OPEN ORIENTED
凹 凸 实 验 室

浅谈前端单元测试

luckyadam

1 单元测试

2 工具

3 拓展

1

单元测试

单元测试

02

“对程序中最小可测试单元进行检查和验证”

单元测试

最小可测试单元

单元测试

单元测试

【保证写出来的代码真的正确】

【强制思考每一个单元会遇到的各种情况】

【提升代码设计能力】

【毫无顾虑地优化代码】

2

工具

工具

工欲善其事必先利其器

工具

Mocha

BDD Style

```
const assert = require('assert')

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

工具

BDD Style

describe 测试套件，可以嵌套

it 测试用例

Hooks

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

  before(function() {
    // runs before all tests in this block
  });

  after(function() {
    // runs after all tests in this block
  });

  beforeEach(function() {
    // runs before each test in this block
  });

  afterEach(function() {
    // runs after each test in this block
  });

  // test cases
});
```

工具

web测试

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mocha</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://cdn.bootcss.com/mocha/3.5.0/mocha.min.css" rel="stylesheet">
  </head>
  <body>
    <div id="mocha"></div>
    <script src="https://cdn.bootcss.com/mocha/3.5.0/mocha.min.js"></script>
    <script src="https://cdn.bootcss.com/chai/4.1.1/chai.min.js"></script>
    <script>mocha.setup('bdd');</script>
    <script src="index.js"></script>
    <script src="tests.js"></script>
    <script>
      mocha.run();
    </script>
  </body>
</html>
```

工具

断言库

chai.js

工具 chai

TDD/BDD

Should

```
chai.should();

foo.should.be.a('string');
foo.should.equal('bar');
foo.should.have.lengthOf(3);
tea.should.have.property('flavors')
  .with.lengthOf(3);
```

Visit Should Guide

Expect

```
var expect = chai.expect;

expect(foo).to.be.a('string');
expect(foo).to.equal('bar');
expect(foo).to.have.lengthOf(3);
expect(tea).to.have.property('flavors')
  .with.lengthOf(3);
```

Visit Expect Guide

Assert

```
var assert = chai.assert;

assert.typeOf(foo, 'string');
assert.equal(foo, 'bar');
assert.lengthOf(foo, 3)
assert.property(tea, 'flavors');
assert.lengthOf(tea.flavors, 3);
```

Visit Assert Guide

工具

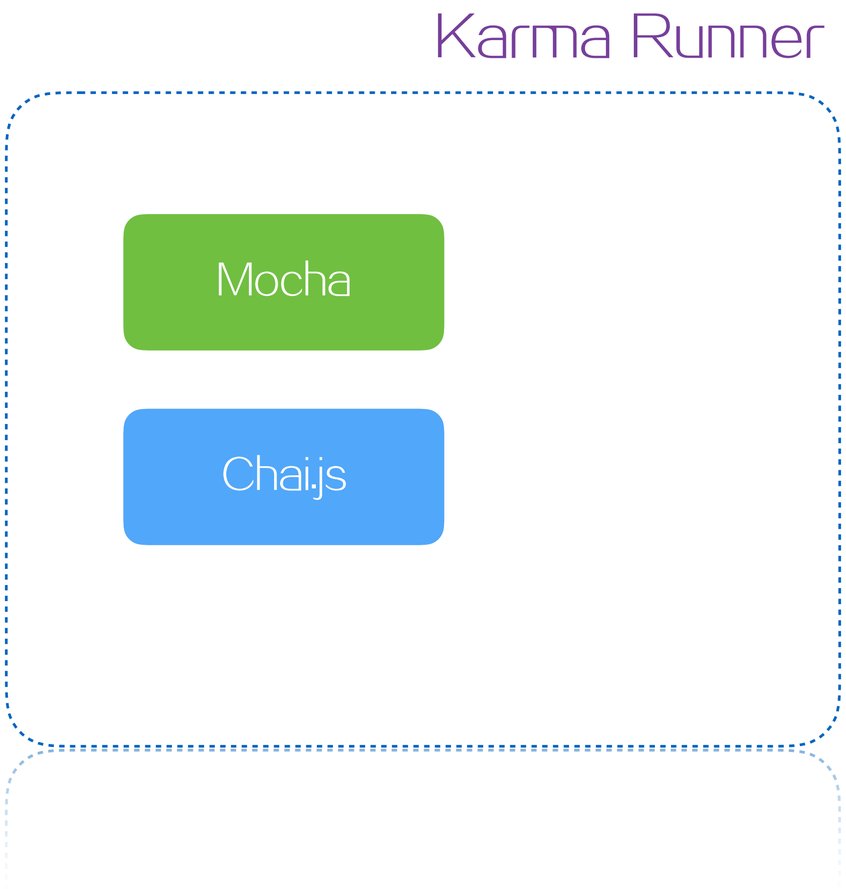
02

【自动化跑起来！】

工具

Runner

工具 KARMA



工具 KARMA

开启web运行环境【本地浏览器、phantom】

生成web测试页面，加载测试框架以及测试脚本

支持watch源码、测试用例改变，重新测试

工具 KARMA


```
$ npm install -g karma-cli
```

```
$ npm install karma --save-dev
```

```
$ karma init
```

工具 KARMA


```
➜ test-learn git:(master) ✗ karma init

Which testing framework do you want to use ?
Press tab to list possible options. Enter to move to the next question.
> mocha

