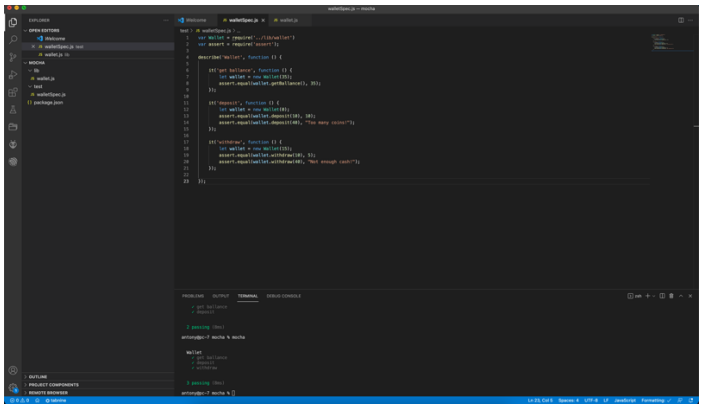
SIO 2	Les tests unitaires. Partie 1	2021/2022

Écrivez des tests unitaires pour valider votre modèle



Le contexte

- Introduction

Dans ce chapitre, nous allons écrire les premiers tests unitaires pour valider le modèle de **CoinCoinTracker** (Notre futur application mobile) Le modèle ? Quel modèle ? Eh oui, rappelez-vous : nous privilégions une approche TDD (Test Driven Development), nous commençons donc par écrire les tests unitaires et nous écrirons ensuite le code du modèle pour valider ces tests.

Pour rappel **CoinCoinTracker** est une application de gestion de porte-monnaie virtuel. Au lieu de mettre des vraies pièces dans votre porte-monnaie, vous allez ajouter de l'argent via cette application mobile. Rien d'extraordinaire, mais grâce à cette application, je suis sûr que vous vous sentirez plus riches !

- Les fonctionnalités de l'application

Avant d'écrire les tests pour valider notre modèle, encore faut-il se poser la bonne question : quelles sont les fonctionnalités offertes par mon modèle ? Quelques réponses possibles :

- Mettre de l'argent dans le porte-monnaie ;
- Retirer de l'argent du porte-monnaie ;
- Consulter la somme disponible dans le porte-monnaie.

Nous pourrons également tester certains comportements, tels que :

- Ne pas pouvoir retirer plus que le montant disponible (cela paraît évident, mais c'est important !) ;
- Ne pas ajouter plus qu'un certain montant car le porte-monnaie dispose d'une capacité de **30**.

- Environnement utilisé

Nous allons voir comment installer et créer nos premiers tests en utilisant le Framework **Mocha**. Pour cela nous utiliseront l'environnement JAVASCRIPT suivant :

- Nodejs et npm (présents sur nos pc)
- Mocha
- Visual Studio Code (présent sur nos pc)

- Installation de l'environnement

Commençons par créer un dossier nommé **« CoinCoinTracker »** dans notre dossier personnel.

Nous allons ensuite ouvrir ce dossier dans Visual Studio Code.

Puis ouvrir une fenêtre de terminal dans VSC et saisir la commande suivante :

- ***npm init***

L'installation de mocha se fait très simplement en utilisant la commande npm.

- ***npm install -g mocha***

Lors de son exécution, mocha va chercher les fichiers contenants les tests unitaires dans le dossier **« test »** de notre application. Nous allons donc créer ce dossier.

Nous allons aussi créer dans ce dossier **« test »** un fichier nommé **« wallet.js »** et l'ouvrir dans l'éditeur. C'est le fichier dans lequel nous allons écrire nos tests.

Notre premier test

Maintenant que **mocha** est installé sur notre système on va pouvoir créer notre premier test. Cette interface commence par le mot-clé ***describe*** qui permet de décrire en quoi va consister notre test, suivi des fonctions ***it*** qui permettront de décrire un scénario de test.

```
describe('Multiplication', function(){
    it('should work', function(){

    });
});
```

En l'état, le test n'est pas réellement utile, il nous faut gérer ce que l'on appelle des assertions. Ces assertions nous permettent de tester qu'une valeur correspond bien à une autre et de vérifier que ce que l'on attend est bien ce que l'on obtient. Pour le moment, nous allons nous contenter du Framework d'assertion de NodeJS.

```
var assert = require('assert');
describe('Multiplication', function(){
    it('should work', function(){
        assert.equal(2*3, 6);
    });
});
```

Insérons le code ci-dessus dans notre fichier -> enregistrer

Dans la fenêtre de terminal, démarrons l'exécution du test par la commande :

- ***mocha***

Quels est le résultat du test ?

Si nous remplaçons la ligne :

```
assert.equal(2*3, 6);
```

Par la ligne :

```
assert.equal(2*3, 8);
```

Quels est le résultat du test ?