

JavaScript tesztelési módszerek

BRAUN PATRIK

JavaScript

Objektumalapú, prototípus alapú szkriptnyelv

C szerű szintaktikával

szabványosították, mégis részben különbözően implementálják a JavaScriptet a különböző böngészők.

Elsősorban webre, de szerver oldalon is futtatható

- Node.js-el
- Paypal js-ben íródott (szerver oldal is)

Hivatalosan típusatlan nyelv

Nem mindig egyértelmű : Truthy, falsey, this

Nagyobb projektek esetén : Typescript, Babel



JavaScript tesztelés: Keretrendszerek

Buster.JS

Capbara

Mocha

FuncUnit

Hiro

Karma (née Testacular)

Jasmine

JsTestDriver

Laika

- Preamble
- phantom-assert
- **QUnit**
- Robot Framework
- Rx Test Runner
- tapedeck
- Testem
- Venus.js
- WebDriver
- wru
- YUITest

Kliens oldali tesztelés: Karma + Jasmine

Karma

- Teszt futtató környezet
- Valós böngészőkön vagy PhantomJs-ben
- Akár több eszközön is
- Forrás figyelés, autómaitus futtatás
- Teszt lefedettség mérés, Istanbul-t is támogat

Jasmine

- Behavior Driven Development tesztelő keretrendszer
- describe block format



Kliens oldali tesztelés: Karma + Jasmine

```
module FrontEndModule {
  export class Utis{
    public static
      calc(a:number) {
        if( a < 0){
          return -a;
        }

        if( a < 10){
          return a * a;
        }
        return a;
      }
  }
}
```

```
describe("A simple test for testing Utils.", function() {
  var foo = 0;

  beforeEach(function() {
    foo += 1;
  });

  afterEach(function() {
    foo = 0;
  });

  it("should get absolute value for input less than 0", function() {
    expect(FrontEndModule.Utis.calc(-foo)).toEqual(1);
  });

  it("should get square value for input between 0 and 10", function() {
    expect(FrontEndModule.Utis.calc(foo+2)).toEqual(9);
  });
});
```

Kliens oldali tesztelés: Karma + Jasmine

```
// karma.conf.js
module.exports = function(config) {
  config.set({
    basePath: '../',
    files: ["test/jasmine.js", "js/Utils.js"],
    frameworks: ['jasmine'],
    browsers: ['Chrome'],
    autoWatch: true,

    reporters: ['progress', 'coverage'],
    preprocessors: { 'js/Utils.js': ['coverage'] }
  });
};
```

```
"C:\work\IntelliJ IDEA 15.0.1\bin\runnerw.exe" C:\work\nodejs\node.exe d:\Work\Sandbox\node_modules\karma\bin\karma start karma.conf.js
02 12 2015 22:18:40.960:WARN [karma]: No captured browser, open http://localhost:9876/
02 12 2015 22:18:40.969:INFO [karma]: Karma v0.13.15 server started at http://localhost:9876/
02 12 2015 22:18:40.976:INFO [launcher]: Starting browser Chrome
02 12 2015 22:18:42.421:INFO [Chrome 46.0.2490 (Windows 10 0.0.0)]: Connected on socket CLfbuwoYyZw_c7z7AAAA with id 96243737
Chrome 46.0.2490 (Windows 10 0.0.0): Executed 0 of 2 SUCCESS (0 secs / 0 secs)
Chrome 46.0.2490 (Windows 10 0.0.0): Executed 1 of 2 SUCCESS (0 secs / 0.002 secs)
Chrome 46.0.2490 (Windows 10 0.0.0): Executed 2 of 2 SUCCESS (0 secs / 0.003 secs)
Chrome 46.0.2490 (Windows 10 0.0.0): Executed 2 of 2 SUCCESS (0.008 secs / 0.003 secs)
```

Test Coverage: Istanbul

all files / js/ Utils.js

91.67% Statements 11/12 83.33% Branches 5/6 75% Functions 3/4 91.67% Lines 11/12

```
1 1x var FrontEndModule;
2 1x (function (FrontEndModule) {
3 1x   var Utis = (function () {
4 1x     function Utis() {
5       }
6 1x     Utis.calc = function (a) {
7 2x       if (a < 0) {
8 1x         return -a;
9       }
10 1x     E if (a < 10) {
11 1x       return a * a;
12     }
13     return a;
14   };
15 1x   return Utis;
16 }());
17 1x FrontEndModule.Utis = Utis;
18 })(FrontEndModule || (FrontEndModule = {}));
19 //# sourceMappingURL=Utils.js.map
```

