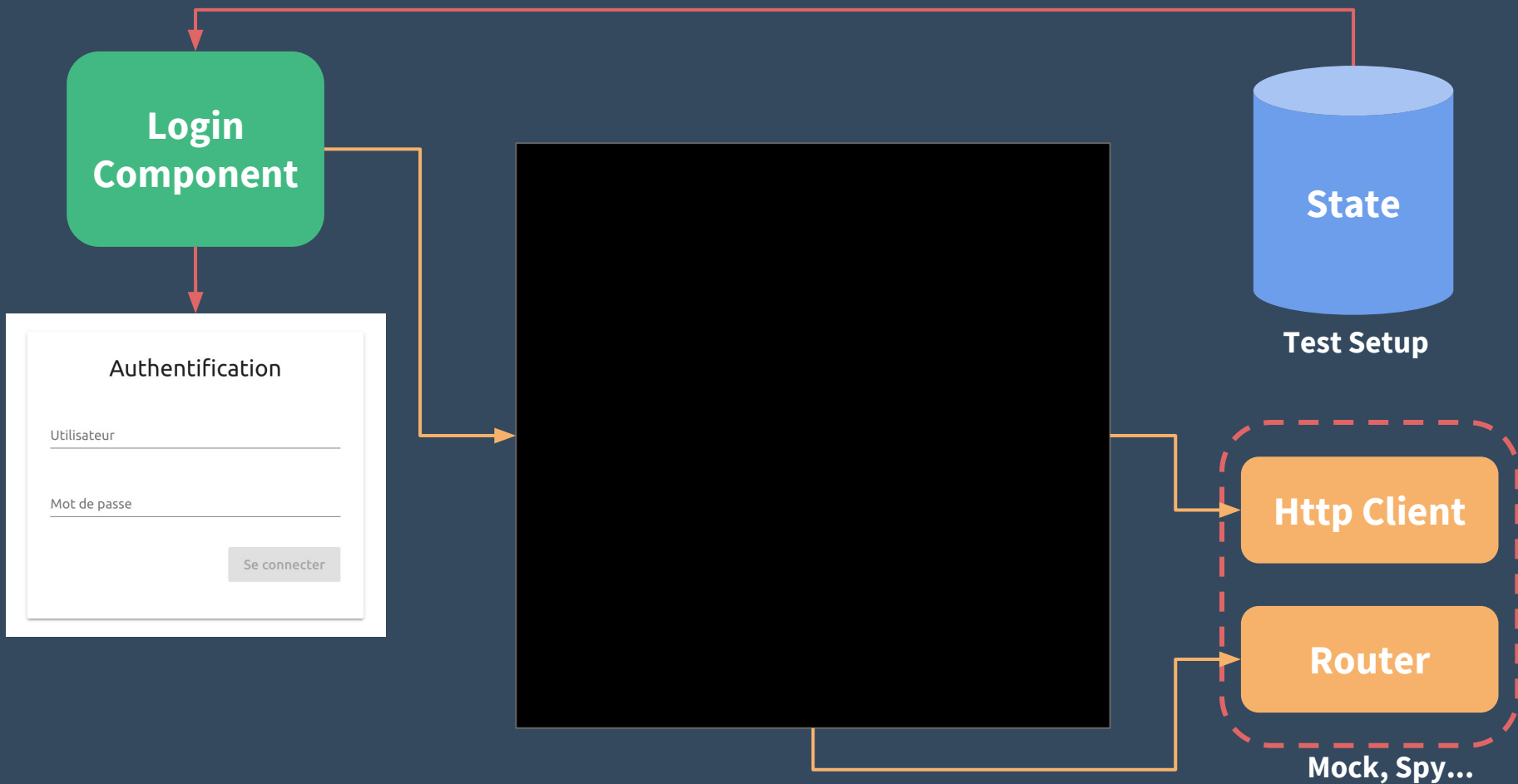


```
it('should render state properly', () => {  
  store = prepareStore(initialState);  
  navBar = compileComponenet(store);  
  
  expect(navBar).toEqual(AnonymousState);  
  
  dispatch(LoginSuccess({user: 'John Doe', isVIP: false}));  
  navBar.detectChanges();  
  
  expect(navBar).toEqual(AuthenticatedState);  
});
```

