



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

О Т Ч Е Т

по лабораторной работе № 9


Дисциплина: Языки интернет-программирования

Вариант: 23

Студент

ИУ6-33Б

(Группа)


(Подпись, дата)

Д. И. Мироненко

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В. Д. Шульман

(И.О. Фамилия)

Москва, 2023

Задание

При помощи Javascript модифицировать код ЛР 8 таким образом, чтобы для отображения результатов вычисления браузер не выполнял полную перезагрузку страницы.

Сформировать тесты для проверки работы программы при помощи Katalon Recorder / Selenium Webdriver.

Код программы

Файл “ app/controllers/calc_controller.rb”

```
class CalcController < ApplicationController
  def input
    render layout: false
  end

  def view
    input = params[:array]
    @longest_sequence = []
    @sequences = []

    return unless input&.match?(/\A\s*\d+(\s*\d+)*\s*\z/)

    array = input.split
    start = 0
    max_len = 0
    curr_len = 1
    array.length.times do |i|
      next if i.zero?

      if array[i - 1].to_i <= array[i].to_i
        curr_len += 1
      else
        @sequences << array[start, curr_len] if curr_len > 1

        if curr_len > max_len
          @longest_sequence = @sequences[-1]
          max_len = curr_len
        end

        start = i
        curr_len = 1
      end
    end

    @sequences << array[start, curr_len] if curr_len > 1

    if curr_len > max_len
      @longest_sequence = @sequences[-1]
      max_len = curr_len
    end

    @result = {
      longest_sequence: @longest_sequence,
      sequences: @sequences
    }
  end
end
```

```

    }
    respond_to do |format|
      format.html
      format.json do
        render json: {
          type: @result.class.to_s,
          value: @result
        }
      end
    end
  end
end
end
end

```

Файл “public/index.html”

```

<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html charset=utf-8">
    <title>AJAX Calc</title>
    <script>
      calc_url= location.protocol + "://" + location.host + "/calc/view.json"

      function send_query() {
        var frm = document.forms.calc_form;
        var param_str = "";

        if (frm.array.value?.trim()) {
          param_str = "?" + "array=" + frm.array.value;
        } else {
          alert("Enter the array of numbers!");
          return false;
        }
      }

      var my_JSON_object = {};
      var http_request = new XMLHttpRequest();
      http_request.open("GET", calc_url + param_str, true);
      http_request.onreadystatechange = function () {
        var done = 4, ok = 200;
        if (http_request.readyState == done && http_request.status == ok) {
          my_JSON_object = JSON.parse(http_request.responseText);
          show_result(my_JSON_object);
        }
      };

      http_request.send(null);

      return false;
    }

    function create_row(title, sequence) {
      let row = document.createElement("tr");

      let title_item = document.createElement("td");
      title_item.innerHTML = title;

      let value_item = document.createElement("td");
      if (sequence) {

```

```

        value_item.innerHTML = sequence.join(" ");
    }

    row.appendChild(title_item);
    row.appendChild(value_item);

    return row;
}

function show_result(data) {
    let table = document.getElementById("result_table");
    table.innerHTML = "";

    for (const sequence of data.value.sequences) {
        table.appendChild(create_row("Sequence Found:", sequence));
    }

    if (data.value.longest_sequence) {
        table.appendChild(create_row("Longest Sequence:", data.value.longest_sequence));
    } else {
        table.appendChild(create_row("Longest Sequence:", ["Not Found"]));
    }
}
</script>
</head>
<body>
    <form name="calc_form" method="get">
        <label>Array</label><br>
        <input id="array" type="text">
        <br><br>
        <input name="commit" type="submit" value="Get Result" onclick="return send_query();">
    </form>
    <div id="result"><br><table id="result_table"></table></div>
</body>
</html>

```

Файл “test/controllers/calc_controller_test.rb”

```

require 'test_helper'

class CalcControllerTest < ActionDispatch::IntegrationTest
  test 'should get input' do
    get calc_input_url
    assert_response :success
  end

  test 'should get view' do
    get calc_view_url
    assert_response :success
  end

  test 'should get 5 6 8 for view with 5 6 8 5 6 3' do
    get calc_view_url, params: { array: '5 6 8 5 6 3' }
    assert_equal assigns[:longest_sequence], %w[5 6 8]
  end

  test 'should get unknown for view with letters' do
    get calc_view_url, params: { array: 'j 3 5 e 4' }
  end
end

```

```

    assert_equal assigns[:longest_sequence], []
  end

  test 'should get unknown for view with empty array' do
    get calc_view_url, params: { array: '' }
    assert_equal assigns[:longest_sequence], []
  end
end

```

Файл “test/spec/regular_spec.rb”

```

require 'selenium-webdriver'
require 'json'

describe 'Regular' do
  before(:each) do
    @driver = Selenium::WebDriver.for :chrome
    @vars = {}
  end
  after(:each) do
    @driver.quit
  end
  it 'regular' do
    @driver.get('http://localhost:3000/')
    @driver.find_element(:id, 'array').send_keys('3 6 5 17 26 21 9 15 7')
    @driver.find_element(:name, 'commit').click
    sleep(0.2)
    html_content = @driver.find_element(:id, 'result_table')['innerHTML']
    expect(html_content).to include('<tr><td>Sequence Found:</td><td>3 6</td></tr><tr><td>Sequence Found:</td><td>5 17 26</td></tr><tr><td>Sequence Found:</td><td>9 15</td></tr><tr><td>Longest Sequence:</td><td>5 17 26</td></tr>')
  end
end

