Des tests Javascript sans prise de tête avec Jest

Talk @ Breizhcamp 2018 - Cédric Patchane

Breizh C@mp   cozy.io

Cédric Patchane

@CPatchane

cozy.io

cozy.io

Cozy Collect

Providers

Connected accounts

Providers

- All
- Banks
- Energy
- Insurance
- ISP
- Shopping
- Telecom
- Transportation
- Others

Ameli
Insurance

EDF
Energy

Malakoff Meder...
Insurance

Cozy Banks

APPS · SETTINGS

All accounts
6 accounts

- Current balance
- Movements
- Categorization
- Settings

Movements

← May 2018 →

TOTAL	TRANSACTIONS	DEBIT	CREDIT
+3 042 €	44	-3 538 €	+6 580 €

DATE	DESCRIPTION	AMOUNT	ACTION	
22 May 2018	Bouygues Telecom	-44,90 €	View your invoice	...
21 May 2018	Fnac Ternes	-26,00 €	Accéder au site Fnac	...
21 May 2018	Franprix	-48,22 €		...
21 May 2018	Salaire	+3 870,54 €	Accéder à votre paie	...
21 May 2018	H&m	-42,89 €		...
21 May 2018	Franprix	-48,22 €		...
20 May 2018	Edf Clients Particuliers	-120,11 €	Access your space EDF	...
20 May 2018	Virement Compl Sante	+7,50 €	Check your refunds	...

Synchronisez et partagez vos fichiers et photos

Importez vos factures automatiquement

Suivez vos dépenses du quotidien

Prenez soin de votre vie privée

Les tests unitaires en Javascript

Oui, mais ...

- "Compliqués pour tester correctement l'UI"

- "Pénibles à écrire pour des fonctions complexes"

- "Compliqués avec tous nos projets différents"

- "Pénibles à écrire et maintenir dans la durée"

=> Pénibles et compliqués 🤦

Jest

Jest

- Développé par Facebook (**MIT**)

- Plus qu'un outil / framework -> une **plateforme**

- Facilite l'écriture des **tests unitaires/UI** en particulier, mais pas seulement

- Inclus plusieurs **fonctionnalités de base** dont :
  - Assertions (similaire `jasmine`)
  - *Snapshots*
  - *Mocking*
  - Couverture de code (via `istanbul`)

- Compatible **ES6** (`.babelrc` + `babel-jest`)

- **Pas restreint à React** (Vue, Angular, …)

Mise en place 😎

Installation :

```
yarn add jest —dev

npm -i --save-dev jest
```

Utilisation :

```
jest
```

```
jest --verbose
```
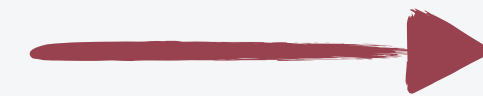
Les bases : tests unitaires

Example: fonction somme

sum.js :

```js
const sum = (a, b) ⇒ a + b

export default sum
```

sum.spec.js :

```js
import sum from '../sum'

describe('Sum library', () ⇒ {
  it('should sum 4 + 3 correctly', () ⇒ {
    expect(sum(4, 3)).toBe(7)
  })
})
```



```
cpatchane ▶ 📂 ▶ ~/Dev/perso/jest-talk 🔅
>_  yarn test:sum
yarn run v1.5.1
$ jest sum/
 PASS  sum/__test__/sum.spec.js
  Sum library
    ✓ should sum 4 + 3 correctly (4ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.001s
Ran all test suites matching /sum\//i.
✨ Done in 1.49s.
```

Example: fonction somme

sum.js :

Régression !

```js
const sum = (a, b) ⇒ a - b

export default sum
```

sum.spec.js :

```js
import sum from '../sum'

describe('Sum library', () ⇒ {
  it('should sum 4 + 3 correctly', () ⇒ {
    expect(sum(4, 3)).toBe(7)
  })
})
```

```
cpatchane    ~/Dev/perso/jest-talk
  yarn test:sum
yarn run v1.5.1
$ jest sum/
 FAIL  sum/__test__/sum.spec.js
  Sum library
    ✕ should sum 4 + 3 correctly (11ms)

  ● Sum library › should sum 4 + 3 correctly

    expect(received).toBe(expected) // Object.is equality

    Expected value to be:
      7
    Received:
      1

       6 |   describe('Sum library', () => {
       7 |     it('should sum 4 + 3 correctly', () => {
    >  8 |       expect(sum(4, 3)).toBe(7)
       9 |     })
      10 |   })
      11 |

      at Object.<anonymous> (sum/__test__/sum.spec.js:8:23)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 total
Snapshots:   0 total
Time:        1.381s
Ran all test suites matching /sum\//i.
```