Test unitaire/intégration



Test unitaire/intégration



Test unitaire ou intégration ?



```
it('should login successfully & redirect to home', () => {  
  
  const body = {username: 'user', password: 'pass'};  
  component.submit(body);  
  
  expect(httpClient.post).toHaveBeenCalledWith('/api/authenticate', body);  
  expect(router.navigate).toHaveBeenCalledWith(['/home']);  
  
});
```

Outillage : Jest



Liran Tal

@liran_tal

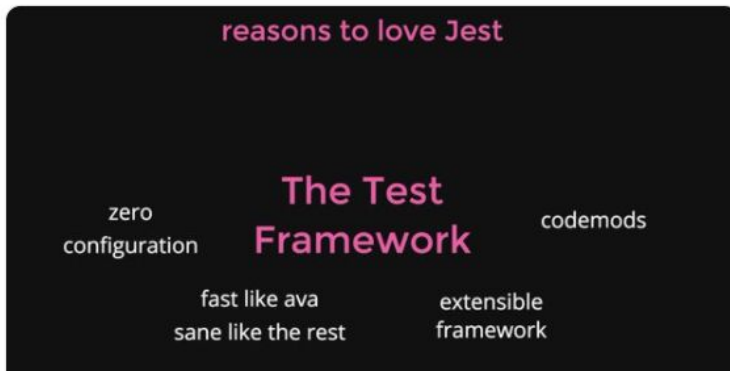
Follow



Reasons to Love Jest: The Test Framework
I enjoy writing tests, but Jest takes it to a whole new level.

The first of several posts on [@fbjest](#)
buff.ly/2yrGabD

cc [@cpojer](#) [@aaronabramov_](#) [@kentcdodds](#)
[@gautam](#) [@BenedekGagy](#)



Le passage à Jest

**Artem Sapegin**
@iamsapegin

Following

I've migrated our tests from AVA to Jest in 15 minutes thanks to codemods. Setup is very easy & it's 4 times faster.

**Migrating Ava to Jest**
Migrating from Ava to Jest
codedaily.io

1:33 AM - 14 Nov 2016

11 Retweets 36 Likes



 4  11  36  

**Kent C. Dodds**
@kentcdodds

Following

I have a habit of pushing fun new tech on my team. Not all of it works out, but people can't stop telling me how happy they are with Jest!



4:55 PM - 12 Oct 2016

16 Retweets 79 Likes



 6  16  79  

Jest + Angular



codeburst.io

Follow

Sign in

Get started

HOME

LEARN WEB DEVELOPMENT

WEB DEV COURSES

WRITE FOR US



Evgeny Barabanov

Follow




Software engineer & Open source enthusiast

Aug 21 · 2 min read





Angular 6: "ng test" with Jest in 3 minutes

Jest + Angular : Tout le monde en parle

 Angular In Depth by  AG-Grid [Follow](#)

[Sign in](#) [Get started](#)

[HOME](#) [ANGULAR](#) [RXJS](#) [NGRX](#) [ABOUT](#) [SUPPORT US](#) | [AG-GRID: THE BEST HTML5 GRID IN THE WORLD](#) 



Tim Deschryver [Follow](#)

