

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

по лабораторной работе № 9

Дисциплина: Языки интернет программирования

Студент	<u>ИУ6-31Б</u>	<u>(Подпись, дата)</u>	<u>Т.Е. Старжевский</u>
	(Группа)		(И.О. Фамилия)
Преподаватель		<u>(Подпись, дата)</u>	<u>Д. В. Малахов</u>
			(И.О. Фамилия)

Москва, 2022

Цель: Научиться выполнять AJAX запросы.

Текст 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.

1) Создаем аналогичное приложение 8 лабы, единственное, контроллер назовем palindrom, вместо palindroms, а также удалим базовые тесты, так как будем использовать Rspec, capybara, Selenium Webdriver.

```
rails new Rails_Ajax_TurboStream -T
```

```
rails g controller palindrom:input show
```

2) Сразу оформим представление input, show переведем в post запрос, подключим стили и оформим контроллер:

config/routes.rb

```
Rails.application.routes.draw do
  root "palindrom#input"
  post 'palindrom/show'
  # Define your application routes per the DSL in https://guides.rubyonrails.org/routing.html

  # Defines the root path route ("/")
  # root "articles#index"
end
```

app/views/palindrom/input.html.erb

```
<h1 class="color-text center">Palindroms</h1>
<p class="center">For example 202 and 202 ** 2 = 40804</p>
<%= form_with url: "/palindrom/show", class: "center",
id: "palindrom_form",
method: :post do |form| %>
  <%= form.label :n, "Input end of palindrom`s range:", class: "center" %>
  <%= form.text_field :n, class: "text-field", value: "1568" %>
  <%= form.submit "Show", class:"button" %>
<% end %>
<div id="result" class="table"></div>
```

app/assets/stylesheets/application.css

```
/*
```

* This is a manifest file that'll be compiled into application.css, which will include all the files
* listed below.
*
* Any CSS (and SCSS, if configured) file within this directory, lib/assets/stylesheets, **or** any
plugin's
* vendor/assets/stylesheets directory can be referenced here using a relative path.
*
* You're free to add application-wide styles to this file **and** they'll appear at the bottom of the
* compiled file so the styles you add here take precedence over styles defined in any other
CSS
* files in this directory. Styles in this file should be added after the last require_* statement.
* It is generally better to create a new file per style scope.
*

```
*/  
body {  
  background: #000;  
  color: #fff;  
  font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen,  
  Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;  
}  
  
.table {  
  font-size: 20px;  
  color: #fff;  
  text-align: center;  
}  
.table .row {  
  border-bottom: 1px solid #444;  
}  
  
.color-text {  
  color: #fc4f83;  
}  
  
.error-text {  
  font-size: 20px;  
  color: red;  
}  
.error-text strong {  
  color: #fc4f83  
}  
  
.center {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}  
.text-field {  
  margin: 0px 10px;  
  padding: 2px 6px;
```

```

width: 150px;
border-radius: 10px;
font-size: 20px;
font-weight: 800;
outline: none;
}
.button {
position: relative;
text-align: center;
border-radius: 10px;
padding: 5px 6px;
font-size: 18px;
font-weight: 800;
transition: 0.5s;

margin: 0px 10px;
color: #cecece;
border: 2px solid #cecece;
background: #000;
}
.button:hover {
cursor: pointer;
color: #fff;
border: 2px solid #fff;
}

```

app/views/layouts/application.html.erb (Подключим Bootstrap)

```

<!DOCTYPE html>
<html>
<head>
<title>RailsAjaxTurbostream</title>
<meta name="viewport" content="width=device-width,initial-scale=1">
<%= csrf_meta_tags %>
<%= csp_meta_tag %>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/
et3URy9Bv1WTRi" crossorigin="anonymous">
<%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>
<%= javascript_importmap_tags %>
</head>

<body>
<%= yield %>
</body>
</html>

