*Федеральное государственное бюджетное образовательное учреждение* *высшего профессионального образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

**Отчет**

**по лабораторной работе № 9**

**Дисциплина: Языки интернет-программирования**

**Название лабораторной работы:** **AJAX**

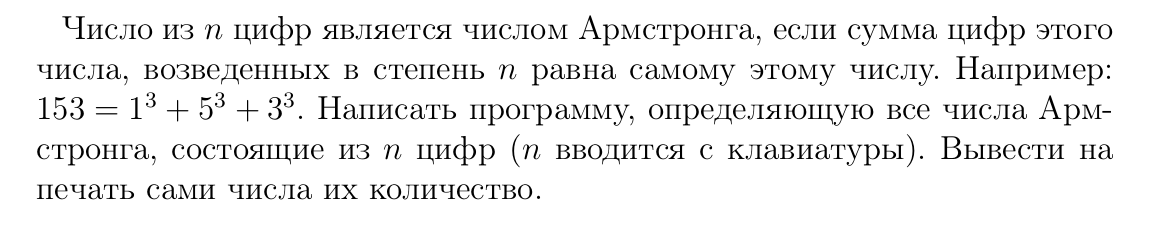
Студент гр. \_\_ИУ6-34 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лавренов И.С.**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2017

Задание

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

Файл output.html.erb:

<h3>Результат</h3>

<% if @result != :ok%> <%= t "results.#{@result}", :number => @number %>

<%else%>

<style>

table{ width: 300px;

border: 2px solid;

text-align: center; }

td{border: 1px solid;}

</style>

<table >

<tr><th >Номер</th><th >Полученное число</th></tr>

<%count = 0%>

<%@numbers.each do |i|%>

<tr><td ><%=@numbers.index(i) + 1%></td><td><%=i.to\_s%></td></tr>

<%count += 1%>

<%end%>

</table>

<p><%="Количество чисел Армстронга для чисел с #{@result[0].to\_s.length} цифрами(-ой): #{count}"%> </p>

<%end%>

<br/>

<%= link\_to "Повторить вычисление", :lr9\_input %>

Файл output\_controller.rb:

class Lr9Controller < ApplicationController

def input

end

def output

word = params[:word]

@numbers=Array.new

count = 0

if word.to\_s.empty?

@result = :blank #"Заполните пустое поле"

elsif word.to\_i == 0

@result = :nan #"Введите числовое значение"

else

@number = word.to\_i

(10\*\*(@number-1)..10\*\*@number - 1).to\_a.each do |ii|

wanted = ii

sum = 0

@number.times do

sum += (ii % 10)\*\*@number

ii /= 10

end

if wanted == sum

@numbers[count] = sum

count += 1

end

@result = @numbers.any? ? :ok : :not\_found #numbers : "Для чисел с #{number} цифрами нет чисел Армстронга"

end

end

respond\_to do |format|

format.html

format.json do

render json:

{type: @numbers.class.to\_s,number: @number, message: @result, values: @numbers}

end

end

end

end

Файл lr9.js:

function show\_result(data){

let result = document.getElementById('result');

msg = document.createElement('div');

if (data.message == 'ok'){

let table = document.createElement('table');

table.border = 1;

let trnew = document.createElement('tr');

let th1 = document.createElement('th'); th1.innerText = "Номер";

let th2 = document.createElement('th'); th2.innerText = "Полученное число";

table.style.textAlign = "center";

trnew.appendChild(th1);

trnew.appendChild(th2);

table.appendChild(trnew);

let tdnew1 = document.createElement('td');

let tdnew2 = document.createElement('td');

data.values.forEach(function(value, index){

tdnew1 = document.createElement("td");

tdnew1.innerText = (index + 1).toString();

tdnew2 = document.createElement("td");

tdnew2.innerText = value.toString();

let trnew = document.createElement('tr');

trnew.appendChild(tdnew1);

trnew.appendChild(tdnew2);

table.appendChild(trnew);

})

result.appendChild(table);

msg.innerHTML = '<br>' + 'Количество чисел Армстронга для чисел с ' + data.number.toString() + ' цифрами(-ой): ' + data.values.length.toString();

}

else{

if (data.message == 'not\_found')

msg.innerText = 'Для чисел с ' + data.number.toString() + ' цифрами нет чисел Армстронга';

else{

if (data.message == 'blank'){

msg.innerText = 'Заполните пустое поле';

}

if (data.message == 'nan'){

msg.innerText = 'Введите числовое значение';