NgRx team member — Writer for AngularInDepth

Aug 15 · 10 min read

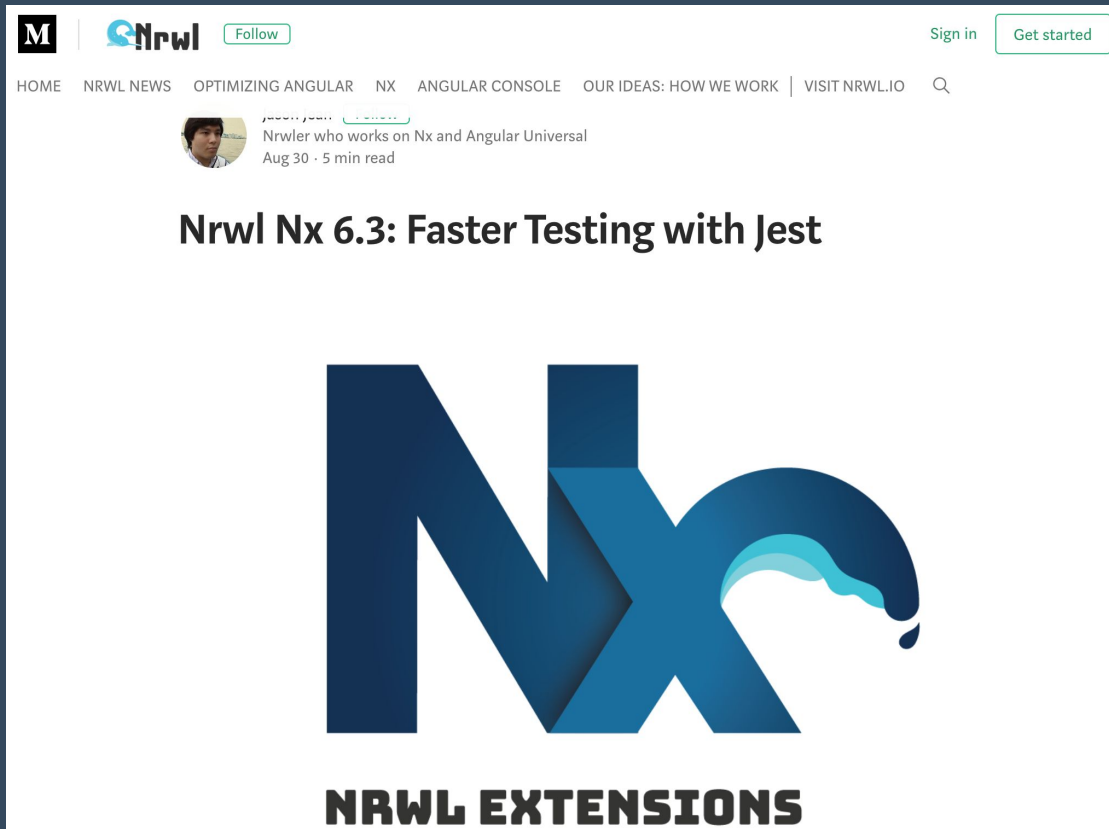
Integrate Jest into an Angular application and library

Follow me in this play-by-play guide and let's integrate Jest within your Angular project!

Why Jest

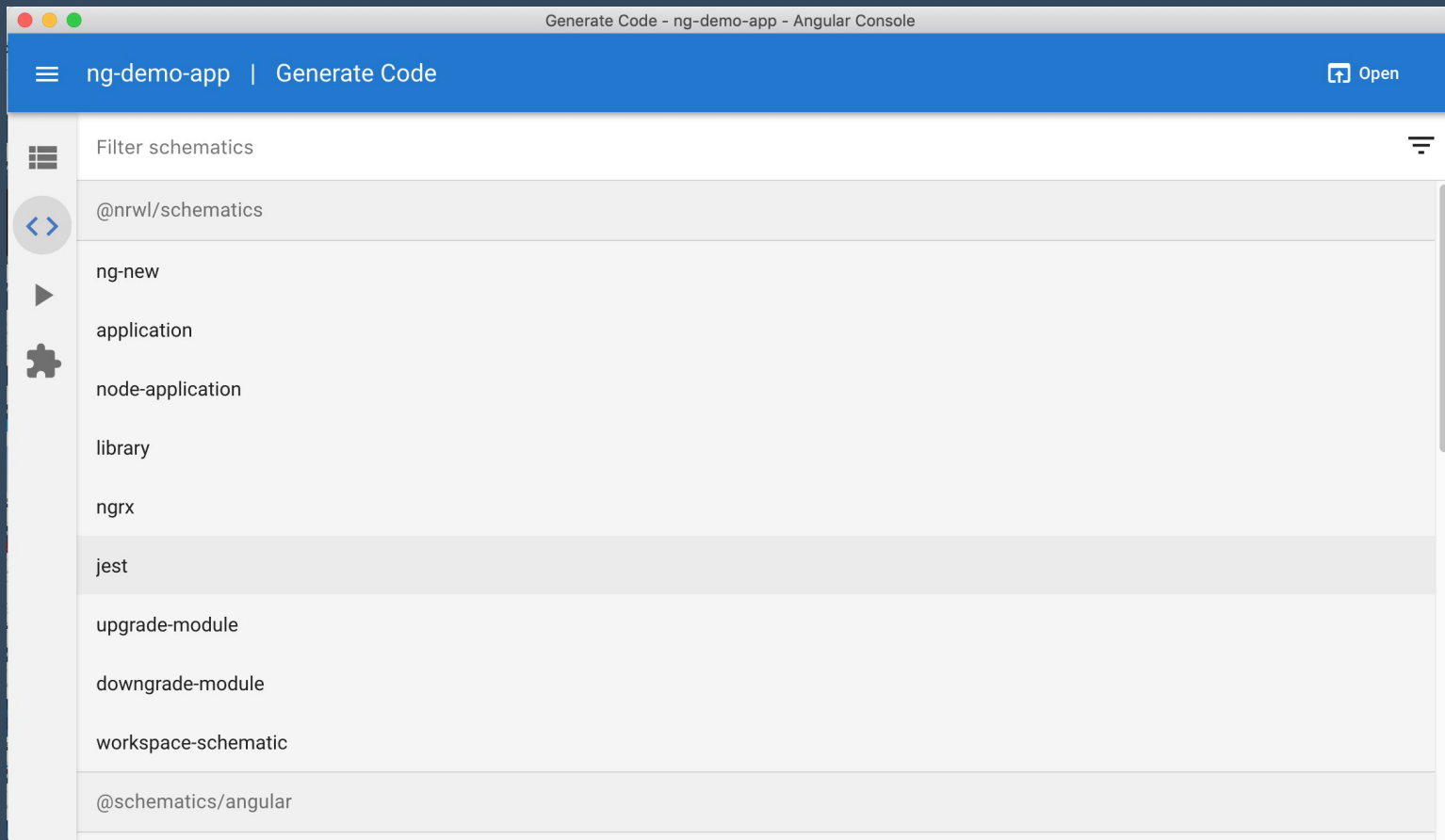
- Sensible faster; parallelized test runs
- Snapshot testing; to make sure your UI does not change unexpectedly
- Rich CLI options; only run failed tests, filter on filename and/or test name, only run related tests since the latest commit
- Readable and useful tests reports
- Sandboxed tests; which means automatic global state resets

