core networks

DESARROLLO WEB CON MEAN

Curso: DESARROLLO WEB CON MEAN (WEB FULL STACK DEVELOPER)

Testing y depuración de aplicaciones MEAN

Introducción a testing y depuración de aplicaciones MEAN

Las ventajas de MEAN como aproximación para desarrollar aplicaciones web usando un solo lenguaje (javascript) como un lenguaje uniforme son claras.

Para realizar las pruebas y comprobar que estamos desarrollando software de calidad, podemos usar una serie de frameworks que ayudan a probar las aplicaciones.

Según el tipo de pruebas que se realicen y a que estén orientadas, podemos usar una serie de frameworks que nos vienen predefinidos, aunque podemos cambiarlos

Testing y depuración de aplicaciones MEAN

Introducción a testing y depuración de aplicaciones MEAN

EL PRIMERO DE ELLOS ES GRUNT

PERMITE LA AUTOMATIZACIÓN DE TESTS REALIZADOS PARA NUESTRA APLICACIÓN

ES UNA FORMA DE ORGANIZAR LOS DISTINTOS SISTEMAS DE PRUEBAS SOPORTADOS POR MEAN

POR DEFECTO VIENE EN DOS CATEGORÍAS

LOS TEST DE MOCHA, QUE SE USAN PARA LA CAPA DE SERVIDOR

LOS TEST DE KARMA COMO SUPERFRAMEWORK, QUE PERMITE LANZAR NAVEGADORES PARA PROBAR CON DISTINTOS FRAMEWORKS DE TEST.
LOS TEST DE JASMINE, QUE VIENE PRECONFIGURADO PARA MEAN Y KARMA

Testing y depuración de aplicaciones MEAN

Introducción a testing y depuración de aplicaciones MEAN

USANDO YEOMAN GENERATOR COMO GENERADOR DE PROYECTOS MEAN, SE ENTREGAN UNA SERIE DE P

PARA INSTALAR EL GENERADOR SOLO NECESITAMOS INSTALAR DOS COMPONENTES

npm install –g yo
npm install –g generator-mean

DE ESTA FORMA CREAREMOS ESQUEMAS DE APLICACIÓN CON UNA SERIE DE PRUEBAS PREDEFINIDAS

testing y depuración de aplicaciones MEAN

PARA EJECUTAR TODAS LAS PRUEBAS DE UNA VEZ, LANZAMOS EL COMANDO

$ grunt test

unning "env:test" (env) task

Running "mochaTest:src" (mochaTest) task

Application loaded using the "test" environment configuration

Running "karma:unit" (karma) task
INFO [karma]: Karma v0.12.31 server started at http://localhost:9876/
INFO [launcher]: Starting browser PhantomJS
INFO [PhantomJS 1.9.8 (Mac OS X)]: Connected on socket 6zkU-H6qx_m2J6lY4zJ8 with id 51669923
PhantomJS 1.9.8 (Mac OS X): Executed 18 of 18 SUCCESS (0.016 secs / 0.093 secs)

Done, without errors.

Testing y depuración de aplicaciones MEAN

Introducción a testing y depuración de aplicaciones MEAN

AL USAR GRUNT TEST EJECUTA EL FICHERO gruntfile.js

EN UNO DE LOS APARTADOS SE DEFINEN LAS TAREAS DE PRUEBAS

```
// Test task.
grunt.registerTask('test', ['env:test', 'mochaTest', 'karma:unit']);
```

EN ESTA PARTE SE ESTÁN REGISTRANDO LAS TAREAS A NIVEL DE ENTORNO Y CON EL FRAMEWORK DESEADO

MÁS ARRIBA APARECE LA VARIABLE POR DEFECTO PARA DEFINIR EL ENTORNO DE LAS PRUEBAS

```
env: {
      test:  {
      NODE_ENV: 'test'
      }
},
```