Szerver oldali tesztelés: Mocha + Chai

Mocha

- Hasonló mint a JasmineJs
- Inkább szerver oldali kód tesztelésre használják
- Nincs beépített assertion

Chai

- BDD/TDD assertion könyvtár
- Támogat: should, expect, assert formátumokat

```
foo.should.be.a('string');  
expect(foo).to.be.a('string');  
assert.typeOf(foo, 'string');
```


Szerver oldali tesztelés: Mocha + Chai

```
module Sandbox{
  export class MyCode{
    static foo(a:number) {
      if( a < 0){
        throw new Error("sqrt
can't work on negative number");
      }
      return 2*a;
    }
  }
}

if(typeof module !== 'undefined' &&
typeof module.exports !== 'undefined'){
  module.exports = Sandbox.MyCode;
}
```

```
var expect = require('chai').expect;
var MyCode = require('../js/MyCode');

describe('foo', function () {
  it('should return 4', function (){
    var c = MyCode.foo(2);
    expect(c).to.be.equal(4);
  });

  it('examples', function(){
    expect({ foo: 'bar' }).to.be.an('object');
    expect(null).to.be.a('null');
    expect(undefined).to.be.an('undefined');

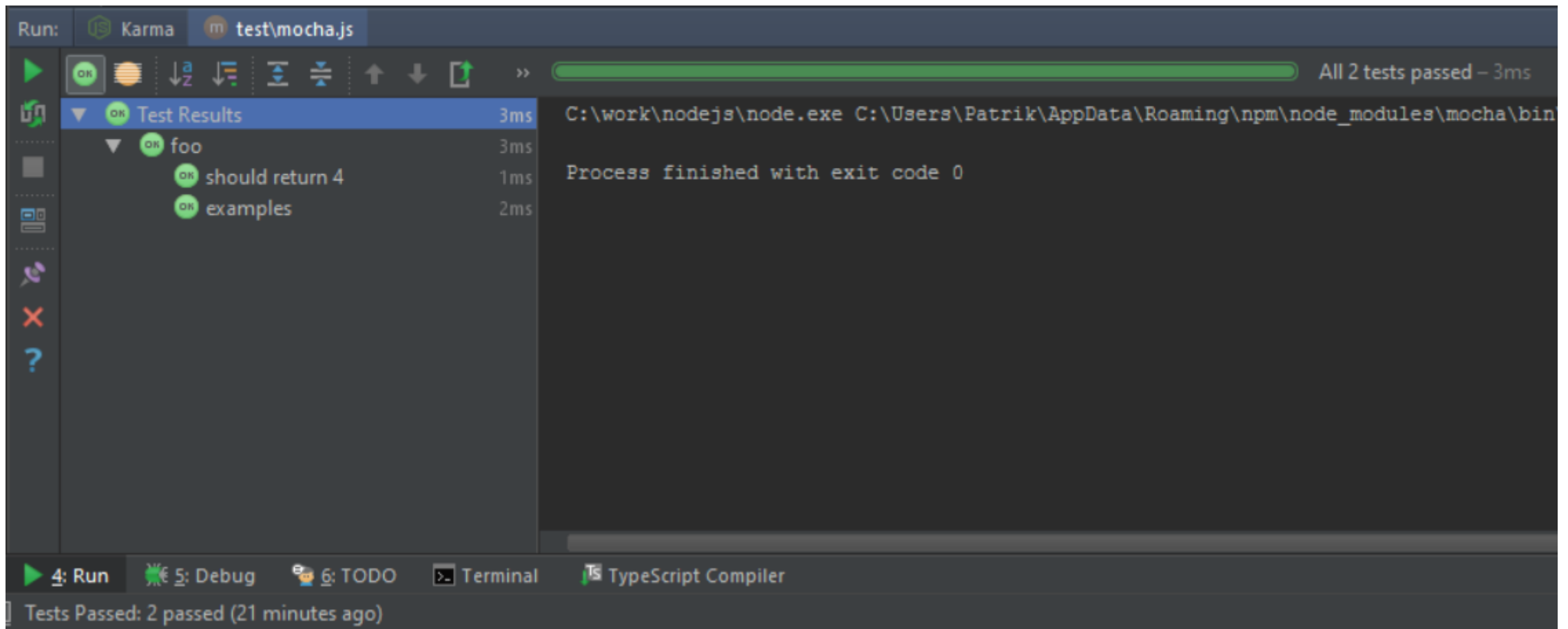
    var obj = { foo: 'bar' };
    expect(obj).to.have.property('foo');
    expect(obj).to.have.property('foo', 'bar');

    var err = new ReferenceError('This is a bad function.');
```

```
var fn = function () { throw err; };
expect(fn).to.throw(ReferenceError);
expect(fn).to.throw(Error);

  });
});
```