Fonction `searchFantasyName() - v1`

searchFantasyName.js :

```js
import names from './names'

function searchFantasyName (query)
  return names.filter(
    name ⟹ name.search(
        new RegExp(query, 'i')
    ) ≢ -1
  )
}


export default searchFantasyName
```

Exemple :

searchFantasyName('Mark')
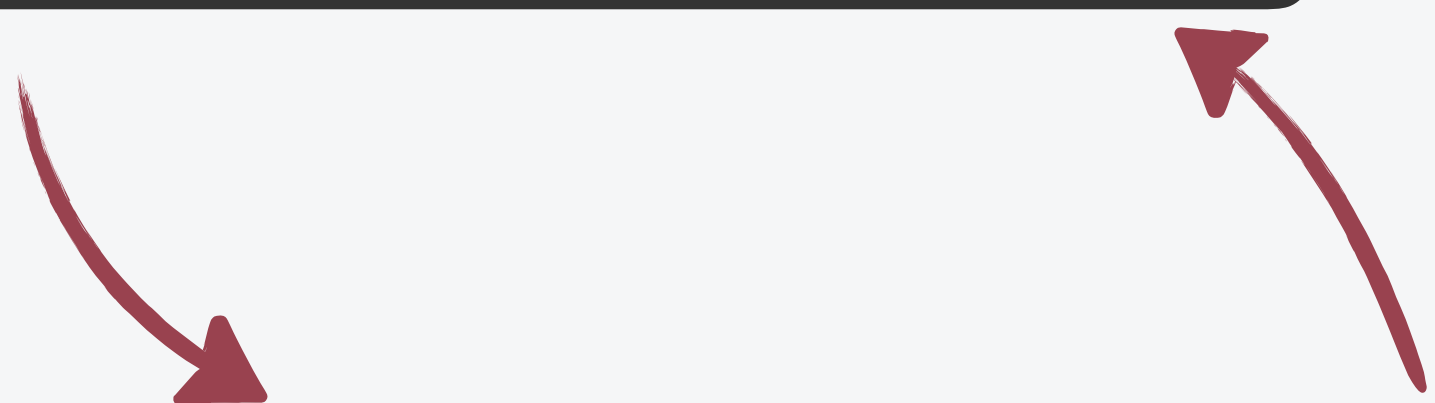['Markke']

names.js :

```js
export default [
  'Lafchris',
  'Richcome',
  'Markke',
  'Retferth',
  'Nathniel',
  'Laydun',
  'Sigcuth',
  'Thy',
  'Ceolliam',
  'Ard',
  'Cwenkev',
  'Palau',
  'Thasrichham',
  'Misere',
  'Dinoke',
  'Ruthbar',
  'Thochael',
  'Muelsig',
  'Cas',
  'Eadonbri'
]
```

Fonction `searchFantasyName()` - v1

searchFantasyName.spec.js :

```javascript
import searchFantasyName from '../searchFantasyName'

describe('searchFantasyName', () ⇒ {
  it('return results from "am" query', () ⇒ {
    console.log(searchFantasyName('am'))
    expect(searchFantasyName('am')).toEqual()
  })
})
```

```
console.log searchFantasyName/test/searchFantasyName.spec.js:8
  [ 'Ceolliam', 'Thasrichham' ]
```

Fonction `searchFantasyName()` - v1

searchFantasyName.spec.js :

```javascript
import searchFantasyName from '../searchFantasyName'

describe('searchFantasyName', () ⇒ {
  it('return results from "am" query', () ⇒ {
    expect(searchFantasyName('am')).toEqual([ 'Ceolliam', 'Thasrichham' ])
  })
})
```
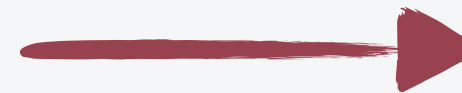
```
PASS  searchFantasyName/test/searchFantasyName.spec.js
  searchFantasyName
    ✓ return results from "am" query (3ms)


Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.961s, estimated 1s
Ran all test suites matching /searchFantasyName\//i.
✨ Done in 1.49s.
```

