

Chai: <https://www.chaijs.com/> 断言库

mocha: <https://mochajs.org/> | <https://mochajs.cn/> 功能丰富的javascript测试框架

nightmare : <https://github.com/segmentio/nightmare> 浏览器自动化测试库，提供较多简便的方法**模仿用户行为**

1、基本环境

1) 必选安装

```
"chai": "^4.2.0",  
"mocha": "^8.0.1",  
"mochawesome": "^6.1.1",  
"mochawesome-report-generator": "^5.1.0"
```

2) 修改package.json , 添加script:

```
"scripts": {  
  "test:single": "mocha test/esmodule.test.js --config .mocharc-unit-testing.jsonc --require babel-core/register",  
  "test:all": "mocha --recursive test/*.js --config .mocharc-unit-testing.jsonc --require babel-core/register",  
  "b-test": "mocha --recursive ./browser/necLogin.js --show=true --DEBUG=nightmare*",  
  "b-test-debug": "mocha --recursive ./browser/necLogin.js --show=true --DEBUG=nightmare* --inspect-brk=8001",  
  "i-test": "mocha --recursive interface/getUsers.js --exit",  
  "i-test-debug": "mocha --recursive interface/getUsers.js --inspect-brk=8001 --exit"  
},
```

./browser/necLogin.js: 运行命令时执行的测试脚本文件

--inspect-brk=8001: debug端口

--show: 是否显示浏览器窗体

--DEBUG : debug info显示级别

.mocharc-unit-testing.jsonc : report生成相关配置文件

2、浏览器功能测试

1) 命令

\$npm run b-test 非debug模式

\$npm run b-test-debug debug模式

```

PS D:\Codes\auto-test> npm run b-test

> Unittest@1.0.0 b-test D:\Codes\auto-test
> mocha --recursive ./browser/necLogin.js --show=true --DEBUG=nightmare*

nec uat admin test
  ✓ 1 login test (91ms)

1 passing (97ms)
登录成功: 1
PS D:\Codes\auto-test>

```

2) 文件目录

auto-test\browser**.js

3、接口测试

1)命令

\$npm run i-test/i-test-debug

```

PS D:\Codes\auto-test> npm run i-test

> Unittest@1.0.0 i-test D:\Codes\auto-test
> mocha --recursive interface/getUsers.js --exit

1 get data from jsonplaceholder
  1) 1.1、 The number of users should be 10 .
  ✓ 1.2、 The first user name should be Leanne Graham.

1 passing (2s)
1 failing

1) 1 get data from jsonplaceholder
   1.1、 The number of users should be 10 .:

  AssertionError: expected 10 to be above 25
    + expected - actual
    -10
    +25

```

2) 目录

auto-test\interface**.js

后续可以将request和response封装为独立的模块，做接口测试只需要维护相关JSON数据即可。

4、单元测试

1) 命令：

\$npm run test:single 运行单个测试文件

\$npm run test:all 运行test目录下的所有测试文件

2) 文件

单元测试源文件: auto-test\src\js**.js

测试文件: auto-test\test**.test.js

注: 文件命名最好一一对应, 方便查找维护。

```
PS D:\Codes\auto-test> npm run test:all
> Unittest@1.0.0 test:all D:\Codes\auto-test
> mocha --recursive test/**/*.js --config .mocharc-unit-testing.jsonc --require babel-core/register

Async Test
  getApi Test
    ✓ valid Api return message (1985ms)

calculate.js test
  Sum test
    ✓ should return 3 when inputs are 1 and 2
    ✓ should return error when input is not number
    ✓ should return 0 when there's no input
  Multi test
    ✓ should return 2 when inputs are 1 and 2

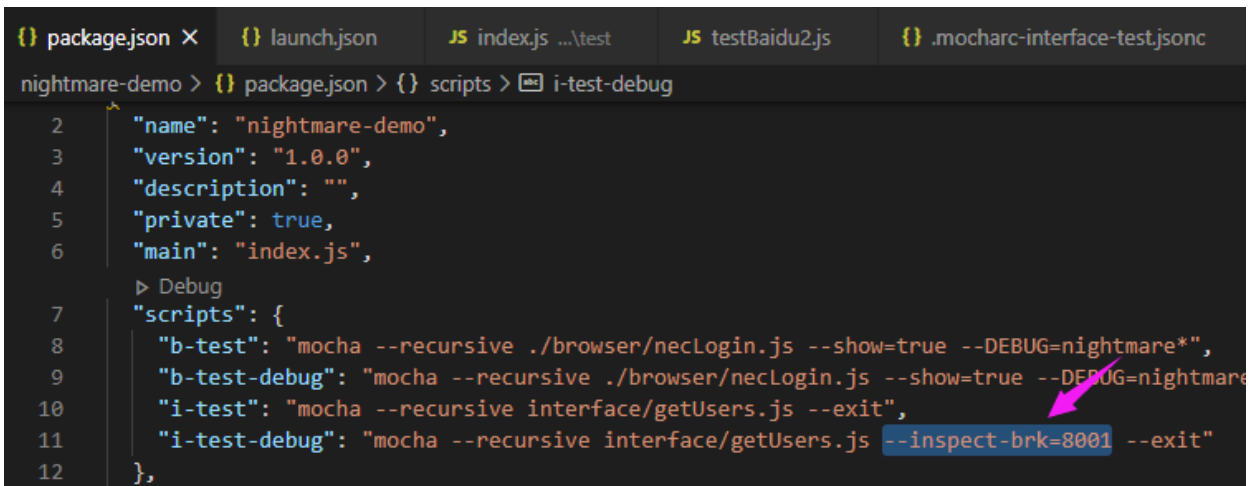
ES Module Test
  test Test
    ✓ valid es module work

6 passing (2s)

[mochaawesome] Report JSON saved to D:\Codes\auto-test\report\Unit-testing\unittest.json
[mochaawesome] Report HTML saved to D:\Codes\auto-test\report\Unit-testing\unittest.html
```