Jest + Angular : Nrwl Nx offre Jest en option



<https://nrwl.io/nx/unit-testing-with-jest>

Jest + Angular : Via Angular Console



Jest : Snapshot Testing



```
expect(sum(1, 2)).toBe(3)
```

Vs

```
expect(sum(1, 2)).toMatchSnapshot( )
```

Jest : Snapshot Testing



```
expect(objetJs).toMatchSnapshot( )
```

```
expect(objetJson).toMatchSnapshot( )
```

```
expect( composantHTML ).toMatchSnapshot( )
```

Jest : Snapshot Testing

```
2. node
FAIL __tests__/Link.react-test.js
  ● renders correctly

    expect(value).toMatchSnapshot()

    Received value does not match stored snapshot 1.

    - Snapshot
    + Received

    <a
      className="normal"
    - href="http://www.facebook.com"
    + href="http://www.instagram.com"
      onMouseEnter={ [Function] }
      onMouseLeave={ [Function] }>
    - Facebook
    + Instagram
    </a>

    at Object.<anonymous> (__tests__/Link.react-test.js:14:16)

    x renders correctly (10ms)

Snapshot Summary
  > 1 snapshot test failed in 1 test suite. Inspect your code changes or press `u` to update
```


Jest : Interactive Snapshot Mode

Watch Usage

- > Press a to run all tests.
- > Press f to run only failed tests.
- > Press p to filter by a filename regex pattern.
- > Press t to filter by a test name regex pattern.
- > Press u to update failing snapshots.
- > Press i to update failing snapshots interactively.
- > Press q to quit watch mode.
- > Press Enter to trigger a test run.

—

Jest : Jest Snapshots and Beyond - React Conf 2017



https://www.youtube.com/watch?v=HAuXJVI_bUs

Jest : jest-extended

Sans jest-extended

```
test('hello world', () => {
  const str = 'Hello, WORLD!';
  expect(typeof str).toBe('string');
  expect(str.toLowerCase()).toBe('hello, world!');
  expect(str.split(' ')[0]).toBe('Hello,');
  expect(str.split(' ')[1]).toBe('WORLD!');
  expect(
    [...str].reduce((acc, char) => (char === 'l' ? acc + 1 : acc), 0)
  ).toBe(2);
});
```

Jest : jest-extended

Avec jest-extended

```
test('hello world', () => {  
  const str = 'Hello, WORLD!';  
  expect(str).toBeString();  
  expect(str).toEqualCaseInsensitive('hello, world!');  
  expect(str).toStartWith('Hello,');  
  expect(str).toEndWith('WORLD!');  
  expect(str).toIncludeRepeated('l', 2);  
});
```