```

app/controllers/palindrom_controller.rb

```

class PalindromController < ApplicationController
  def input; end

  def show
    @result = palindroms_before(params[:n].to_i)
    @error = require_params
  end

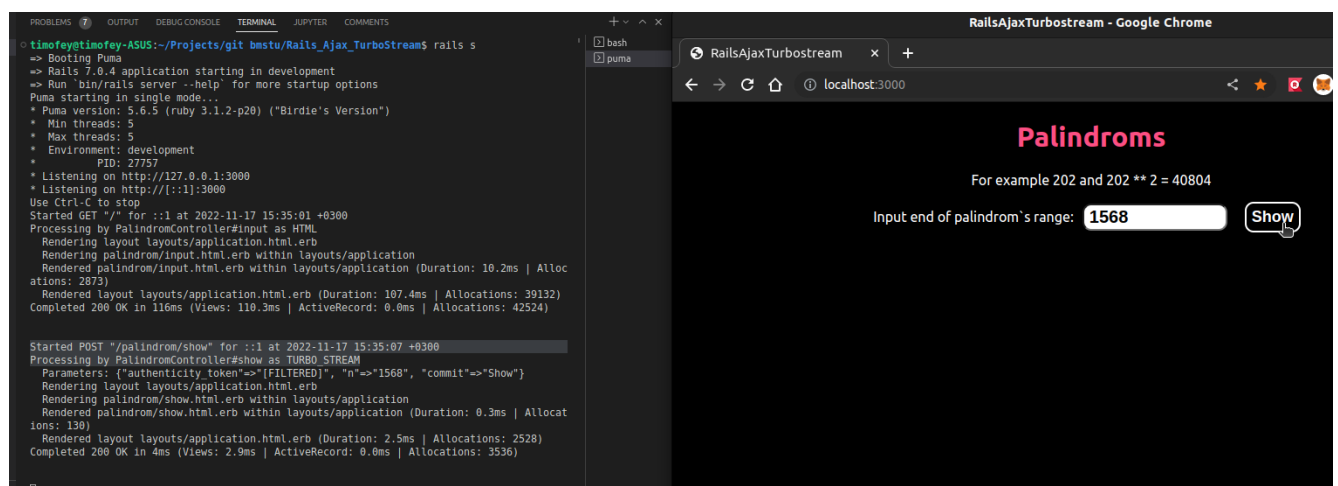
  private

  def palindroms_before(n)
    [*1..n].select { |el| el.to_s.reverse == el.to_s && (el**2).to_s.reverse == (el**2).to_s }
  end

  def require_params
    if params[:n].nil? || params[:n].empty?
      'Error: Empty params'
    elsif !params[:n].match(/^\\d+$/)
      "Error: Incorrect params'#{params[:n]}"
    end
  end
end

```

3) После чего запустим сервер и посмотрим что происходит при отправке формы:
rails s



Как видим, post запрос автоматически ожидает ответ в виде TURBO STREAM, именно так следует реализовать динамическую страницу в Rails 7.0.4

4) Создадим файл

app/views/palindrom/show.turbo_stream.erb

Который автоматически ищется по совпадению имен для отображения формата пост запроса. Добавим в его содержание следующий код:

```
<% if @error.nil?%>
```

```

<%= turbo_stream.replace 'result', partial: 'result' %>
<% else %>
<%= turbo_stream.replace 'result', partial: 'error' %>
<% end %>

```

Для отображения результата и ошибки создадим шаблоны

app/views/palindrom/_result.html.erb

```

<div id="result">
<h1 class="color-text center"> Palindroms from 1 to <%= params[:n] %> </h1>
<div class="table">
<div class="row">
<div class="col-1"> # </div>
<div class="col-5"> Number </div>
<div class="col-6"> Number ** 2</div>
</div>

<% @result.each_with_index do |el, idx| %>
<div class="row">
<div class="col-1"> <%= idx+1 %>.</div>
<div class="col-5"> <%= el %> </div>
<div class="col-6"> <%= el**2 %> </div>
</div>
<% end %>
</div>
</div>