```

Файл “test/spec/onenumber_spec.rb”

```

require 'selenium-webdriver'
require 'json'

describe 'Onenumber' do
  before(:each) do
    @driver = Selenium::WebDriver.for :chrome
    @vars = {}
  end
  after(:each) do
    @driver.quit
  end
  it 'onenumber' do
    @driver.get('http://localhost:3000/')
    @driver.find_element(:id, 'array').click
    @driver.find_element(:id, 'array').send_keys('1')
    @driver.find_element(:name, 'commit').click
    expect(@driver.find_element(:id, 'result_table').text).to eq('Longest Sequence: Not Found')
  end
end

```

Файл “test/spec/empty_spec.rb”

```
require 'selenium-webdriver'
require 'json'

describe 'Empty' do
  before(:each) do
    @driver = Selenium::WebDriver.for :chrome
    @vars = {}
  end
  after(:each) do
    @driver.quit
  end
  it 'empty' do
    @driver.get('http://localhost:3000/')
    @driver.find_element(:id, 'array').click
    @driver.find_element(:name, 'commit').click
    expect(@driver.switch_to.alert.text).to eq('Enter the array of numbers!')
  end
end
```

Файл “test/spec/spacesonly_spec.rb”

```
require 'selenium-webdriver'
require 'json'

describe 'Spacesonly' do
  before(:each) do
    @driver = Selenium::WebDriver.for :chrome
    @vars = {}
  end
  after(:each) do
    @driver.quit
  end
  it 'spacesonly' do
    @driver.get('http://localhost:3000/')
    @driver.find_element(:id, 'array').click
    @driver.find_element(:id, 'array').send_keys('    ')
    @driver.find_element(:name, 'commit').click
    expect(@driver.switch_to.alert.text).to eq('Enter the array of numbers!')
  end
end
```

Работа программы

Array

3 6 5 17 26 21 9 15 7

Get Result

Рисунок 1 – ввод значений

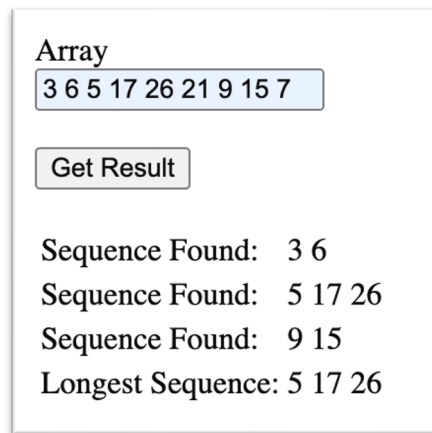


Рисунок 2 – вывод результатов

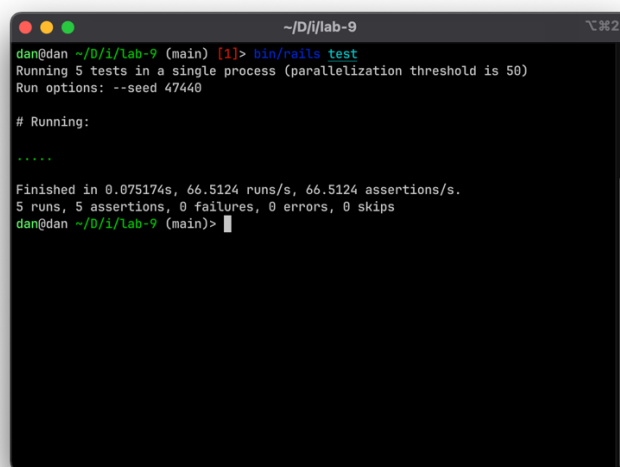


Рисунок 3 – результат выполнения функциональных тестов

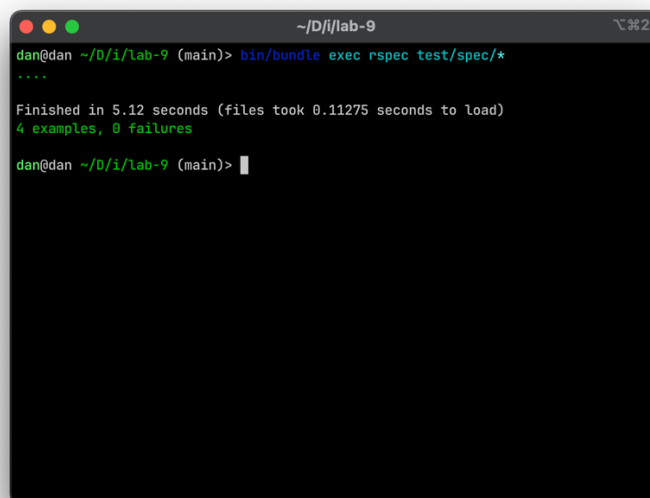


Рисунок 4 – результат выполнения selenium тестов

Вывод

Изучена технология AJAX для динамического отображения контента на web-странице; полученные знания отработаны при создании простого web-сайта; написаны и выполнены функциональные тесты, а также тесты, записанные в selenium.