



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 09.03.03 Прикладная информатика

О Т Ч Е Т

по лабораторной работе № 9

Название: Создание простейших веб-приложений
RUBY ON RAILS. AJAX

Дисциплина: Языки Интернет-программирования

Студент

ИУ6-35Б

(Группа)

10.11.2023

(Подпись, дата)

В. И. Мамыкин

(И.О. Фамилия)

Преподаватель

Е.Ю. Гаврилова

(Подпись, дата)

(И.О. Фамилия)

Москва, 2023

Задание:

1. Используйте веб-приложение, разработанное при выполнении практикума № 5 (расчетная задача, выполненная по индивидуальному заданию, которая требует ввод параметров, а результатом ввода является таблица).
2. Выберите способ добавления асинхронного запроса и реализуйте.
3. Отладьте веб-приложение.
4. С помощью Selenium IDE запишите тест для проверки при нескольких значениях выходных параметров.
5. Проверьте выполнение теста.

Цель: углубление теоретических сведений о принципах работы асинхронного веб-интерфейса и получение практических навыков создания веб-приложения с использованием средств Ruby on Rails и технологии AJAX.

sequences_controller.rb

```
# frozen_string_literal: true

class SequencesController < ApplicationController
  def calculate
    longest_subsequence = [] # Самая длинная подпоследовательность
    current_subsequence = [] # Текущая подпоследовательность
    all_subsequences = [] # Все подпоследовательности
    # unless params[:v2].nil?
    sequence = params[:v2]&.split(' ').&.map(&:to_i)
    sequence&.each do |number|
      if (Math.sqrt(number) % 1).zero?
        # Если число является полным квадратом
        current_subsequence << number
      else
```

```

# Если число не является полным квадратом

longest_subsequence = current_subsequence.clone if
current_subsequence.length > longest_subsequence.length

cur = current_subsequence.clone
all_subsequences << cur.join(' ')
all_subsequences.pop if all_subsequences[all_subsequences.size - 1] == "
current_subsequence = []
end
end
cur = current_subsequence.clone
all_subsequences << cur.join(' ') if cur.length.positive?
longest_subsequence = current_subsequence.clone if current_subsequence.length
> longest_subsequence.length
all_subsequences.pop if all_subsequences[all_subsequences.size - 1] == "
subsequence_count = longest_subsequence.length
@number = [params[:v1], sequence&.size.to_s]
@result = [all_subsequences, longest_subsequence.join(' '),
subsequence_count.to_s, sequence&.join(' ')]

# end
# @number = [params[:v1], 0]
# @result = [all_subsequences, ", '0', "]
logger.debug "from view controller: #{@result}"
respond_to do |format|
  format.html
  format.json do
    render json:
      { type: @result.class.to_s, value1: @result, value2: @number }
  end
end
end
end
end

```

end

sequences_controller_test.rb

require 'test_helper'

class SequencesControllerTest < ActionDispatch::IntegrationTest

test 'should get calculate' do

get sequences_calculate_url

assert_response :success

end

test 'should get sequence' do

get sequences_calculate_url, params: { v1: '10', v2: '1 2 3 4 9 16 3 2 4 10' }

assert_equal assigns[:result], [['1', '4 9 16', '4'], '4 9 16', '3', '1 2 3 4 9 16 3 2 4 10']

end

test 'should get sequence - 2' do

get sequences_calculate_url, params: { v1: '3', v2: '1 4 4' }

assert_equal assigns[:result], [['1 4 4'], '1 4 4', '3', '1 4 4']

end

end

calculate.html.erb

<h1>Sequences#input</h1>

<p>Find me in app/views/sequences/input.html.erb</p>

<form action="/sequences/calculate" method="get" data-remote="true"
id="calc_form" accept-charset="UTF-8" data-type="json">