Szerver oldali tesztelés: Mocha + Chai



Run: Karma test\mocha.js

All 2 tests passed – 3ms

Test Results 3ms

- foo 3ms
 - should return 4 1ms
 - examples 2ms

```
C:\work\nodejs\node.exe C:\Users\Patrik\AppData\Roaming\npm\node_modules\mocha\bin
Process finished with exit code 0
```

4: Run 5: Debug 6: TODO Terminal TypeScript Compiler

Tests Passed: 2 passed (21 minutes ago)

End to End tesztek :Selenium+ Cucumber

- TDD tesztek
- CucumberJs a Cucumber portja
- define Feature Specs in a Domain-Specific-Language (DSL), Gherkin

```
Feature: Shopper can add an item to their Grocery List  
As a grocery shopper  
I want to add an item to my grocery list  
So that I can remember to buy that item at the grocery store
```

```
Scenario: Item added to grocery list  
  Given I have an empty grocery list  
  When I add an item to the list  
  Then The grocery list contains a single item
```

```
Scenario: Item accessible from grocery list  
  Given I have an empty grocery list  
  When I add an item to the list  
  Then I can access that item from the grocery list
```

End to End tesztek :Selenium+ Cucumber

Feature: Gmail Login

As a user I should able to login into Gmail.

Scenario: I login with valid credentials

Given I open "http://www.gmail.com"

And I enter username "abc@gmail.com"

Then I click next button

And I enter password "123456"

When I click signin button

And I wait for 15 sec

Then this is my another custom step

End to End tesztek: FuncUnit

Egy újabb keretrendszer tesztelése

- Támogat Jasmine és Qunit szintaxist
- Részletek és teszt példa itt: <http://funcunit.com/>

```
//slowing funcunit down for dramatic effect
F.speed = 100;

describe('TodoMVC', function() {
  it('should create and complete todos', function() {
    var newTodo = F('#new-todo');
    newTodo.type('FuncUnit [enter]');
    newTodo.type('is [enter]');
    newTodo.type('awesome! [enter]');

    F('.todo label:contains("FuncUnit")').visible();
    F('.todo label:contains("is")').visible();
    F('.todo label:contains("awesome")').visible();

    F('.toggle:not(:checked)').click();
    F('.toggle:not(:checked)').click();
    F('.toggle:not(:checked)').click();

    F('#clear-completed').click();
    F('.todo.completed').missing();
  });
});
```

Tesztelés: Mocking Sinon-al

Javascript könnyen mockolható

- minden függvény és paraméter a kód futása során bármikor felülírható

Sinon:

- Támogat: Spy-t és stub-ot

Példa:

```
function getTodos(listId, callback) {
  jQuery.ajax({
    url: "/todo/" + listId + "/items",
    success: function (data) {
      // Node-style CPS: callback(err, data)
      callback(null, data);
    }
  });
}
```

Tesztelés: Mocking Sinon-al

```
after(function () {
  jQuery.ajax.restore();
});

it("makes a GET request for todo items", function () {
  sinon.stub(jQuery, "ajax");
  getTodos(42, sinon.spy());

  assert(jQuery.ajax.calledWithMatch({ url: "/todo/42/items" }));
});
```

```
var server;

before(function () { server = sinon.fakeServer.create(); });
after(function () { server.restore(); });

it("calls callback with deserialized data", function () {
  var callback = sinon.spy();
  getTodos(42, callback);

  // This is part of the FakeXMLHttpRequest API
  server.requests[0].respond(
    200,
    { "Content-Type": "application/json" },
    JSON.stringify([{ id: 1, text: "Provide examples", done: true }])
  );

  assert(callback.calledOnce);
});
```

Headless futtatás: PhantomJS

Teljes értékű böngészőt helyettesít

- Js-ben írható futási script

Sandbox-ban futtatja az oldalakat

Lehetőség screen capture

Sztelt oldalak DOM-jához házzáfér és manipulálható

Minotorozási és Profiling lehetőségek



Köszönöm a figyelmet!