<https://www.npmjs.com/package/jest-extended>

Jest : jest-chain

Sans jest-chain

```
test('hello world', () => {  
  const str = 'Hello, WORLD!';  
  expect(str).toBeString();  
  expect(str).toEqualCaseInsensitive('hello, world!');  
  expect(str).toStartWith('Hello,');  
  expect(str).toEndWith('WORLD!');  
  expect(str).toIncludeRepeated('l', 2);  
});
```

Jest : jest-chain

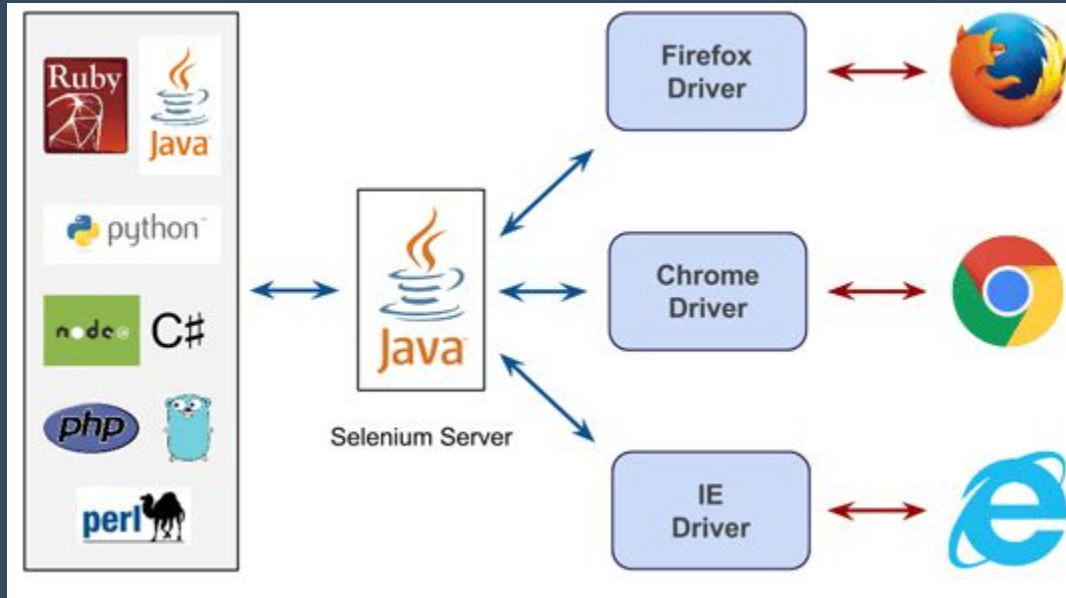
Avec jest-chain

```
test('hello world', () => {  
  const str = 'Hello, WORLD!';  
  expect(str)  
    .toBeString()  
    .toEqualCaseInsensitive('hello, world!')  
    .toStartWith('Hello,')  
    .toEndWith('WORLD!')  
    .toIncludeRepeated('l', 2);  
});
```

<https://www.npmjs.com/package/jest-chain>

Outillage : Ce qui cloche avec Protractor ?

Selenium !



Outillage : Ce qui diffère avec Cypress

The screenshot displays the Cypress test runner interface. On the left, the 'Actions' panel lists various Cypress commands like `.type()`, `.focus()`, `.blur()`, `.clear()`, `.submit()`, and `.click()`. Below this, the 'BEFORE EACH' and 'TEST' sections show a test suite with 18 steps. The main area shows the test runner with a visual representation of the test suite, including a canvas with red dots and a button labeled 'I'm being covered'. The right panel shows the 'Elements' panel with the DOM tree and the 'Styles' panel with the CSS styles for the selected element.

workspace
File Edit Window Help Developer Tools
13.49
https://example.cypress.io/commands/actions

Actions

- ✓ `.type()` - type into a DOM element
- ✓ `.focus()` - focus on a DOM element
- ✓ `.blur()` - blur off a DOM element
- ✓ `.clear()` - clears an input or textarea ele...
- ✓ `.submit()` - submit a form
- ⚙ `.click()` - click on a DOM element

