



**«Московский государственный технический университет
имени Н.Э. Баумана»**

(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

Дисциплина: Языки Интернет-программирования

Название лабораторной работы: Ajax. Вариант 10.

Студент гр. ИУ6-34Б

К.Ю. Каташинский

Преподаватель

Москва, 2020

Условие задания:

1. При помощи Javascript модифицировать код ЛР 8 таким образом, чтобы для отображения результатов вычисления браузер не выполнял полную перезагрузку страницы.
2. Сформировать тесты для проверки работы программы при помощи Katalon Recorder / Selenium Webdriver.

Код файла base_controller.rb:

```
# frozen_string_literal: true
# BaseController class
class BaseController < ApplicationController
  def index; end
  def create
    number = params[:number].to_i
    array = params[:str].split.map{|elem| elem.to_i}
    @error = number != array.length ? 'Введите n чисел' : ''
    @all_segments = array.chunk_while do |first, second|
      multiply_five?(first) == multiply_five?(second)
    end.find_all {|segment| multiply_five?(segment[0]) }
    @has_elements = (@all_segments.length > 0)
    @count = @all_segments.length
    @max_segment = @all_segments.max_by { |segment| segment.length }
    @all_segments = @all_segments.join(" ")
    @max_segment = @max_segment&.join(" ")
    @array = array.join(" ")
    respond_to do |format|
      format.html
      format.json do
        render json:
          { array: @array, all_segments: @all_segments, max_segment: @max_segment,
count: @count, error: @error }
        end
      end
    end
  end
  private def multiply_five?(elem)
    x = Math.log(elem, 5)
    elem != 1 ? (x.round == x) : false
  end
end
```

Код файла application.js:

```
handleAjaxSuccess = function(event) {
  [data, status, xhr] = event.detail;
  show_result(data);
}
document.addEventListener('DOMContentLoaded', function() {
  document.querySelector("#add").addEventListener(
    'ajax:success', handleAjaxSuccess)
})
function show_result(data)
{
  if (data.error)
  {
    document.getElementById("error").innerText = data.error;
    return;
  }
  document.getElementById("error").innerText = "";
}
```

Продолжение кода файла application.js:

```
var table = document.getElementById("add_table");
var tr = document.createElement("tr");
if (data.all_segments)
    tr.innerHTML = '<td class="col">' + data.array + ' </td>' + '<td class="col">'
+ data.all_segments + '</td>' +
    '<td class="col">' + data.max_segment + '</td>' + '<td class="col">' +
data.count + '</td>';
else
    tr.innerHTML = '<td class="col">' + data.array + ' </td>' + '<td class="col"
colspan="3" style="color: red;">Нет отрезков</td>';
table.appendChild(tr);
}
```

Код файла index.html.erb:

```
<%= form_with(url: "/create.json", method: "post", id: "add") do |f| %>
  <table id="add_form">
    <caption><h1>Форма ввода</h1></caption>
    <tr>
      <td width="40%">Введите число n</td>
      <td><%= text_field_tag :number, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td>Введите n чисел через пробел</td>
      <td><%= text_field_tag :str, nil, class: "enter" %></td>
    </tr>
    <tr>
      <td><%= submit_tag "Найти", class: "btn" %></td>
      <td><span style="color: red;" id="error"></span></td>
    </tr>
  </table>
<% end %>
<table id="add_table">
  <caption><h1>Таблица вывода</h1></caption>
  <tr>
    <td class="col">Входные данные</td>
    <td class="col">Полученные отрезки</td>
    <td class="col">Максимальный отрезок</td>
    <td class="col">Количество</td>
  </tr>
</table>
```

Код файла base_controller_test.rb:

```
require 'test_helper'
class BaseControllerTest < ActionDispatch::IntegrationTest
  test "should get index" do
    get "/"
    assert_response :success
  end
  test "should get 5 5 25 for create entered data" do
    post "/create", params: { number: 10, str: "1 5 5 25 7 8 10 5 125 10" }
    assert_equal assigns[:max_segment], "5 5 25"
  end
  test "should get Введите n чисел for incorrect params" do
    post "/create", params: { number: 2, str: "1 2 3 4 5" }
    assert_equal assigns[:error], "Введите n чисел"
  end
end
```

