

Le Test Driven Development et les Tests Unitaires

Jérémy A. < jp1.eu >

Le principe de base du TDD

Le **TDD** (Test Driven Development) est une manière de développer des logiciels qui met **les tests** (notamment unitaires) **en priorité** par rapport aux premières phases d'écriture de code.

On écrit d'abord les tests et **ensuite** on commence à coder les fonctionnalités correspondantes.

Les étapes du TDD

- Écrire un test
- Lancer le test (vérifier qu'il échoue)
- Écrire le code nécessaire pour que le test passe
- Optimiser le code écrit

Les Tests Unitaires

Un test unitaire c'est quoi?

Un **test unitaire** a pour but (comme son nom l'indique) de **tester une "unité" d'un programme.**

Ce qu'on appelle une "unité" dans un programme peut être une **méthode, une fonction, une classe, même quelque lignes seulement...**

Si l'"unité" qui doit être testée dans un test unitaire n'est pas nécessairement définie et que le concept est relatif, le but d'un test unitaire est quand à lui assez précis !

À quoi ça sert ?

Le but d'un test unitaire est de tester, que pour un comportement **A** (une variable, un évènement), l'unité de code testée ait un comportement **B** strictement défini.

L'avantage est de pouvoir s'assurer à chacune des étapes du développement que les fonctionnalités du projet sont conformes à la manière dont elles ont été architecturées et également de pouvoir éviter et repérer de nombreuses régressions dans le code !

Exemple

Prenons le cas d'une fonction **checkPassword** qui servirait à valider le mot de passe d'un utilisateur.

```
function checkPassword(password: string) {  
}
```

Elle doit renvoyer une erreur dans ces conditions :

- Le mot de passe fait moins de 8 caractères
- Le mot de passe est un mot du dictionnaire
- Le mot de passe ne comporte que des chiffres

Nous pourrions écrire les tests de la manière suivante :

```
describe('checkPassword', () => {  
  // We start by checking the first rule  
  it("shouldn't allow passwords shorter than 8 chars",  
    expect(checkPassword("test1")).toThrow();  
  });  
  // Then the second one  
  it("shouldn't allow words from the dictionary", () =>  
    expect(checkPassword("aerodynamic")).toThrow();  
  });  
  // And finally the third one  
  it("shouldn't allow only numeric passwords", () => {  
    expect(checkPassword("8008135707")).toThrow();  
  });  
})
```