<label for="v1" > Число: </label>

```

<input type="text" id="v1" name="v1" pattern="^([0-9])+" value="10" required/>
<label for="v2" style="margin-left: 20px"> Последовательность: </label>
<input type="text" id="v2" name="v2" pattern="^([0-9]+[\s]{0,1})+" value="1 2
3 4 9 16 3 2 4 10" required/>
<button>Найти!</button>
</form>
<div id="result"> </div>

```

calc.js

```

function formation_table (data) {

    let tableHTML = "<table>"; // Начало таблицы

    if (data.value2[0] == data.value2[1]) {

        if (data.value1[2] != '0') {

            tableHTML +=

                '<tr><td style="border: 2px solid darkblue;"><strong>Введенная
последовательность:</strong></td></tr>' +

                '<tr><td style="border: 2px solid darkblue;">' + data.value1[3] +
                '</td></tr>' +

                '<tr><td style="border: 2px solid
darkblue;"><strong>Подпоследовательности:</strong></td></tr>';

            data.value1[0].forEach(function (res) {

                tableHTML += '<tr><td style="border: 2px solid darkblue;">' + res +
                '</td></tr>';

            });

```

```

tableHTML +=

    '<tr><td style="border: 2px solid darkblue;"><strong>Самая длинная
подпоследовательность:</strong></td></tr>' +

    '<tr><td style="border: 2px solid darkblue;">' + data.value1[1] +
    '</td></tr>' +

    '<tr><td style="border: 2px solid darkblue;"><strong>Ее
длина:</strong></td></tr>' +

    '<tr><td style="border: 2px solid darkblue;">' + data.value1[2] +
    '</td></tr>';

    } else {

        tableHTML +=

            '<tr><td style="border: 2px solid darkblue;"><strong>Введенная
последовательность:</strong></td></tr>' +

            '<tr><td style="border: 2px solid darkblue;">' + data.value1[3] +
            '</td></tr>' +

            '<tr><td style="border: 2px solid
darkblue;"><strong>Подпоследовательностей квадратов натуральных чисел
нет</strong></td></tr>';

        }

    } else {

        tableHTML +=

            '<tr><td style="border: 2px solid darkblue;"><strong>Введенная
последовательность:</strong></td></tr>' +

```

```

        '<tr><td style="border: 2px solid darkblue;">' + data.value1[3] + '</td></tr>'
+
        '<tr><td style="border: 2px solid darkblue;"><strong>Введенное число
n:</strong></td></tr>' +
        '<tr><td style="border: 2px solid darkblue;">' + data.value2[0] + '</td></tr>'
+
        '<tr><td style="border: 2px solid darkblue;"><strong>Число n не
совпадает с длиной введенной последовательности!</strong></td></tr>';
    }

```

```

    tableHTML += "</table>"; // Конец таблицы
    return "<hr/>" + tableHTML + "<hr/><p id='date'" + Date() + "</p>"
}

```

```

function show_result(data) {
    console.log(data);
    const result = document.getElementById("result");
    result.innerHTML = formation_table(data); // Вставляем сформированную
таблицу в элемент с id "result"
}

```

```

$(document).ready(function () {
    $("#calc_form").bind("ajax:success",
        function (xhr, data, status) {

```

```
// data is already an object

show_result(data);

})

});
```

Sequences#input

Find me in app/views/sequences/input.html.erb

Число: Последовательность:

Введенная последовательность:
1 2 3 4 9 16 3 2 4 10
Подпоследовательности:
1
4 9 16
4
Самая длинная подпоследовательность:
4 9 16
Ее длина:
3

Sat Jan 13 2024 20:35:23 GMT+0300 (Москва, стандартное время)

Рисунок 1 – страница ввода/вывод результата

```
# Running:

...

Finished in 1.529950s, 1.9608 runs/s, 1.9608 assertions/s.
3 runs, 3 assertions, 0 failures, 0 errors, 0 skips
```

Рисунок 2 – результат функционального тестирования

Вывод: были изучены теоретические сведения о принципах работы асинхронного веб-интерфейса и получены практические навыки создания веб-приложения с использованием средств Ruby on Rails и технологии AJAX.