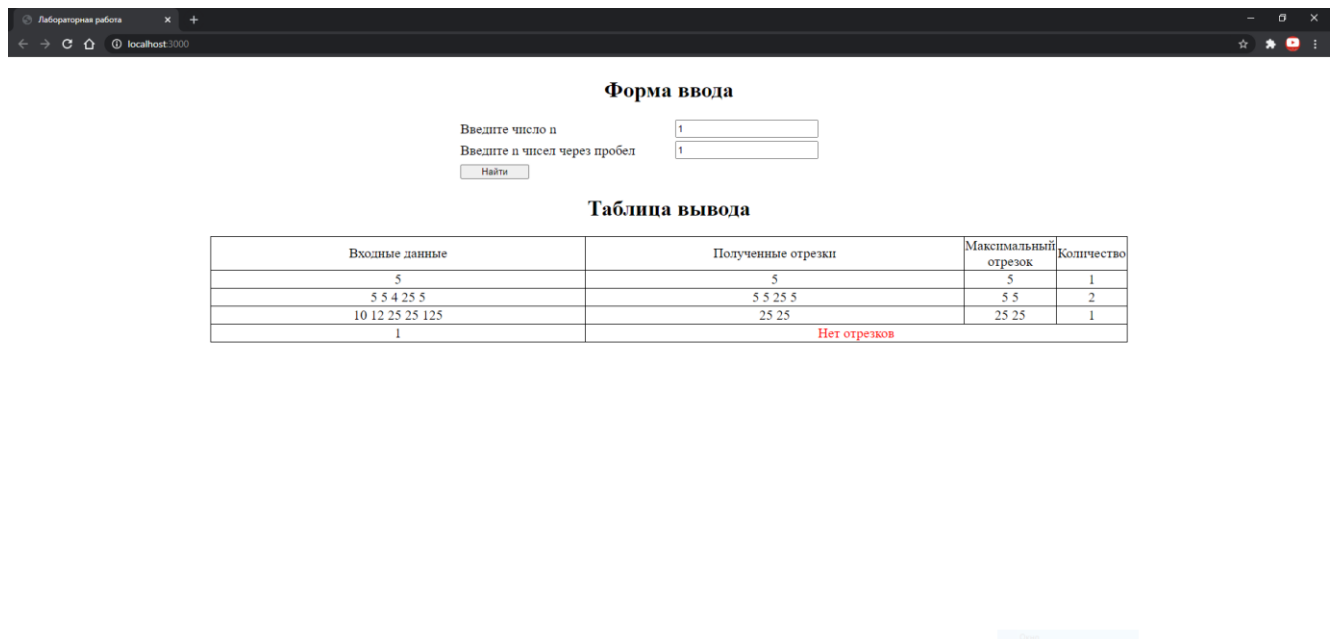


Рисунок 1 «Результаты выполнения программы»

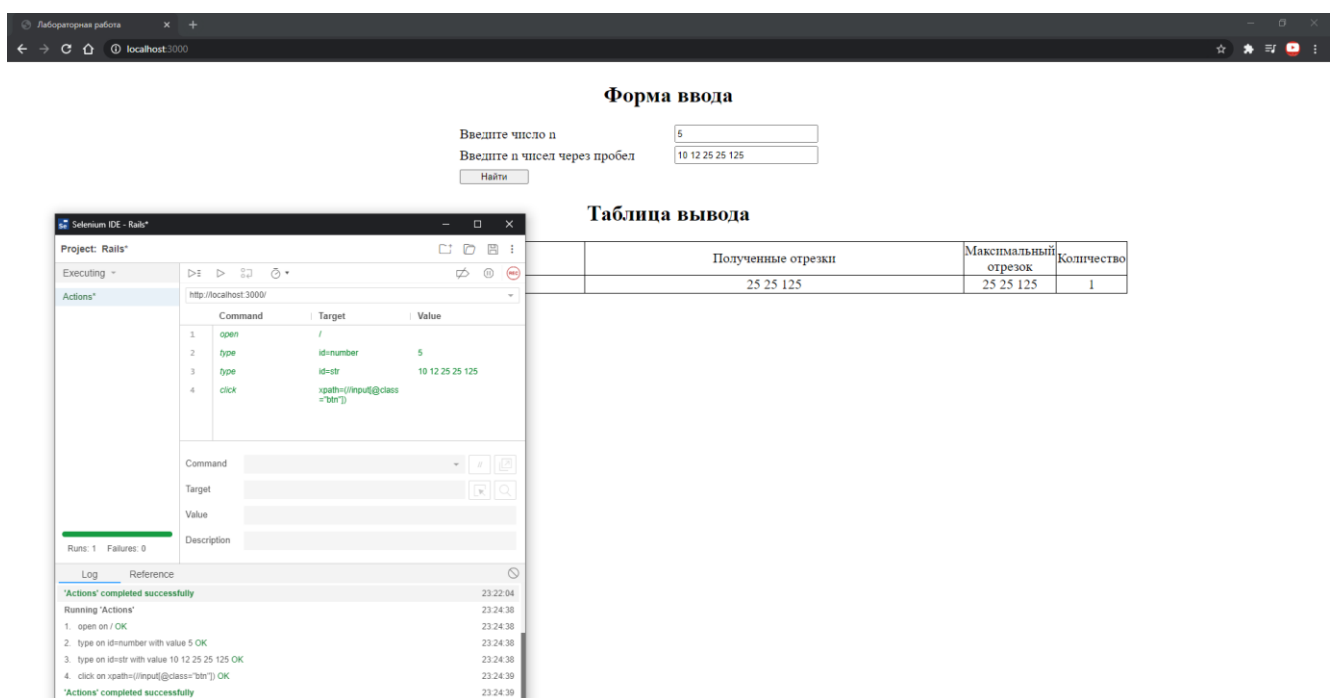


Рисунок 2 «Тестирование приложения с помощью Selenium IDE »

Вывод

В ходе данной лабораторной работы были изучены основные инструменты технологии ajax, а также средства тестирования приложения с помощью Selenium IDE.