Fonction `searchFantasyName()` - v2

`names.js` :

```
diff --git a/names.js b/names.js
index a11d284..213eb22 100644
--- a/names.js
+++ b/names.js
@@ -19,4 +19,13 @@ export default [
    'Muelsig',
    'Cas',
    'Eadonbri',
+   'Amarr',
+   'Etuari',
+   'Fresdam',
+   'Naldbert',
+   'Gard',
+   'Mondan',
+   'Georna',
+   'Wardlaf',
+   'Ricro'
 ]
```

searchFantasyName.spec.js →

```
FAIL  searchFantasyName/test/searchFantasyName.spec.js
searchFantasyName
  ✕ return results from "am" query (13ms)

● searchFantasyName › return results from "am" query

  expect(received).toEqual(expected)

  Expected value to equal:
    ["Ceolliam", "Thasrichham"]
  Received:
    ["Ceolliam", "Thasrichham", "Amarr", "Fresdam"]

  Difference:

  - Expected
  + Received

    Array [
      "Ceolliam",
      "Thasrichham",
  +   "Amarr",
  +   "Fresdam",
    ]
```

Ajout de nouveaux noms dans la liste = tests en échec

Workflow de mise à jour des tests

Exécuter les tests

Tests en succès

Tests en échec

Mettre à jour les tests

(1) (2) (3) (4)

Beaucoup de `console.log` et de copier/coller

=> tests pénibles à maintenir dans la durée

Les snapshots
à la rescousse 📸

Comment l'utiliser?

```
expect(searchFantasyName('am')).toEqual([ 'Ceolliam', 'Thasrichham' ])
```
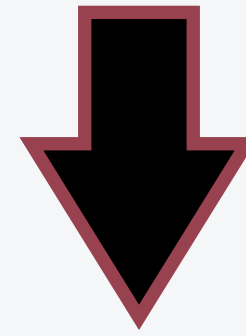
```
expect(searchFantasyName('am')).toMatchSnapshot()
```

Comment cela fonctionne ?

```
expect(searchFantasyName('am')).toMatchSnapshot()
```

Récupérer

Valeur passée à `expect()`

Vérifier

Déjà un snapshot existant ?

Oui → Comparer au snapshot existant

Non → Sauvegarder dans un fichier `.snap`

Différent → Test en échec

Identique → Test en succès

Sauvegarder dans un fichier `.snap` → **Test en succès**
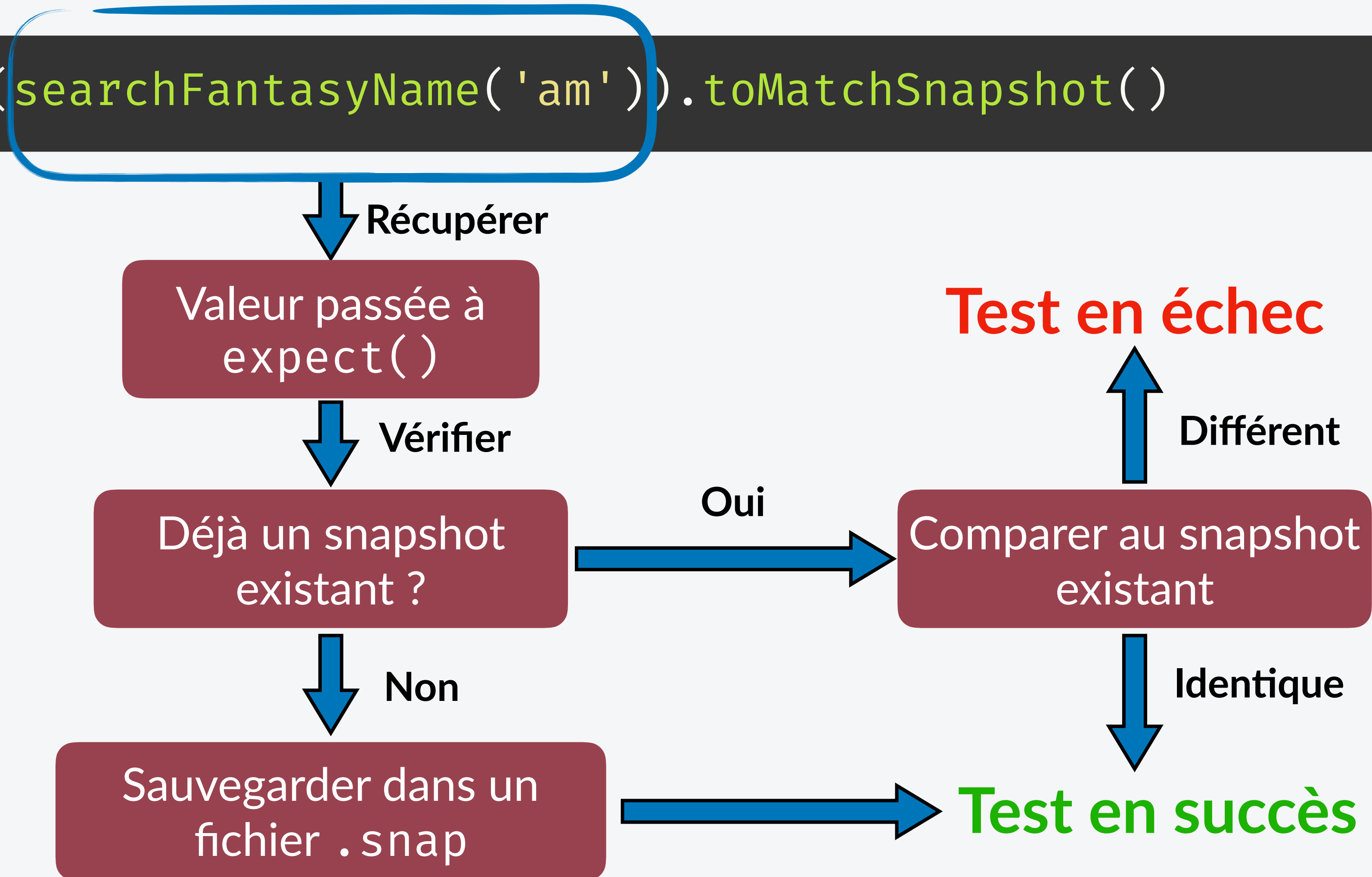
```
Snapshot Summary
› 1 snapshot written in 1 test suite.
```