```

app/views/palindrom/_error.html.erb

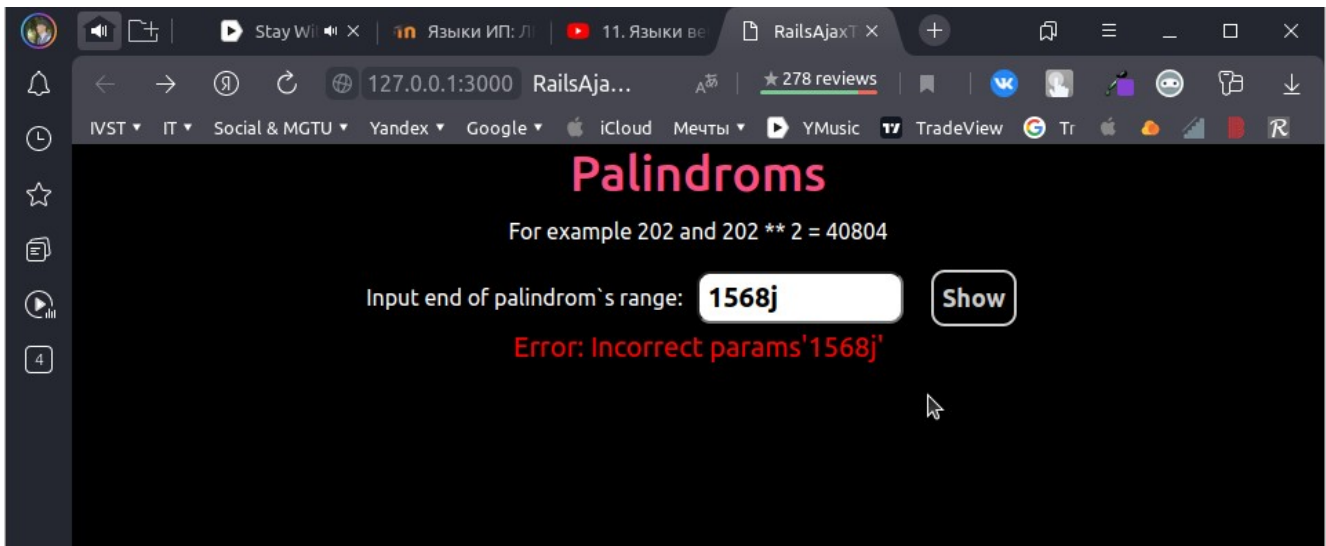
```

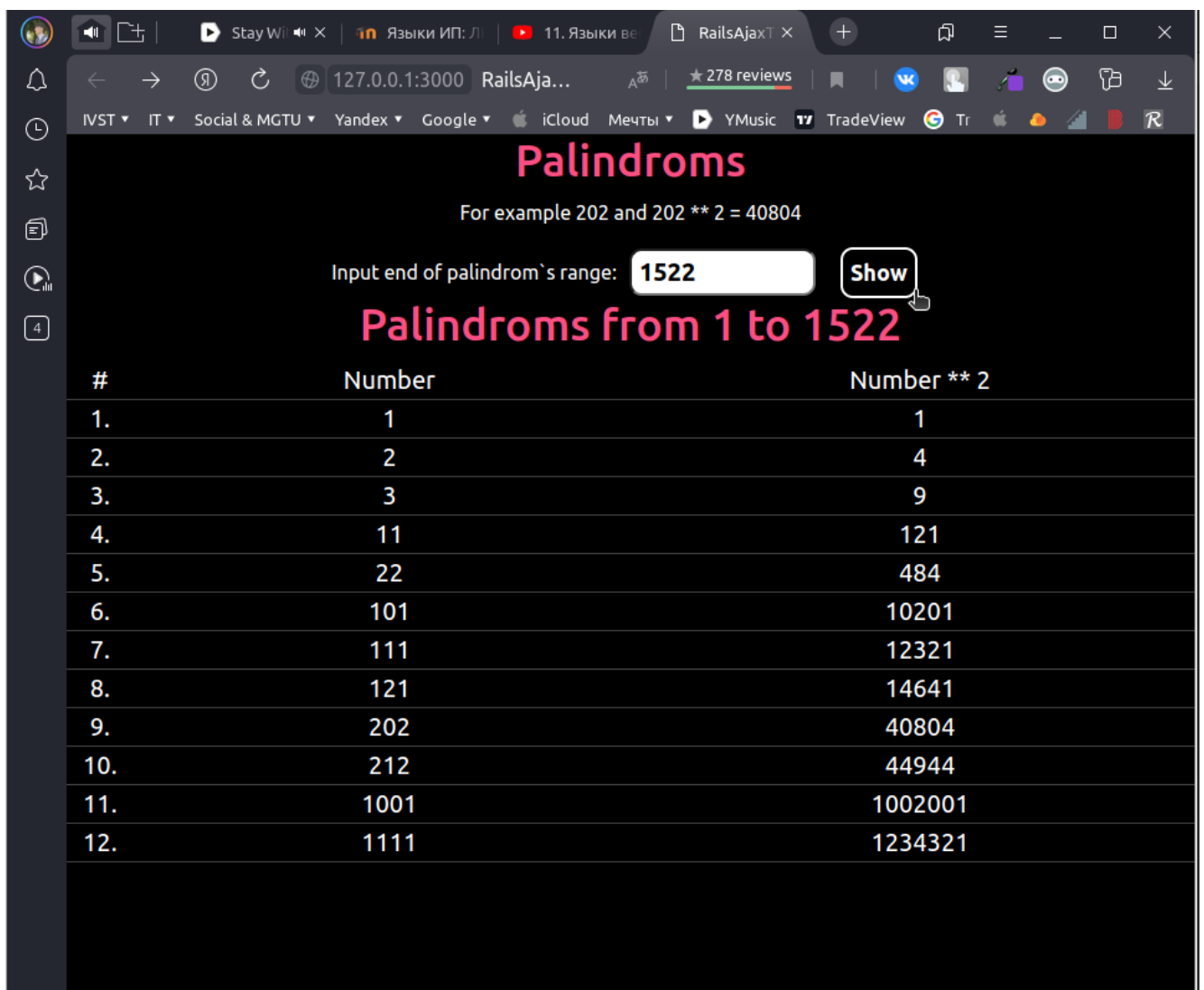
