

Cadriciel Web côté client

Cours 12 - Tests Unitaires

Steve Lévesque, Tous droits réservés © où applicables

Table des matières

- 1 Utilité des tests unitaires Frontend
- 2 Tests unitaires Frontend avec Jest (next/jests.js)
- 3 Tests unitaires d'interface (Render)
- 4 Tests unitaires d'état (Snapshots)

Utilité des tests unitaires Frontend

Traditionnellement, les tests unitaires sont au Backend et permettent de tester les fonctions.

Maintenant, avec les avancées au niveau des technologies Frontend, il est maintenant possible de faire des tests unitaires pour valider l'intégrité de composantes graphiques.

Tests unitaires Frontend avec Jest (next/jests.js)

Il y a deux catégories de tests :

- d'interface (render) : pour valider les éléments actuels de l'interface avec une logique algorithmique au besoin
- d'état (snapshot) : pour valider s'il l'interface à changé vis-à-vis une sauvegarde précédente dénotée comme “pure”

Tests unitaires d'interface (Render)

Le principe :

- Créer un fichier par page
- Importer la page à tester
- Cibler l'élément graphique à tester
- Assurer un couvrage le plus global possible en ciblant plusieurs éléments graphiques

Tests unitaires d'interface (Render)

```
describe("Page email", () => {  
  it("renders a email header", () => {  
    render(<Page />);  
  
    const item = screen.getByText(/email address/i);  
    expect(item).toBeInTheDocument();  
  });  
});  
  
describe("Page password", () => {  
  it("renders a password header", () => {  
    render(<Page />);  
  
    const item = screen.getByText(/password/i);  
    expect(item).toBeInTheDocument();  
  });  
});
```

Figure: Des tests d'interfaces (Render) pour la page de login, le /i veut dire "insensible à la case" en RegEx (Regular Expression)

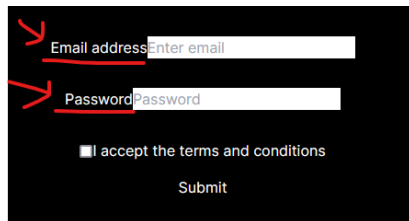


Figure: L'interface de la page de login

Tests unitaires d'état (Snapshots)

Que peut-on faire si on veut s'assurer qu'un développeur mandaté de continuer notre implémentation ne brise rien, surtout s'il est junior/stagiaire ?

On peut prendre une sauvegarde (Snapshot) de notre page et inclure cette validation comme un test intégratif de notre pile de tests de validation.

Si quelqu'un change le code de l'interface lorsqu'il n'est pas supposé le faire, le test snapshot va échouer.

Pour qu'une page soit considérée comme non modifiable, il suffit de créer le snapshot dessus et d'inclure le test dans la pile de tests de validation à exécuter.

Tests unitaires d'état (Snapshots)

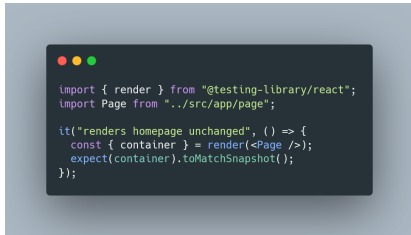


Figure: Un test d'états (Snapshot) pour la page de login

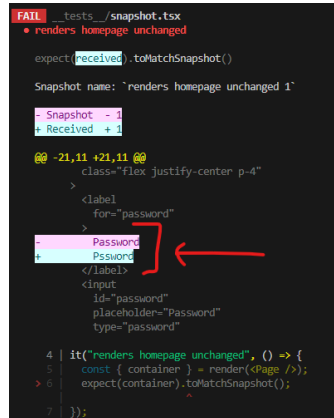


Figure: Une erreur à cause d'un changement d'interface non souhaité par le développeur initial

Bibliographie

- <https://testing-playground.com/>
- <https://jestjs.io/>