Comment cela fonctionne ?

```
expect(searchFantasyName('am')).toMatchSnapshot()
```

1ère exécution (v1)

```
exports[`searchFantasyName return
results from "am" query 1`] = `
Array [
  "Ceolliam",
  "Thasrichham",
]
`;
```

```
Snapshot Summary
› 1 snapshot written in 1 test suite.
```

2ème exécution (v2)

```
- Snapshot
+ Received

  Array [
    "Ceolliam",
    "Thasrichham",
+   "Amarr",
+   "Fresdam",
  ]
```

jest -u (v2)

```
exports[`searchFantasyName
return results from "am" query
1`] = `
Array [
  "Ceolliam",
  "Thasrichham",
  "Amarr",
  "Fresdam",
]
`;
```

```
PASS  searchFantasyName/test/searchFantasyName.spec.js
  searchFantasyName
    ✓ return results from "am" query (3ms)

› 1 snapshot updated.
```

v2 = searchFantasyNames v1 + ajout des nouveaux noms

Les tests de composants (tests unitaires d'UI)

Top 30 Hacker News

To Change Habits, Try Replacement Instead by ALee

The need for tooling and the need for mastering your tools by r4um

Storyboarder: free, cross-platform storyboard tool by open-source-ux

Google Cloud Text-To-Speech Powered by DeepMind WaveNet Technology by pseudobry

Uber will not re-apply for self-driving car permit in California by ExcelSaga

Show HN: Launched my first app where people hum songs for others to guess by depomoty

Fiber Lasers Mean Ray Guns Are Coming by sohkamyung

Waymo teams up with Jaguar to intro a new, premium self-driving car by kaboro

How I Erased Facebook Comments and Likes by Jaruzel

Charles Proxy now available on iOS by eddyg

Update on Async Rendering by runesoerensen

Fast, Safe, and Complete(ish) Web Service in Rust by henridf

<StoriesList />

StoriesList.jsx :

```jsx
import React from 'react'

const StoriesList = (props) => {
  const { stories } = props
  return (
    <div className='stories'>
      {stories.map((s, i) => (
        <div key={i}>
          <a href={s.url} target='_blank'>
            <div>
              {`${s.title} by ${s.by}`}
            </div>
          </a>
        </div>)
      )}
    </div>
  )
}


export default StoriesList
```

