



Mocha

Alvin Berthelot

Version 1.0.0



Ce(tte) œuvre est mise à disposition selon les termes de la Licence Creative Commons.

Attribution - Partage dans les Mêmes Conditions 3.0 non transposé.



La licence, ses explications ainsi que les moyens de contribution et réappropriation sont détaillés à la fin.

Fonctionnement basique de Mocha

Définition de Mocha

Mocha est une librairie pouvant exécuter des tests ("test runner") à la fois dans un environnement Node.js et dans un navigateur web.

Elle permet de réaliser des tests synchrones et asynchrones.

Elle ne dispose pas de librairie d'assertions par défaut.

Installation de Mocha

La librairie Mocha est généralement exécutée via un environnement [Node.js](#), elle est donc installée comme un module npm.

Soit de manière globale.

```
npm install --global mocha
```

Soit de manière locale.

```
npm install --save-dev mocha
```

Installation d'une librairie d'assertions

La librairie Mocha n'inclut pas de librairie d'assertions par défaut. Si il est possible d'utiliser [la librairie d'assertion par défaut de Node.js](#), il est généralement conseillé d'en installer une plus complète :

- [Chai](#).
- [should.js](#).
- [expect.js](#).



[Chai](#) étant la plus complète, c'est celle-ci que l'on retiendra pour la suite.

Installation de Chai

La librairie Chai est exécutée sur le même environnement que Mocha donc via un environnement [Node.js](#), elle est donc installée comme un module npm.

Soit de manière globale.

```
npm install --global chai
```

Soit de manière locale.

```
npm install --save-dev chai
```

Utilisation de Mocha et Chai

Une fois les librairies Mocha et Chai accessibles, il convient d'écrire les tests appropriés.

```
// file test/hello.test.js

var expect = require('chai').expect;

describe('Hello world testing !', function() {
  it('should be true', function() {
    expect(true).to.be.true;
  })
});
```

Pour ensuite pouvoir les lancer.

```
mocha test/hello.test.js
```


Écriture des tests avec Mocha

Les tests seront regroupés logiquement avec la fonction `describe()` et chaque test sera exécuté avec la fonction `it()`. Tous ayant un libellé.

```
// file test/hello.test.js

var expect = require('chai').expect;

describe('GIVEN a list of 4 colours: white, blue, green, red', function() {
  var colours = ['white', 'blue', 'green', 'red'];
  it('WHEN the client request the list THEN 4 items are returned', function() {
    expect(colours.length).to.equal(4);
  })
  it('WHEN the client request the second item THEN the item expected is blue', function() {
    expect(colours[1]).to.equal('blue');
  })
});
```



Il est possible d'avoir plusieurs niveaux de regroupement avec `describe()`.

Se faciliter l'écriture des tests avec Mocha

Il n'est pas toujours aisé d'écrire des tests et de les lancer si de nombreux tests existent déjà. Cela peut être long et contraignant.

Pour palier cela, Mocha propose les fonctions `skip()` et `only()` qui permettent respectivement de ne pas exécuter les tests mentionnés ou à l'inverse de n'exécuter que ceux mentionnés.

```
describe('GIVEN a list of 4 colours: white, blue, green, red', function() {  
  var colours = ['white', 'blue', 'green', 'red'];  
  it.skip('WHEN the client request the list THEN 4 items are returned', function() {  
    expect(colours.length).to.equal(4);  
  });  
  it('WHEN the client request the second item THEN the item expected is blue', function() {  
    expect(colours[1]).to.equal('blue');  
  });  
});
```

Fonctionnement de Chai

Il existe 3 syntaxes d'assertions disponibles avec Chai : Assert, Expect, Should.

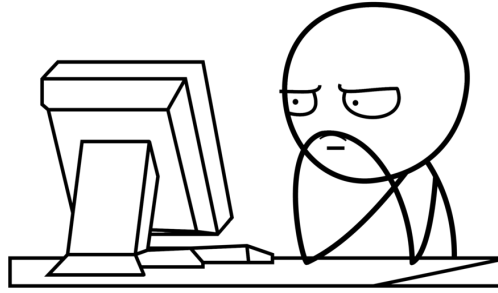
```
var assert = require('chai').assert;
var expect = require('chai').expect;
var should = require('chai').should();

// Assert
it('WHEN the client request the list THEN 4 items are returned', function() {
  assert.lengthOf(colours, 4);
})

// Expect
it('WHEN the client request the list THEN 4 items are returned', function() {
  expect(colours).to.have.lengthOf(4);
})

// Should
it('WHEN the client request the list THEN 4 items are returned', function() {
  colours.should.have.lengthOf(4);
})
```

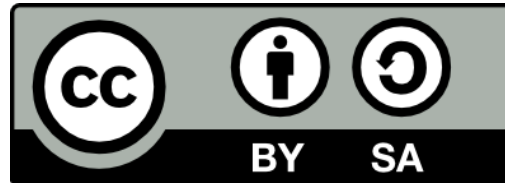
Ces 3 syntaxes réalisent les mêmes traitements, c'est uniquement la manière de les exprimer qui diffère. Expect et Should prônant une [approche BDD](#), plus fluide car s'apparentant à un langage naturel.



Exercices

Appuyez vous sur le projet [workshop-mocha-noob](#) pour réaliser cet exercice.

Licence



CC BY-SA 3.0

Ce(tte) œuvre est mise à disposition selon les termes de la Licence Creative Commons. Attribution - Partage dans les Mêmes Conditions 3.0 non transposé.

Copyright © 2017 [Alvin Berthelot](#).

Pour toutes questions, réclamations ou remarques, merci d'envoyer un message à alvin.berthelot@webyousoon.com.

Explications licence CC BY-SA 3.0

Cette licence permet aux autres de remixer, arranger, et adapter votre œuvre, même à des fins commerciales, tant qu'on vous accorde le mérite en citant votre nom et qu'on diffuse les nouvelles créations selon des conditions identiques.

Cette licence est souvent comparée aux licences de logiciels libres, "open source" ou "copyleft".

Toutes les nouvelles œuvres basées sur les vôtres auront la même licence, et toute œuvre dérivée pourra être utilisée même à des fins commerciales.

C'est la licence utilisée par Wikipédia ; elle est recommandée pour des œuvres qui pourraient bénéficier de l'incorporation de contenu depuis Wikipédia et d'autres projets sous licence similaire.

Contribution et réappropriation

Ce fichier PDF est généré avec [Asciidoctor](#) à partir d'un dépôt Git se trouvant sous GitHub.

<https://github.com/alvinberthelot/slides-node-js>

Cela signifie que vous n'avez pas besoin de vous battre avec un fichier binaire (le PDF) pour **contribuer**, **vous réapproprier le contenu** ou **modifier le thème** de présentation.



Contribution

Vous voulez **contribuer au contenu** car :

- Il y a une erreur (ça arrive à tout le monde), de typographie, de compréhension, ou tout autre chose.
- Vous souhaitez apporter une précision.

Il vous suffit de [contribuer au projet via Git](#) par le moyen d'une "pull request" sur le [dépôt Git](#).



Réappropriation



N'oubliez pas les conditions de la licence.

Vous voulez vous **réapproprier le contenu** car :

- Vous souhaitez donner un style différent.
- Vous souhaitez enlever/ajouter/modifier des sections dans votre contexte.

Il vous suffit de "forker" le [dépôt Git](#) et d'y apporter vos propres modifications, puis de générer par vous même le nouveau PDF.