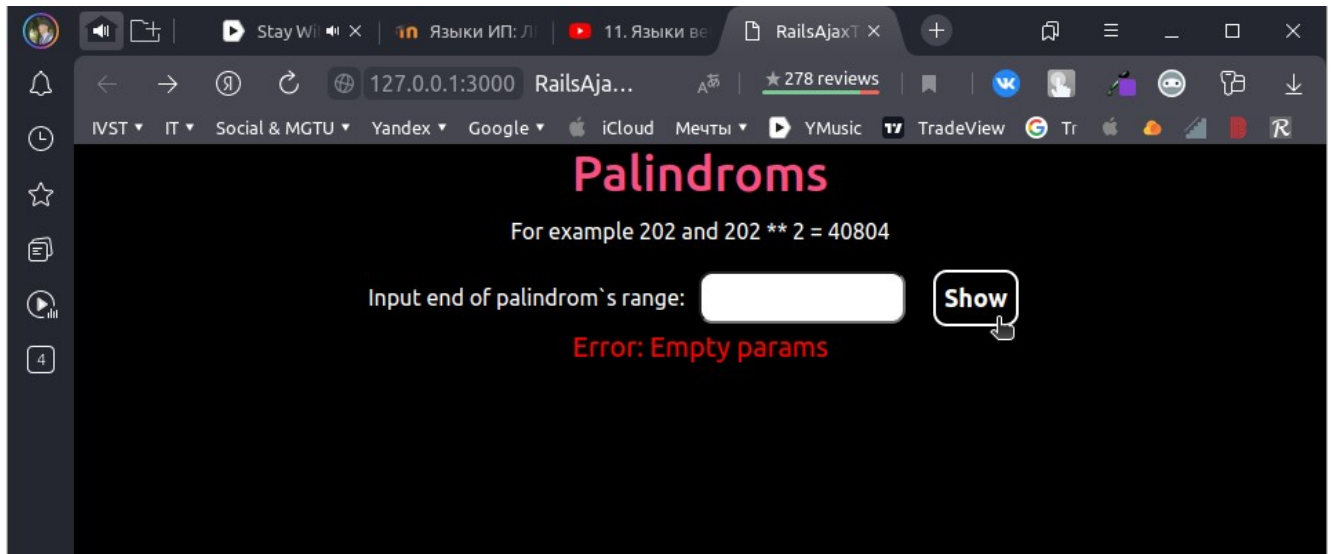
<div id="result">
<div class="error-text center"><%= @error %></div>
</div>

