



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

Название: СОЗДАНИЕ ПРОСТЕЙШИХ ВЕБ-ПРИЛОЖЕНИЙ RUBY ON
RAILS. AJAX

Дисциплина: Языки интернет-программирования

Студент

ИУ6-32Б

(Группа)

(Подпись, дата)

Джафаров Э.Э.

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Петрова Я.С.

(И.О. Фамилия)

Москва, 2022

Цель работы — 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.

Задание:

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.

Решение:

1. Контроллер calc_controller.rb

```
1 class CalcController < ApplicationController
2   def input
3     render layout: false
4   end
5
6   def show
7     @right_numbers = []
8     @a = params[:power].to_i
9     (10**(@a - 1)...(10**@a)).step(1) do |number|
10      @right_numbers.push(number) if find(number) == number
11    end
12    rjson
13  end
14
15  def rjson
16    respond_to do |format|
17      format.html
18      format.json do
19        render json:
20          { length: @right_numbers.length, value: @right_numbers }
21      end
22    end
23  end
24
25  def find(number)
26    summa = 0
27    @a.times do
28      tri = number % 10
29      number /= 10
30      summa += tri**@a
31    end
32    summa
33  end
34 end
```

2. input.html.erb

```
lab work > IPL-Laboratory9 > app > views > calc > input.html.erb
1 <h1>Здравейте</h1>
2 <%= form_tag('/calc/show.json', :method => "get", :remote => true, id: 'input_form') do %>
3   <%= label_tag("Введите количество цифр в числе:") %>
4   <br/>
5   <%= number_field_tag(:power, nil, in: 1..6, step:1)%>
6   <br/>
7   <%= submit_tag("Yosh") %>
8 <% end %>
9 <div id="result">
10 </div>
11
```

3. routes.rb

```
1 Rails.application.routes.draw do
2   root 'calc#input'
3   get 'calc/show'
4   # Define your application routes per the DSL in https://guides.r
5
6   # Defines the root path route ("/")
7   # root "articles#index"
8 end
9
```

4. index.html

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6      <title>Четыре</title>
7      <script>
8          input_url = location.protocol + "://" + location.host + "/calc/show.json"
9          function send_query() {
10              var frm = document.forms.input_form;
11              param_str = "?" +
12                  "power=" + frm.power.value + "&commit=Yosh"
13              if (param_str == "") return false;
14              var my_JSON_object = {};
15              var http_request = new XMLHttpRequest();
16              http_request.open("GET", input_url + param_str, true);
17              http_request.onreadystatechange = function () {
18                  var done = 4, ok = 200;
19                  if (http_request.readyState == done &&
20                      http_request.status == ok) {
21                      my_JSON_object =
22                          JSON.parse(http_request.responseText);
23                      show_result(my_JSON_object);
24                  }
25              };
26              http_request.send(null);
27              return false;
28          }
```

```
29  function show_result(data) {
30      var result = document.getElementById("result");
31      var table = document.createElement("table");
32      var tr = document.createElement('tr');
33      table.setAttribute("border", 1)
34      result.innerHTML = "<hr/>Result is:<br>";
35      var th1 = document.createElement('th'); // это ячейка заголовка
36      var text1 = document.createTextNode("Порядковый номер"); // текст заголовка
37      var th2 = document.createElement('th');
38      var text2 = document.createTextNode("Число");
39      th1.appendChild(text1); // прикрепляем текст к заголовку
40      th2.appendChild(text2);
41      tr.appendChild(th1); // прикрепляем заголовок к ряду
42      tr.appendChild(th2);
43      table.appendChild(tr); // прикрепляем ряд к таблице
44      for (let i = 0; i < data.length; i++) { // это длина моего массива, которое я передал через контроллер
45          var tr = document.createElement('tr'); // создаем ряд
46          var td1 = document.createElement('td'); // ячейку
47          var td2 = document.createElement('td'); // вторую ячейку
48          var text1 = document.createTextNode(i); // пишем текст
49          var text2 = document.createTextNode(data.value[i]);
50          td1.appendChild(text1); // прикрепляем текст к ячейке
51          td2.appendChild(text2);
52          td2.setAttribute("id", i)
53          td1.setAttribute("align", "center") // устанавливаем атрибут, чтобы текст был в центре
54          td2.setAttribute("align", "center")
55          tr.appendChild(td1); // прикрепляем ячейку к ряду
56          tr.appendChild(td2);
57          table.appendChild(tr); // прикрепляем ряд к таблице
58      }
59      result.appendChild(table)
60      result.innerHTML += "<p>Количество найденных чисел: " + data.length + "</p><hr/>"
61  }
62  </script>
63  </head>
```

5. Тест Selenium

	Command	Target	Value
1	open	/	
2	type	id=power	3
3	click	id=Yosh	
4	pause		30000
5	store text	id=0	x
6	verify	x	153
7	store text	id=1	x
8	verify	x	370
9	store text	id=2	x
10	verify	x	371
11	store text	id=3	x
12	verify	x	407
13	store text	css=p	x
14	verify	x	Количество найденных чисел: 4

6. Результат теста:

	Command	Target	Value
1	✓ open	/	
2	✓ type	id=power	3
3	✓ click	id=Yosh	
4	✓ pause		30000
5	✓ store text	id=0	x
6	✓ verify	x	153
7	✓ store text	id=1	x
8	✓ verify	x	370
9	✓ store text	id=2	x
10	✓ verify	x	371
11	✓ store text	id=3	x
12	✓ verify	x	407
13	✓ store text	css=p	x
14	✓ verify	x	Количество найденных чисел: 4

7. Результат работы:

7.1 Страница ввода:

Введите количество цифр:

Yosh

[Ссылочка \(в Сибирь\).](#)

7.2 После нажатия на кнопку:

Введите количество цифр:

Yosh

[Ссылочка \(в Сибирь\)](#)

Result is:

Порядковый номер	Число
0	153
1	370
2	371
3	407

Количество найденных чисел: 4

8. Проверка Rubocop:

```
k/IPL-Laboratory9$ rubocop -A app/controllers/calc_controller.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
```

Вывод:

Я познакомился с AJAX и научился работать с асинхронными запросами