5. debug

1) package.json 中 script 配置 --inspect-brk参数



```
{
  "name": "nightmare-demo",
  "version": "1.0.0",
  "description": "",
  "private": true,
  "main": "index.js",
  "scripts": {
    "b-test": "mocha --recursive ./browser/necLogin.js --show=true --DEBUG=nightmare*",
    "b-test-debug": "mocha --recursive ./browser/necLogin.js --show=true --DEBUG=nightmare",
    "i-test": "mocha --recursive interface/getUsers.js --exit",
    "i-test-debug": "mocha --recursive interface/getUsers.js --inspect-brk=8001 --exit"
  }
}
```

2) Launch.json中添加相关attach信息, 例如

```
{
  "type": "node",
  "name": "nightmare demo Attach",
  "port": 8001, //注: 这里的端口号要和script中的相对于
  "request": "attach",
  "skipFiles": [
    "<node_internals>/**"
  ]
}
```

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "node",
      "name": "nightmare demo Attach",
      "port": 8001,
      "request": "attach",
      "skipFiles": [
        "<node_internals>/**"
      ]
    },
    {
      "type": "node",
      "request": "launch",
      "name": "Launch Program",
      "skipFiles": [
        "<node_internals>/**"
      ],
      "program": "${file}"
    }
  ]
}
```

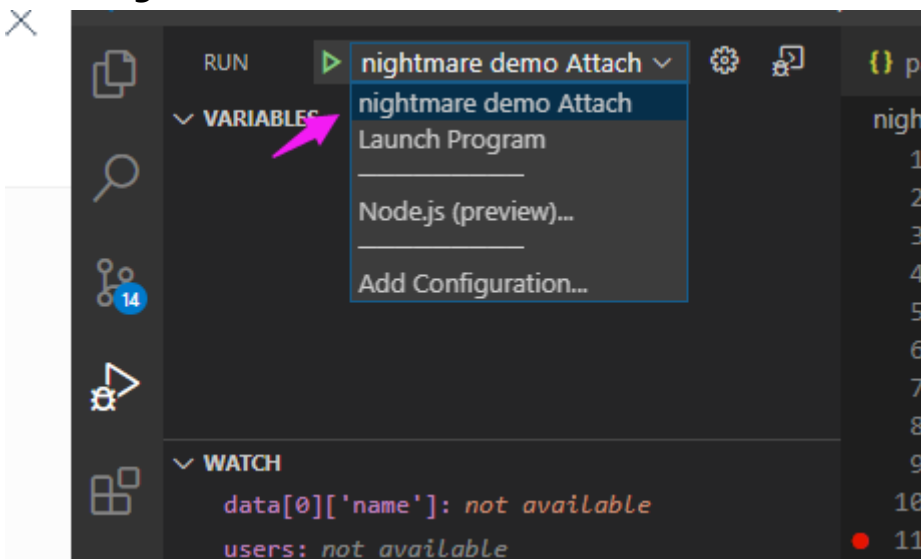
3) 运行 `npm run i-test-debug` 出现以下提示信息，运行vscode

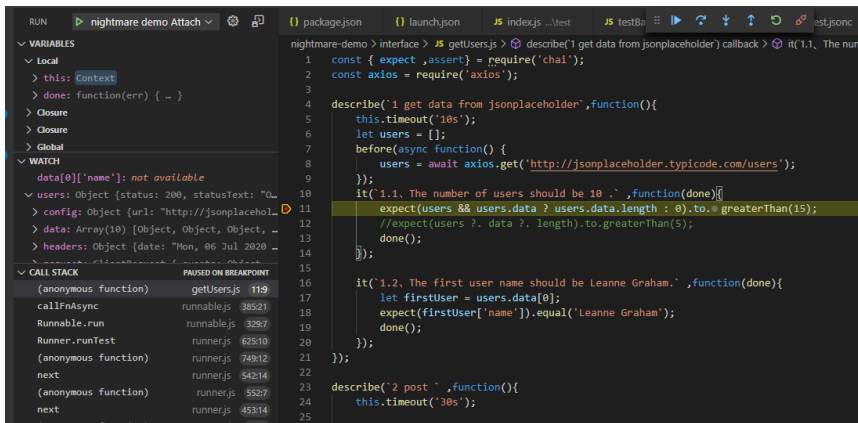
```
PS D:\Codes\temp\nightmare-demo> npm run b-test-debug

> nightmare-demo@1.0.0 b-test-debug D:\Codes\temp\nightmare-demo
> mocha --recursive ./browser/necLogin.js --show=true --DEBUG=nightmare* --inspect-brk=8001

Debugger listening on ws://127.0.0.1:8001/a92c5513-e3af-441d-b891-8cf3a0b13a0a
For help, see: https://nodejs.org/en/docs/inspector
```

4)运行nightmare demo attach





6、nightmare存在的局限性

无法做浏览器兼容性测试

相关文档较少