Introducción a testing y depuración de aplicaciones MEAN

SI VAMOS A config/env/test.js

```
'use strict';

module.exports = {
   db: 'mongodb://localhost/test-test',
   port: 3001,
       app:  {
      title: 'Test - Test Environment'
      },
      facebook:  {
      clientID: process.env.FACEBOOK_ID || 'APP_ID',
      clientSecret: process.env.FACEBOOK_SECRET || 'APP_SECRET',
      callbackURL: 'http://localhost:3000/auth/facebook/callback'
      },
      google:  {
      clientID: process.env.GOOGLE_ID || 'APP_ID',
      clientSecret: process.env.GOOGLE_SECRET || 'APP_SECRET',
      callbackURL: 'http://localhost:3000/auth/google/callback'
            },
      //  snip
};
```

Introducción a testing y depuración de aplicaciones MEAN

AQUÍ SE ESTÁN DEFINIENDO LOS MOCKS DE AUTENTICACIÓN DE FACEBOOK Y GOOGLE PARA PASSPORT

```
mochaTest: {
   src: watchFiles.mochaTests,
       options:  {
              reporter:  'spec',
      require: 'server.js'
       }
},
```

Testing y depuración de aplicaciones MEAN

Introducción a testing y depuración de aplicaciones MEAN

LOS TEST DE MOCHA QUE ESTÁN DEFINIDOS NO SON LANZADOS POR KARMA, YA QUE NO POSEE PARTE DE CLIENTE.

LA ULTIMA PARTE DE DEFINICIÓN DEL FICHERO GRUNT INDICA LA CARGA DE KARMA PARA LOS TEST DE CLIENTE

```
karma: {
        unit:  {
       configFile: 'karma.conf.js'
        }
}
```

Testing y depuración de aplicaciones MEAN

Introducción a testing y depuración de aplicaciones MEAN

KARMA PERMITE ESCRIBIR LAS PRUEBAS EN EL FRAMEWORK QUE QUERAMOS
PERMITE EL USO DE DIVERSOS PLUGINS PARA LANZAR NAVEGADORES REALES BAJO DEMANDA
PERMITE LANZAR NAVEGADORES EN LOCAL PARA REALIZAR LAS PRUEBAS
TAMBIÉN PERMITE CAPTURAR NAVEGADORES REMOTOS CON EL KARMA SERVER
POSEE PLUGINS PARA GENERACIÓN DE INFORMES CON LOS RESULTADOS DE LAS PRUEBAS

Testing y depuración de aplicaciones MEAN

karma.conf.js

ESTÁ DIVIDIDO EN DISTINTAS SECCIONES

USO DE FRAMEWORKS

```
module.exports = function(config) {
  config.set({
    // Frameworks to use
    frameworks: ['jasmine'],
```

FICHEROS A CARGAR EN EL NAVEGADOR

```
files:
applicationConfiguration.assets.lib.js.concat(applicationConfiguration.assets.js,
    applicationConfiguration.assets.tests),
```

Testing y depuración de aplicaciones MEAN

karma.conf.js

GENERADORES DE INFORME DE RESULTADOS
reporters: ['progress'],

PUERTO DEL SERVIDOR WEB
port: 9876,

NAVEGADOR: INCLUYE CHROME, CHROME CANARY, OPERA….
browsers: ['PhantomJS'],

SI QUEREMOS QUE SE EJECUTE Y SE SALGA O LO HAGA EN BUCLE
singleRun: true

Introducción a testing y depuración de aplicaciones MEAN

PARA GENERAR LOS TEST DE LA CAPA DE ANGULAR SE USA EL GENERADOR DE YEOMAN

$ yo meanjs:angular-tests <nombre-del-controlador>

PERMITE CREAR LOS FICHEROS DE TEST PARA HACER PRUEBAS EN ANGULAR

GENERA LOS FICHEROS NECESARIOS PARA GENERAR LAS PRUEBAS

SOLO NECESITAMOS MODIFICARLOS PARA HACER NUESTRAS PRUEBAS

Testing y depuración de aplicaciones MEAN

EJEMPLO JASMINE

```
"use strict";

var rewire = require("rewire");

describe("The logger library", function() {
    var logger;

    beforeEach(function() {
        logger = rewire("../../../../app/lib/logger");
    });

    it("should exist", function() {
        expect(logger).toBeDefined();
    });

    it("should return test decorator in test environment", function() {
        var _getDecorator;

        _getDecorator = logger.__get__("_getDecorator");

        expect(_getDecorator()).toEqual("meanjs-template:test_environment");
    });

    // ...
});
```