```

Они также ищутся приложением по совпадению имен, «_» перед именем входит в критерий поиска по умолчанию.

5) После чего запустим и протестируем приложение:





Тестирование

Для тестирования приложения будем использовать Rspec, Capybara, Selenium

1) Добавим в Gemfile зависимости

```
# Adding by me
group :test do
  gem 'capybara'
  gem 'rspec-rails'
  gem 'selenium-webdriver'
end
```

2) Тестирование

```
rails g rspec install

rails g rspec:controller palindrom
```

Добавление файлов тестов:
spec/requests/palindrom_spec.rb

```
# frozen_string_literal: true

require 'rails_helper'
require 'nokogiri'
require 'capybara/rails'
require 'selenium-webdriver'
require 'json'
include RSpec::Expectations

RSpec.describe 'Palindrom' do
  describe '/input' do
    context 'should return' do
      it 'http success' do
        get root_path
        expect(response).to have_http_status(:success)
      end

      it 'body contains form' do
        get root_path, params: { n: '145' }
        xml = Nokogiri::XML(response.body)
        form = xml.xpath('//*[@id="n"]')
        expect(form.attr('name').content).to eq('n')
      end
    end

    context 'When params are correct' do
      subject { post result_path, params: { n: 145 }, xhr: true }

      it 'should return http success' do
        subject
        expect(response).to have_http_status(:success)
      end

      it 'should calculate correct result' do
        subject
      end
    end
  end
end
```



```

expect(assigns(:result)).to eq([1, 2, 3, 11, 22, 101, 111, 121])
end

it 'should render partial _result' do
  subject
  expect(response).to render_template("_result")
end

it 'should render partial _error' do
  post result_path, xhr: true
  expect(response).to render_template("_error")
end
end
end

describe 'Capybara test case', type: :feature do
  context 'Capybara test case' do
    it 'body should contain header' do
      visit root_path
      expect(page).to have_content 'Palindroms'
    end
  end
end

describe 'Selenium WebDriver open root_path and' do
  before(:each) do
    @driver = Selenium::WebDriver.for :firefox
    @accept_next_alert = true
    @driver.manage.timeouts.implicit_wait = 30
    @verification_errors = []
    # @vars = {}
  end

  after(:each) do
    @driver.quit
    expect(@verification_errors.size).to be_zero
  end

  it 'send n = 1568l body should contain error' do
    @driver.get 'http://localhost:3000/'
    @driver.find_element(:id, 'n').click
    @driver.find_element(:id, 'n').clear
    @driver.find_element(:id, 'n').send_keys '1568l'
    @driver.find_element(:name, 'commit').click

    expect(@driver.find_element(:xpath, '//div[@id=\'result\']/div').text).to eq('Error: Incorrect
    params\'1568l\'')
  end

  it 'params are empty body should contain error' do
    @driver.get 'http://localhost:3000/'
    @driver.find_element(:id, 'n').click
  end
end