}

}

}

result.appendChild(msg);

}

$(document).ready(function() {

$('#lr9\_form').bind("ajax:success", function (xrs, data, status) {

$('#table').empty();

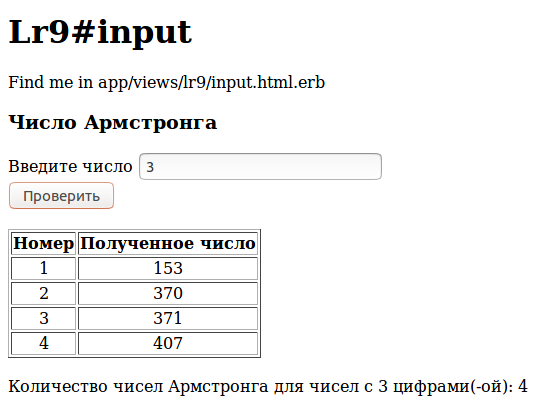
$('#result').empty();

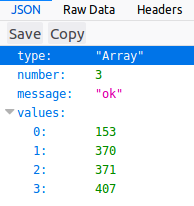
show\_result(xrs.detail[0]);

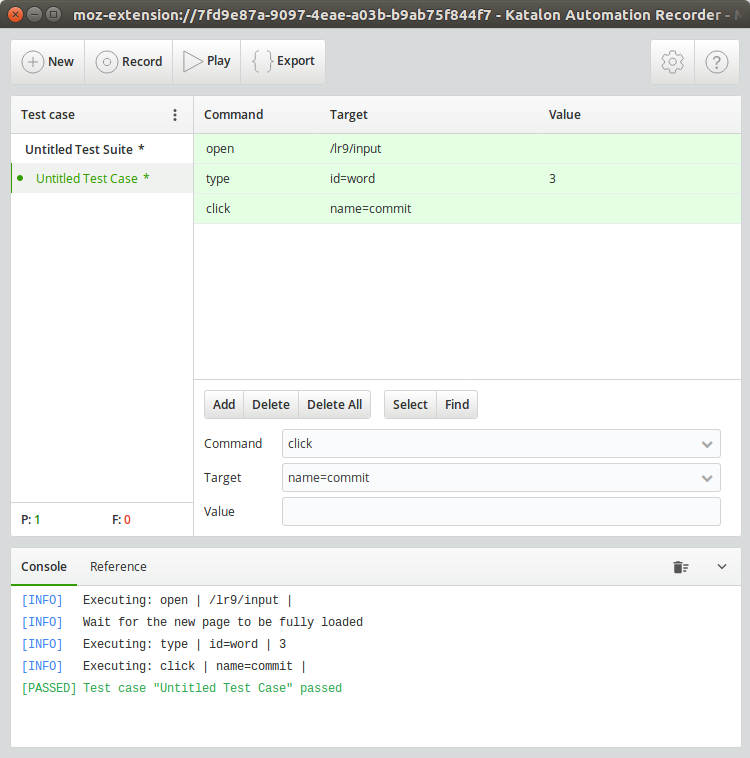
})

});

Результаты







Сгенерированный код в каталоне:

require "json"

require "selenium-webdriver"

require "rspec"

include RSpec::Expectations

describe "UntitledTestCase" do

before(:each) do

@driver = Selenium::WebDriver.for :firefox

@base\_url = "https://www.katalon.com/"

@accept\_next\_alert = true

@driver.manage.timeouts.implicit\_wait = 30

@verification\_errors = []

end

after(:each) do

@driver.quit

@verification\_errors.should == []

end

it "test\_untitled\_test\_case" do

@driver.get(@base\_url + "/lr9/input")

@driver.find\_element(:id, "word").clear

@driver.find\_element(:id, "word").send\_keys "3"

@driver.find\_element(:name, "commit").click

end

def element\_present?(how, what)

${receiver}.find\_element(how, what)

true

rescue Selenium::WebDriver::Error::NoSuchElementError

false

end

def alert\_present?()

${receiver}.switch\_to.alert

true

rescue Selenium::WebDriver::Error::NoAlertPresentError

false

end

def verify(&blk)

yield

rescue ExpectationNotMetError => ex

@verification\_errors << ex

end

def close\_alert\_and\_get\_its\_text(how, what)

alert = ${receiver}.switch\_to().alert()

alert\_text = alert.text

if (@accept\_next\_alert) then

alert.accept()

else

alert.dismiss()

end

alert\_text

ensure

@accept\_next\_alert = true

end

end