Do you want to use Require.js ?
This will add Require.js plugin.
Press tab to list possible options. Enter to move to the next question.
> no

Do you want to capture any browsers automatically ?
Press tab to list possible options. Enter empty string to move to the next question.
> Chrome
>

What is the location of your source and test files ?
You can use glob patterns, eg. "js/*.js" or "test/**/*Spec.js".
Enter empty string to move to the next question.
>

Should any of the files included by the previous patterns be excluded ?
You can use glob patterns, eg. "**/*.swp".
Enter empty string to move to the next question.
>

Do you want Karma to watch all the files and run the tests on change ?
Press tab to list possible options.
> yes


Config file generated at "/Users/luckyadam/Project/test-learn/karma.conf.js".
```

工具 KARMA


```
"devDependencies": {
  "chai": "^4.1.1",
  "karma": "^1.7.0",
  "karma-chrome-launcher": "^2.2.0",
  "karma-mocha": "^1.3.0",
  "karma-sinon-chai": "^1.3.1",
  "mocha": "^3.5.0",
  "sinon": "^3.2.0",
  "sinon-chai": "^2.13.0"
}
```

工具 KARMA

karma.conf.js

```
// frameworks to use
// available frameworks: https://npmjs.org/browse/keyword/karma-adapter
frameworks: ['mocha', 'sinon-chai'],
```

配置所需的测试框架

工具 KARMA

karma.conf.js

```
// start these browsers
// available browser launchers: https://npmjs.org/browse/keyword/karma-launcher
browsers: ['Chrome'],
```

测试启动web环境

工具 KARMA

karma.conf.js

```
// test results reporter to use
// possible values: 'dots', 'progress'
// available reporters: https://npmjs.org/browse/keyword/karma-reporter
reporters: ['mocha'],


mochaReporter: {
  showDiff: true
},
```

测试报告展示

工具 KARMA

【ES2015+Babel?】

webpack

工具 KARMA

karma.conf.js

```
// preprocess matching files before serving them to the browser
// available preprocessors: https://npmjs.org/browse/keyword/karma-preprocessor
preprocessors: {
  'test/**/*.js': ['webpack', 'sourcemap']
},
```

预处理测试用例文件

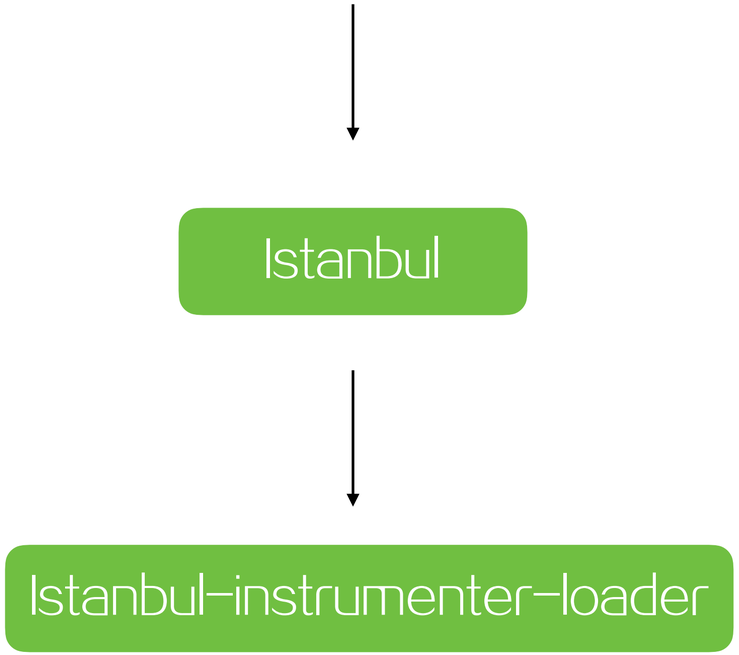
工具 KARMA

karma.conf.js

```
webpack: {
  devtool: 'inline-source-map',
  module: {
    rules: [
      {
        enforce: 'pre',
        test: /\.js$/,
        loader: 'babel-loader',
        exclude: /node_modules/
      }
    ]
  },
  plugins: [
    new webpack.DefinePlugin({
      coverage: coverage,
      NODE_ENV: JSON.stringify(process.env.NODE_ENV || ''),
      DISABLE_FLAKEY: !!String(process.env.FLAKEY).match(/^(0|false)$/gi)
    })
  ]
},
```

工具 KARMA

【代码测试覆盖率】



工具 KARMA

karma.conf.js

```
webpack: {
  devtool: 'inline-source-map',
  module: {
    rules: [
      {
        enforce: 'pre',
        test: /\.js$/,
        loader: 'babel-loader',
        exclude: /node_modules/
      },
      {
        enforce: 'post',
        test: /\.js$/,
        use: {
          loader: 'istanbul-instrumenter-loader',
          options: { esModules: true }
        },
        include: path.resolve('src/'),
        exclude: /node_modules/
      }
    ]
  },
```

3

拓展

拓展

持续集成

拓展

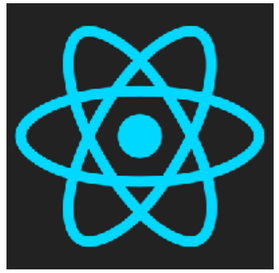
02

多终端测试

拓展

02

组件化测试



02

OPEN ORIENTED

凹 凸 实 验 室