```

```

@driver.find_element(:id, 'n').clear
@driver.find_element(:name, 'commit').click

expect(@driver.find_element(:xpath, '//div[@id=\'result\']/div').text).to eq('Error: Empty
params')
end

it 'send default params body should contain ' do
@driver.get 'http://localhost:3000/'
@driver.find_element(:id, 'n').click
@driver.find_element(:name, 'commit').click

expect(@driver.find_element(:css, '.row:nth-child(13) > .col-6').text).to eq('1234321')
end

it 'n = 145 body should contain 14641' do
@driver.get('http://localhost:3000/')
@driver.find_element(:id, 'n').click
@driver.find_element(:id, 'n').clear
@driver.find_element(:id, 'n').send_keys('145')
@driver.find_element(:name, 'commit').click

expect(@driver.find_element(:xpath, '//div[@id=\'result\']/div/div[9]/div[3]').text).to
eq('14641')
end

it 'After errors body should contain result' do
@driver.get('http://localhost:3000/')
@driver.find_element(:id, 'n').click
@driver.find_element(:id, 'n').clear
@driver.find_element(:id, 'n').send_keys('122')
@driver.find_element(:name, 'commit').click

@driver.find_element(:id, 'n').click
@driver.find_element(:id, 'n').clear
@driver.find_element(:id, 'n').send_keys('')
@driver.find_element(:name, 'commit').click

@driver.find_element(:id, 'n').click
@driver.find_element(:id, 'n').clear
@driver.find_element(:id, 'n').send_keys('s')
@driver.find_element(:name, 'commit').click

@driver.find_element(:id, 'n').click
@driver.find_element(:id, 'n').clear
@driver.find_element(:id, 'n').send_keys('500')
@driver.find_element(:name, 'commit').click

expect(@driver.find_element(:css, '.row:nth-child(11) > .col-6').text).to eq('44944')
end
end
end

```

The screenshot shows a web browser window displaying a Rails application titled "Palindroms". The application has a form where the user can input the end of a palindrome's range. The input field contains "122", and a "Show" button is next to it. Below the form, a table lists palindromes from 1 to 122. The table has three columns: "#", "Number", and "Number ** 2".

#	Number	Number ** 2
1.	1	1
2.	2	4
3.	3	9
4.	11	121
5.	22	484
6.	101	10201
7.	111	12321
8.	121	14641

Below the browser window, a code editor shows the terminal output of the application. The terminal displays the command `rspec` and its output, which includes the number of examples and failures.

```
app bin config config.ru db Gemfile Gemfile.lock lib log public Rakefile README.md Report_lab9.odt spec storage tmp vendor
timofey@timofey-ASUS:~/Projects/git/bmstu/lab9/Rails_Ajax_TurboStream$ kazyac
kazyac: command not found
timofey@timofey-ASUS:~/Projects/git/bmstu/lab9/Rails_Ajax_TurboStream$ rspec
.....DEBUGGER: Attaching after process 76188 fork to child process 76239
DEBUGGER[/home/timofey/.rbenv/versions/3.1.2/bin/rspec#76762]: Attaching after process 76188 fork to child process 76762
DEBUGGER[/home/timofey/.rbenv/versions/3.1.2/bin/rspec#77228]: Attaching after process 76188 fork to child process 77228
DEBUGGER[/home/timofey/.rbenv/versions/3.1.2/bin/rspec#77686]: Attaching after process 76188 fork to child process 77686
DEBUGGER[/home/timofey/.rbenv/versions/3.1.2/bin/rspec#78145]: Attaching after process 76188 fork to child process 78145

Finished in 24.29 seconds (files took 0.83801 seconds to load)
12 examples, 0 failures

timofey@timofey-ASUS:~/Projects/git/bmstu/lab9/Rails_Ajax_TurboStream$
```

Тесты пройдены успешно. Во время них использовал Selenium, capybara, Selenium IDE, Koton studio

Вывод: Научился создавать Rails-приложение, создавать формы, передавать данные от клиента к серверу.