BEFORE EACH

1 VISIT https://example.cypres...

TEST

- 1 GET `.action-btn`
- 2 - CLICK
- 3 GET `#action-canvas`
- 4 - CLICK
- 5 GET `#action-canvas`
- 6 - CLICK `topLeft`
- 7 GET `#action-canvas`
- 8 - CLICK `top`
- 9 GET `#action-canvas`
- 10 - CLICK `topRight`
- 11 GET `#action-canvas`
- 12 - CLICK `left`
- 13 GET `#action-canvas`
- 14 - CLICK `right`
- 15 GET `#action-canvas`
- 16 - CLICK `bottomLeft`
- 17 GET `#action-canvas`
- 18 - CLICK `bottom`

1000 x 660 (57%)

```
cy.get('#action-canvas').click('top')
cy.get('#action-canvas').click('topRight')
cy.get('#action-canvas').click('left')
cy.get('#action-canvas').click('right')
cy.get('#action-canvas').click('bottomLeft')
cy.get('#action-canvas').click('bottom')
cy.get('#action-canvas').click('bottomRight')
```

// click() accepts an x and y coordinate
// that controls where the click occurs :}
cy.get('#action-canvas')
 .click(80, 75)
 .click(170, 75)
 .click(80, 165)
 .click(180, 185)
 .click(125, 190)
 .click(180, 185)
 .click(170, 185)

// click multiple elements by passing multiple: true
cy.get('#action-labels').click({ multiple: true })

// Ignore error checking prior to clicking
cy.get('#action-opacity-btn').click({ force: true })

`.dblclick()`
To double click a DOM element, use the `.dblclick()` command.

```
cy.get('#action-div').dblclick().should('not.be.visible')
cy.get('#action-input-hidden').should('be.visible')
```

Double click to edit

element.style {
 cyypress_runner.css:5456
 body, div, d1, dt, dd,
 ul, ol, li, h1, h2, h3,
 h4, h5, h6, pre, code,
 form, fieldset, legend,
 input, button, textarea,
 p, blockquote, th, td {
 margin: 0;
 padding: 0;
 }
 cyypress_runner.css:4120
 body {
 background: #F6F6F6;
 width: 100%;
 height: 100%;
 }
 cyypress_runner.css:4050
 body, div, d1, dt, dd,
 ul, ol, li, h1, h2, h3,
 h4, h5, h6, pre, code,
 form, fieldset, legend,
 input, button, textarea,
 p, blockquote, th, td {
 margin: 0;
 padding: 0;
 }

Console Elements Sources 2691

html body

Styles Event Listeners DOM Breakpoints Properties

Filter :hov .cls +

margin -
border -
padding -
919 x 0

Filter Show all

- background-at...
- background-cl...
- background-co...
- background-im...
- background-or...
- background-po...
- background-po...
- background-re...
- background-si...
- display block
- height 0px
- margin-bottom 0px
- margin-left 0px
- margin-right 0px

Console What's New

top Filter Info 2 items hidden by filters

Les tests sont exécutés directement sur le navigateur (Pas de driver intermédiaires, facilité de débogage...)

Cypress

Avantages

- **Stabilité des assertions**
- **Fonctionne parfaitement pour les SPA full JS**
- **Améliore l'expérience développeur (GUI, Sélecteur, DevTools...)**
- **Se base sur Mocha et Chai (Syntaxe familière)**
- **Peut intercepter les requêtes HTTP sortantes avec la possibilité de les boucher**
- **Extensible via des commandes personnalisées**
- **Peut se lancer en ligne de commande sur console en mode headless**

We're officially out of beta!

📅 09-20-2018 👤 Brian Mann

[↩ Back to Cypress blogs](#)

It's been over four years since our first commit. Since then, we've seen Cypress achieve widespread and rapidly growing adoption. It's now been [downloaded more than 2.5 million times](#), with thousands of organizations recording millions of tests per month. And today, we're thrilled to announce that Cypress has officially left beta. 🎉

To keep up, we've expanded our [engineering team](#), while working hard to grow a community that helps thousands of QA engineers and developers in more than 90 countries. None of this would have been possible without the incredible support we've received from our users. You've made all of this possible and we're honored to continue building a tool and a service that makes testing an enjoyable experience.

Angular + Cypress

Angular + Cypress = Love

📅 03-20-2018 👤 Jeff Whelpley

[↩ Back to Cypress blogs](#)

This is a guest post from Jeff Whelpley, CTO at Swish, Google Developer Expert, co-creator of Angular Universal (server rendering for Angular 2+), co-organizer of Boston Angular Meetup, co-organizer of Boston AI Meetup, guest/panelist on AngularAir and Adventures in Angular podcasts, and frequent speaker at local Boston meetups.