<StoriesList />

Rendu HTML :

```html
<div
  class="stories"
>
  <div>
    <a
      href="http://mock.example.com/mock1"
      target="_blank"
    >
      <div>
        Story number one by MissOne
      </div>
    </a>
  </div>
  <div>
    <a
      href="http://mock.example.com/mock2"
      target="_blank"
    >
      <div>
        Story number two by MisterTwo
      </div>
    </a>
  </div>
</div>
```

```
<StoriesList
  stories={stories}
/>
```

ReactJS
~
VueJS
Angular
Inferno
Preact

Tester avec un *serializer*

Composant React :

```
import Enzyme, { shallow } from 'enzyme'
import Adapter from 'enzyme-adapter-react-16'
Enzyme.configure({ adapter: new Adapter() })

const component = shallow(
  <StoriesList stories={mockStories} />
).getElement()
```

Composant Vue :

```
import { shallow } from '@vue/test-utils'

const component = shallow(StoriesList, {
  propsData: { mockStories }
}).html()
```

Assertion

```
expect(component).toMatchSnapshot()
```

Démo 👨🏻‍💻

Pour aller plus loin 🤔

Mocking

jest.fn(implementation)

```javascript
import logSum from '../logSum'
import * as sumModule from '../sum'

describe('logSum library', () ⇒ {
  it('call sum correctly', () ⇒ {
    sumModule.default = jest.fn(sumModule.default)
    logSum(3, 4)
    expect(sumModule.default).toHaveBeenCalledTimes(1)
    expect(sumModule.default).toHaveBeenCalledWith(3, 4)
  })
})
```

```
PASS  sum/test/logSum.spec.js
  ● Console

    console.log sum/logSum.js:6
      7

PASS  sum/test/sum.spec.js

Test Suites: 2 passed, 2 total
Tests:       2 passed, 2 total
```

Couverture de code

istanbul

`jest --coverage`

```
----------------------------|---------|----------|---------|---------|-------------------
File                        | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------------|---------|----------|---------|---------|-------------------
All files                   |   67.19 |       40 |      80 |   67.19 |
 src/components             |   45.45 |        0 |   33.33 |   45.45 |
  App.vue                   |     100 |      100 |     100 |     100 |
  Icon.vue                  |       0 |        0 |       0 |       0 | ... 40,41,44,45,47
 src/components/HelloViews  |     100 |      100 |     100 |     100 |
  Hello1.vue                |     100 |      100 |     100 |     100 |
  Hello2.vue                |     100 |      100 |     100 |     100 |
  Hello3.vue                |     100 |      100 |     100 |     100 |
 src/lib/I18n               |      55 |       80 |      80 |      55 |
  format.js                 |      50 |       80 |      50 |      50 |       9,10,12,16
  index.js                  |     100 |      100 |     100 |     100 |
  translation.js            |   44.44 |       80 |     100 |   44.44 |    18,19,20,21,23
 test                       |     100 |      100 |     100 |     100 |
  app.spec.js               |     100 |      100 |     100 |     100 |
 test/ducks/components      |     100 |      100 |     100 |     100 |
  helloViews.spec.js        |     100 |      100 |     100 |     100 |
 test/jestLib              |     100 |      100 |     100 |     100 |
  setup.js                  |     100 |      100 |     100 |     100 |
 test/lib                   |     100 |      100 |     100 |     100 |
  I18n.js                   |     100 |      100 |     100 |     100 |
----------------------------|---------|----------|---------|---------|-------------------
```

Vue détaillée dans : `coverage/lcov-report/index.html`

Test Driven Development (TDD) ?

