

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский политехнический университет»

Кафедра «Информативные технологии»
Образовательная программа «Веб-технологии»

Лабораторная работа № 8
по дисциплине «Программная инженерия»

Выполнил студент
группы 181-322 Холщёв Илья

Проверил:
преподаватель Будылина Евгения Александровна

Москва 2020

Модульное тестирование

Для тестирования веб-приложения были выбраны следующие инструменты: Jest, vue-test-utils, vue-jest.

Модульное тестирование — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы, наборы из одного или более программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки

Однофайловые компоненты Vue позволяют просто писать модульные тесты для компонентов в изоляции. Это помогает разрабатывать новую функциональность с уверенностью, что это не ломает работу существующей, и помогает другим разработчикам понять, как работает компонент.

Тесты на примере компонента формы входа

```
import { shallowMount, createLocalVue } from '@vue/test-utils'
import Vuex from 'vuex'
import Basket from '../components/BasketDude.vue'

const localVue = createLocalVue()

localVue.use(Vuex)

describe('Компонент Корзины', () => {
  let actions
  let store
  let mutations

  beforeEach(() => {
    actions = {
      addItemToBasket: jest.fn(),
      createOrder: jest.fn()
    };
    mutations = {
      incrementItemCount: jest.fn(),
      pushToBasket: jest.fn(),
      incrementItemCount: jest.fn(),
      decrementItemCount: jest.fn(),
      clearAllBasket: jest.fn(),
      setBasketVendor: jest.fn(),
    };

    store = new Vuex.Store({
      state: {
        basketItems: [],
        customerAddress: '',
        customerPhone: '',
        vendor: '',
      },
    });
  });
});
```

```

    getters: {
      getBasketItems: jest.fn(() => [{
        address: "Средняя Семеновская",
        count: 3,
        id: "YR43zDg2YTMRwurQiRSo",
        price: 526,
        title: "Сырный соус",
      }]),
      getTotalPrice: jest.fn(() => 1000),
      getBasketVendor: jest.fn(function () {
        return {name: 'Test Name'};
      })
    },
    actions,
    mutations
  })
})

test('Компонент является экземпляром Vue', () => {
  const wrapper = shallowMount(Basket, { store, localVue })
  expect(wrapper.findComponent(Basket).vm).toBeTruthy()
})

test('Компоненту присваивается название заведения', () => {
  const wrapper = shallowMount(Basket, { store, localVue })

  const heading = wrapper.find('#vendorName');
  expect(heading.text()).toBe('Test Name')
})

test('Компонент переводит на страницу оплаты', () => {
  const push = jest.fn();
  const $router = {
    push: jest.fn()
  }
  const $route = {
    name: "checkout"
  };
  const wrapper = shallowMount(Basket, { store, localVue, mocks: { $router, $route } })
  const button = wrapper.find('#checkout')

  button.trigger('click')

  expect(wrapper.vm.$route.name).toBe($route.name)
})

test('Компонент корзины правильно выводит сумму заказа', () => {
  const wrapper = shallowMount(Basket, { store, localVue })
  const button = wrapper.find('#checkout')

  expect(button.props('label')).toBe('Заказать 1000 ₺ ')
})

```

```
test('Компонент корзины очищается', () => {
  const wrapper = shallowMount(Basket, { store, localVue })
  const button = wrapper.find('#clear')

  button.trigger('click')

  expect(mutations.clearAllBasket).toHaveBeenCalled()
})

test('В корзине можно уменьшить количество товара', () => {
  const wrapper = shallowMount(Basket, { store, localVue })
  const buttonDecrement = wrapper.find('#decrement')

  buttonDecrement.trigger('click')

  expect(mutations.decrementItemCount).toHaveBeenCalled()
})

test('В корзине можно увеличить количество товара', () => {
  const wrapper = shallowMount(Basket, { store, localVue })
  const buttonDecrement = wrapper.find('#increment')

  buttonDecrement.trigger('click')

  expect(mutations.incrementItemCount).toHaveBeenCalled()
})
}
```

Результаты тестирования

C:\WINDOWS\system32\cmd.exe

```
C:\Users\iliah\Desktop\vue-delivery-app>npm run test
```

```
> vue-delivery-app@0.1.0 test C:\Users\iliah\Desktop\vue-delivery-app
> jest
```

```
PASS src/tests/LoginFormTests.spec.js
```

Компонент Входа

- ✓ Компонент является экземпляром Vue (98 ms)
- ✓ Кнопка отправки по умолчанию в статичном состоянии (30 ms)
- ✓ Компонент формы закрывается по клику (88 ms)
- ✓ Осуществляется переход на страницу регистрации (38 ms)
- ✓ Компонент формы отправляет данные на сервер (40 ms)

```
===== Coverage summary =====
```

```
Statements   : 100% ( 7/7 )
Branches     : 100% ( 0/0 )
Functions    : 100% ( 4/4 )
Lines        : 100% ( 7/7 )
```

```
=====
Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        9.21 s, estimated 11 s
Ran all test suites.
```

```
C:\Users\iliah\Desktop\vue-delivery-app>
```