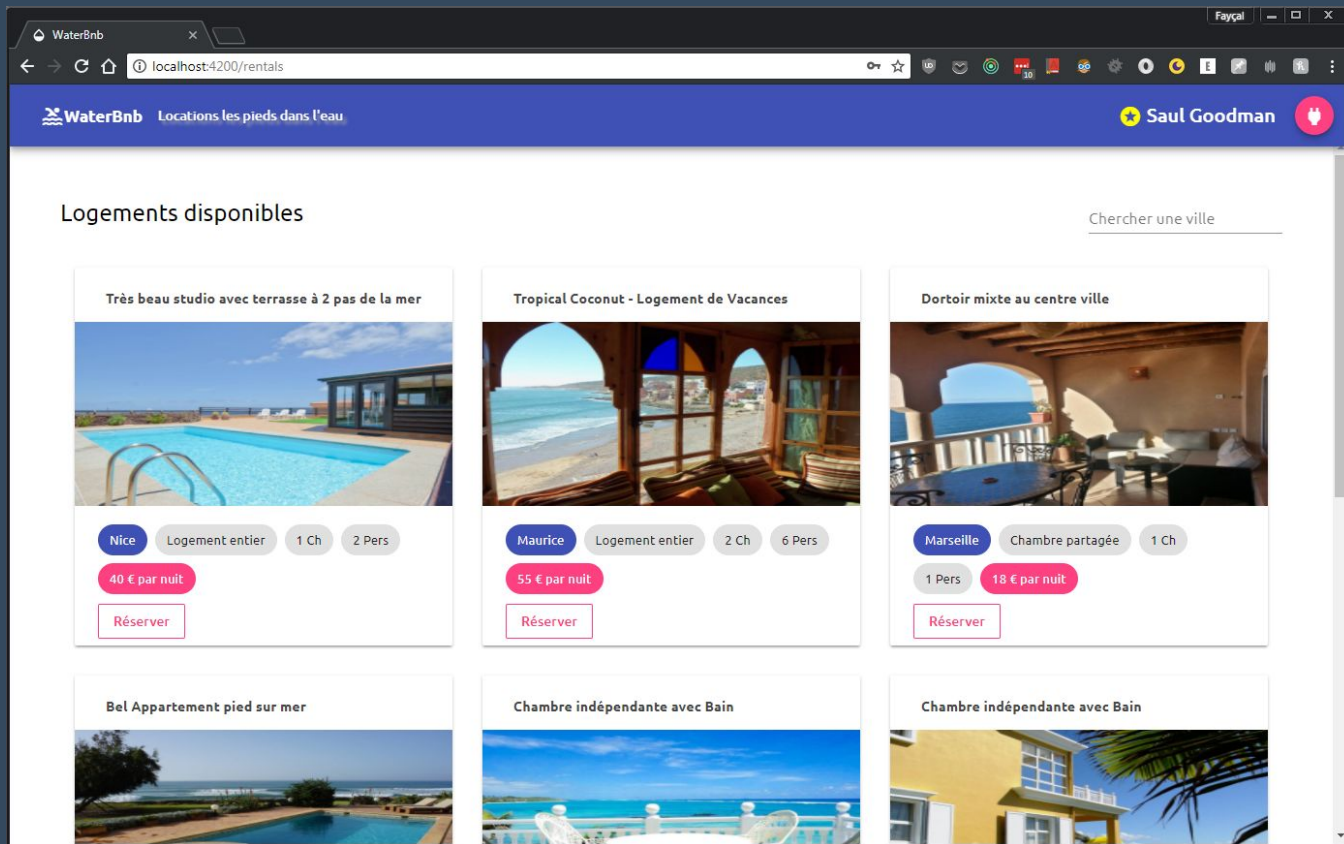
I've always been a big fan of really good all-in-one solutions like Angular. My general philosophy has been to rely on a small set of closely related technologies and only bring in something new if it is substantially more beneficial to the organization.

Contents

- [The API](#)
- [Stability](#)
- [Tooling](#)
- [Final Thought](#)

[↑ Back to top](#)

Demo



<https://github.com/fayway/tester-angular-autrement>