- *Snapshoting* pas vraiment conçu pour du TDD

- Mais : `expect.addSnapshotSerializer(serializer)`

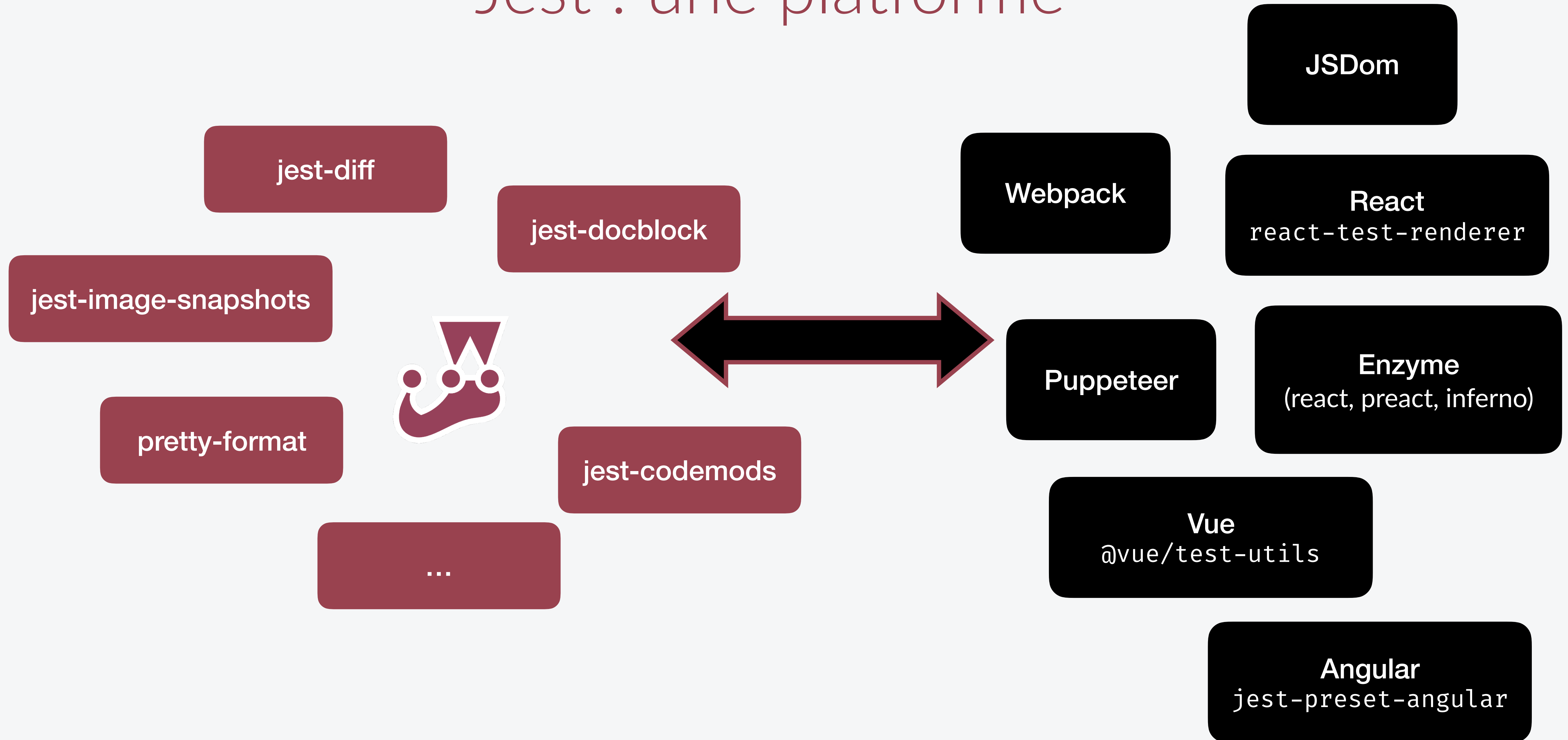**Snapshots :**

```
[
  {
    id: 1,
    title: 'Story number one with B',
    by: 'MissOne',
    score: 189,
    url: 'http://mock.example.com/mock1'
  },
  {
    id: 2,
    title: 'Story number two with B',
    by: 'MisterTwo',
    score: 145,
    url: 'http://mock.example.com/mock2'
  }
]
```

⟶

```
[
  'Story number one with B',
  'Story number two with B'
]
```

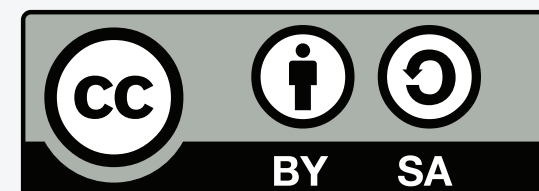**=> snapshot plus facile à éditer en amont**

Jest : une platforme

jest-diff

jest-docblock

jest-image-snapshots

pretty-format

jest-codemods

...

JSDom

Webpack

React
`react-test-renderer`

Puppeteer

Enzyme
(react, preact, inferno)

Vue
`@vue/test-utils`

Angular
`jest-preset-angular`

Merci !

 github.com/CPatchane/jest-talk

 @CPatchane

Sources images :

Logo Jest : https://github.com/facebook/jest
Fond (Diap. 1) : Photo by Sergey Zolkin on Unsplash