

单元测试工具

行为驱动开发（BDD）

获取：

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

建立一个test.js文件

```
var assert = require('assert');describe('Array', function() {  
  describe('#indexOf()', function() {  
    it('should return -1 when the value is not present', function() {  
      assert.equal(-1, [1,2,3].indexOf(4));  
    });  
  });  
});
```

describe> it

describe块称为“测试套件”（test suite），表示一组相关的测试。它是一个函数，第一个参数是测试套件的名称（“加法函数的测试”），第二个参数是一个实际执行的函数。

it块称为“测试用例”（test case），表示一个单独的测试，是测试的最小单位。它也是一个函数，第一个参数是测试用例的名称（“1 加 1 应该等于 2”），第二个参数是一个实际执行的函数。

测试：

```
$ mocha test.js
```

断言：

```
// 相等或不相等  
expect(4 + 5).to.be.equal(9);expect(4 +  
5).to.be.not.equal(10);expect(foo).to.be.deep.equal({ bar: 'baz' });
```

```

// 布尔值为true
expect('everthing').to.be.ok;expect(false).to.not.be.ok;

// typeof
expect('test').to.be.a('string');expect({ foo: 'bar'
}).to.be.an('object');expect(foo).to.be.an.instanceof(Foo);

// include
expect([1,2,3]).to.include(2);expect('foobar').to.contain('foo');expect({ foo:
'bar', hello: 'universe' }).to.include.keys('foo');

// empty
expect([]).to.be.empty;expect('').to.be.empty;expect({}).to.be.empty;

// match
expect('foobar').to.match(/^foo/);

```

目录：Mocha默认运行test子目录里面的测试脚本。所以，一般都会把测试脚本放在test目录里面，然后执行mocha就不需要参数了。

命令行参数：

```

# 帮助
$ mocha --help

#--reporter参数用来指定测试报告的格式，默认是spec格式。
$ mocha
# 等同于
$ mocha --reporter spec

# --reporters参数可以显示所有内置的报告格式。
$ mocha --reporters

# 使用mochawesome模块，可以生成漂亮的HTML格式的报告。
$ mocha --reporter mochawesome

# 打开--growl参数，就会将测试结果在桌面显示。
$ mocha --growl

```

```
# --watch参数用来监视指定的测试脚本。只要测试脚本有变化，就会自动运行Mocha。
$ mocha --watch

#--bail参数指定只要有一个测试用例没有通过，就停止执行后面的测试用例。这对持续集成很有用。
$ mocha --bail

# --grep参数用于搜索测试用例的名称（即it块的第一个参数），然后只执行匹配的测试用例。
$ mocha --grep "1 加 1"

# --invert参数表示只运行不符合条件的测试脚本，必须与--grep参数配合使用。
$ mocha --grep "1 加 1" --invert
```

配置文件mocha.opts： Mocha允许在test目录下面，放置配置文件mocha.opts，把命令行参数写在里面。默认运行mocha时运用配置好的参数

异步测试： Mocha默认每个测试用例最多执行2000毫秒，如果到时没有得到结果，就报错。对于涉及异步操作的测试用例，这个时间往往是不够的，需要用-t或--timeout参数指定超时门槛。

```
$ mocha -t 5000 timeout.test.js
```

测试用例的钩子： Mocha在describe块之中，提供测试用例的四个钩子：before()、after()、beforeEach()和afterEach()。它们会在指定时间执行。

```
describe('hooks', function() {

  before(function() {
    // 在本区块的所有测试用例之前执行
  });

  after(function() {
    // 在本区块的所有测试用例之后执行
  });

  beforeEach(function() {
    // 在本区块的每个测试用例之前执行
  });
```

```
afterEach(function() {  
  // 在本区块的每个测试用例之后执行  
});  
  
// test cases  
});
```

浏览器测试： 除了在命令行运行，Mocha还可以在浏览器运行。

使用mocha init命令在指定目录生成初始化文件

```
$mocha init mocha-test
```

[mochawesome](#)