Mocha Testing

http://visionmedia.github.io/mocha/

Installation

- install the mocha & eslint-plugin-mocha plugins via npm

```
$ npm install --save-dev mocha
$ npm install --save-dev eslint-plugin-mocha
```

- create test folder and put test case there
- to run test, just issue command mocha

```
$ mocha
```

Edit the .eslintrc.json file to include the eslint-plugin-mocha plugins

```
{
  ...
  "plugins": [
    "mocha"
  ],
  "rules": {
    ...
    "mocha/no-exclusive-tests": "error",
    "prefer-arrow-callback": 0,
    "mocha/prefer-arrow-callback": 2
  }
}
```

Using .only for exclusive test only

Mocha has a feature that allows you to run tests exclusively by appending .only to a test-suite or a test-case. This feature is really helpful to debug a failing test, so you don’t have to execute all of your tests. After you have fixed your test and before committing the changes you have to remove .only to ensure all tests are executed on your build system.

This rule looks for every describe.only, it.only, suite.only, test.only, context.only and specify.onlyoccurrences within the source code.

e.g.

```
describe('.saveOne', function () {
  it.only('will run the test', async function () {
    ...
```

```
  })
  it.only('will not run this test', async function () {
    ...
  })
})
```

prefer arrow function callbacks (mocha-aware)

In ES6, we use arrow function a lot. But in Mocha, both describe and it callback should NOT use arrow function as the this will no longer refer to suite/test.

e.g. INCORRECT

```
describe('User', () => {
  this.timeout(5000)   // this is not refer to suite
  ...
})
```

CORRECT

```
describe('User', function () {
  this.timeout(5000)   // this is ok
  ...
})
```

Sinon JS

Sinon JS

Test Case

- Reference

http://www.wekeroad.com/2012/02/25/testing-your-model-with-mocha-mongo-and-nodejs/

From:
https://wiki.questwork.com/ - **Questwork's Wiki**

Permanent link:
https://wiki.questwork.com/doku.php?id=development:testing:mocha:start&rev=1547354073

Last update: **2019/01/13 04:34**