

- [Basic]
- "
- [Basic]
- [Basic]
- [Basic]
- [Basic]
- [Doc] Assert ()

" ? ?

,



, ABCD , EFGH (), A BUG H , A BC,
, (80% , 90%) , , (,
 , A BUG " , .H B?" , BC
, :

- ,
- ,

“ ? ”

, , , .

(Black-box Testing),  , . , (Integration Testing) , (System

(White-box Testing) , (). , (Unit

(Unit Testing) . , . ; (Unit) , , .

, , / / .

(Test Coverage) , 90% , 90% .

:

- (line coverage)
- (function coverage)
- (branch coverage) if
- (statement coverage)

[istanbul](#).

? , .

Mock

Mock . (/) . , .

(/ /), mock . , Mock behav

Mock Stub [Mocks Aren't Stubs](#)

- [Mocha](#)
- [ava](#)
- [Jest](#)

. , , . , ,

web app , :

```

describe('Path API', () => {
  // ...

  describe('GET /v2/path/:_id', () => {
    it('should return 200 GET /v2/path/:_id', () => {
      return request
        .get('/v2/path/' + pathId)
        .set('Cookie', 'common_user=xxx')
        .expect(200);
    });
  });

  describe('POST /v2/path', () => {
    it('should return 412 POST /v2/path lost params path', () => {
      return request
        .post('/v2/path')
        .set('Cookie', 'common_user=xxx')
        .expect(412);
    });

    it('should return 409 POST /v2/path when path exist', () => {
      return request
        .post('/v2/path')
        .send({path: '/'})
        .set('Cookie', 'common_user=xxx')
        .expect(409);
    });

    it('should return 200 POST /v2/path successfully', () => {
      return request
        .post('/v2/path')
        .send({path: '/comment'})
        .set('Cookie', 'common_user=xxx')
        .expect(200);
    });
  });

  // ...

```

```
});
```

Node.js [benchmark](#).

```
const Benchmark = require('benchmark');
const suite = new Benchmark.Suite;

suite.add('RegExp#test', function() {
  /o/.test('Hello World!');
})
.add('String#indexOf', function() {
  'Hello World!'.indexOf('o') > -1;
})
.on('cycle', function(event) {
  console.log(String(event.target));
})
.on('complete', function() {
  console.log('Fastest is ' + this.filter('fastest').map('name'));
})
// run async
.run({ 'async': true });
```

, .

[Apache ab wrk](#) , :

```
ab -n 100 -c 10 https://ele.me/
```

:

Server Software:	Tengine/2.1.1
Server Hostname:	ele.me
Server Port:	443
SSL/TLS Protocol:	TLSv1.2, ECDHE-RSA-AES256-GCM-SHA384, 2048, 256
Document Path:	/
Document Length:	284 bytes
Concurrency Level:	10
Time taken for tests:	1.775 seconds

```
Complete requests:      100
Failed requests:        0
Non-2xx responses:      100
Total transferred:      62400 bytes
HTML transferred:       28400 bytes
Requests per second:    56.33 [#/sec] (mean)
Time per request:       177.511 [ms] (mean)
Time per request:       17.751 [ms] (mean, across all concurrent requests)
Transfer rate:          34.33 [Kbytes/sec] received
```

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	88	116 26.0	104	234
Processing:	33	55 39.6	47	394
Waiting:	33	54 39.0	46	394
Total:	124	171 48.1	152	491

Percentage of the requests served within a certain time (ms)

50%	152
66%	184
75%	193
80%	199
90%	224
95%	242
98%	288
99%	491
100%	491 (longest request)

, ab . / , ab wrk .

(Stress testing), [Jmeter](#) QPS, TPS , .

, , . , , IO IO .

Assert

(Assert) . :

```
if (condition) {
```

```
    throw new Error(' Sth wrong' );  
  }
```

:

```
assert(! condition, ' Sth wrong' );
```

```
equal . , / .
```

Node.js `assert` **not intended to be used as a general purpose assertion library**)

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

Revision #1

Created 19 July 2021 15:23:02 by

Updated 19 July 2021 15:28:28 by