Testing y depuración de aplicaciones MEAN

EJEMPLO TEST CREADO CON YEOMAN

```
"use strict";

describe("AppCtrl", function() {
    var $scope, controller;

    beforeEach(module(ApplicationConfiguration.applicationModuleName));

    beforeEach(inject(function($rootScope, $controller) {
        // create a new $scope for each test
        $scope = $rootScope.$new();

        // use the new $scope in creating the controller
        controller = $controller("AppCtrl", {
            $scope: $scope
        });

    }));

    it("should exist", function() {
        expect(controller).toBeDefined();
    });
});
```

Testing y depuración de aplicaciones MEAN

PROTRACTOR

OTRA CAPA QUE PERMITE TRABAJAR CON SELENIUM E INTERACTUAR CON LA CAPA WEB

COMPLEMENTA LOS TESTS DE APLICACIÓN

SE DEFINE DE FORMA SIMILAR A KARMA

PERMITE REALIZAR LAS PRUEBAS EN DISTINTOS NAVEGADORES

Testing y depuración de aplicaciones MEAN

Introducción a testing y depuración de aplicaciones MEAN

```
"use strict";

exports.config = {
    allScriptsTimeout: 11000,

    specs: [
        "test/client/e2e/*.js"
    ],

    multiCapabilities: [{
        browserName: "firefox"
    }, {
        browserName: "chrome"
    }],

    baseUrl: "http://localhost:3000/",

    beforeLaunch: function() {
        console.log("Starting setup...");

        // Return a promise when dealing with asynchronous
        // functions here (like creating users in the database)
    },

    afterLaunch: function() {
        console.log("Starting cleanup...");

        // Return a promise when dealing with asynchronous
        // functions here (like removing users from the database)
    },

    framework: "jasmine",

    jasmineNodeOpts: {
        defaultTimeoutInterval: 30000
    }
};
```

PROTRACTOR

SE PUEDE OBSERVAR QUE EXISTEN PRE Y POST PROCESADORES

PERMITE DEFINIR SI SE TIENEN QUE HACER TAREAS PREVIAS Y POSTERIORES A LOS TESTS

PERMITE DEFINIR EN QUE NAVEGADORES SE QUIERE EJECUTAR

EL FRAMEWORK A UTILIZAR EN LOS TESTS

TIMEOUT DE LAS PRUEBAS

Testing y depuración de aplicaciones MEAN

PROTRACTOR

EJEMPLO DE TEST

```
"use strict";

describe("meanjs-template App", function() {
    beforeEach(function() {
        // get the root of the project
        browser.get("/");
    });

    it("should display heading", function(done) {
        browser.getTitle().then(function(title) {
            expect(title).toEqual("meanjs-template");
            done();
        });
    });
});
```

Testing y depuración de aplicaciones MEAN

COMBINADO CON GRUNT

UNA VEZ CREADOS LOS TESTS SE MAPEAN EN GRUNT

DE ESA FORMA, PODEMOS INTEGRAR TODAS LAS PRUEBAS EN UN SOLO LUGAR CON

$ grunt test

PERMITE MOSTRAR TODA LA INFORMACIÓN DE LAS PRUEBAS GENERADAS

EJERCICIO

GENERE LAS PRUEBAS DE LA APLICACIÓN DE BIBLIOTECA ANTERIORMENTE GENERADA

DEBE EXISTIR UNA PRUEBA POR CONTROLADOR

SE DEBEN CUBRIR TODOS LOS MÉTODOS

CREATE / READ / UPDATE / DELETE / LIST / FIND_BY_ID