БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

ИИТ БГУИР

Программное обеспечение информационных технологий.

Дисциплина: «Тестирование веб-ориентированных приложений»

Отчет

По лабораторной работе №3

«ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЯ МЕТОДОМ БЕЛОГО/СЕРОГО ЯЩИКА»

Выполнил студент группы 481064 Сорока А.А.

Проверил преподаватель: Калитеня И. Л.

Минск 2016 г

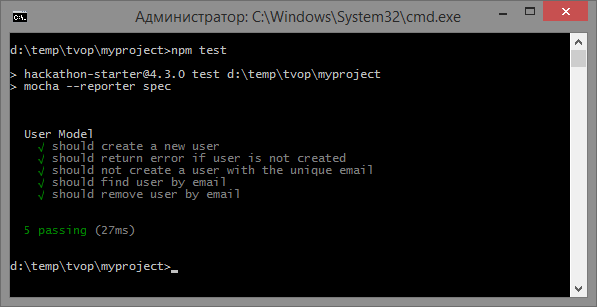
**Цель:**

1. Научиться тестирования веб-приложение методом белого/серого ящика, использую инструменты mocha.

**Задание:**

Протестировать веб-приложение методом белого/серого ящика.

**Скриншоты тестирования:**



**Код:**

const mongoose = require('mongoose');

const {expect} = require('chai');

const sinon = require('sinon');

require('sinon-mongoose');

const User = require('../models/User');

describe('User Model', () => {

it('should create a new user', (done) => {

const UserMock = sinon.mock(new User({ email: 'test@gmail.com', password: 'root' }));

const user = UserMock.object;

UserMock

.expects('save')

.yields(null);

user.save(function (err, result) {

UserMock.verify();

UserMock.restore();

expect(err).to.be.null;

done();

});

});

it('should return error if user is not created', (done) => {

const UserMock = sinon.mock(new User({ email: 'test@gmail.com', password: 'root' }));

const user = UserMock.object;

const expectedError = {

name: 'ValidationError'

};

UserMock

.expects('save')

.yields(expectedError);

user.save((err, result) => {

UserMock.verify();

UserMock.restore();

expect(err.name).to.equal('ValidationError');

expect(result).to.be.undefined;

done();

});

});

it('should not create a user with the unique email', (done) => {

const UserMock = sinon.mock(User({ email: 'test@gmail.com', password: 'root' }));

const user = UserMock.object;

const expectedError = {

name: 'MongoError',

code: 11000

};

UserMock

.expects('save')

.yields(expectedError);

user.save((err, result) => {

UserMock.verify();

UserMock.restore();

expect(err.name).to.equal('MongoError');

expect(err.code).to.equal(11000);

expect(result).to.be.undefined;

done();

});

});

it('should find user by email', (done) => {

const userMock = sinon.mock(User);

const expectedUser = {

\_id: '5700a128bd97c1341d8fb365',

email: 'test@gmail.com'

};

userMock

.expects('findOne')

.withArgs({ email: 'test@gmail.com' })

.yields(null, expectedUser);

User.findOne({ email: 'test@gmail.com' }, (err, result) => {

userMock.verify();

userMock.restore();

expect(result.email).to.equal('test@gmail.com');

done();

})

});

it('should remove user by email', (done) => {

const userMock = sinon.mock(User);

const expectedResult = {

nRemoved: 1

};

userMock

.expects('remove')

.withArgs({ email: 'test@gmail.com' })

.yields(null, expectedResult);

User.remove({ email: 'test@gmail.com' }, (err, result) => {

userMock.verify();

userMock.restore();

expect(err).to.be.null;

expect(result.nRemoved).to.equal(1);

done();

})

});

});