



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 № 8

Название: Создание каркаса RoR-приложения. Настройка базовых параметров. Создание форм. Передача данных от клиента к серверу.

Дисциплина: Языки интернет программирования

Студент

ИУ6-31Б
(Группа)

(Подпись, дата)

Т.Е. Старжевский
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Д. В. Малахов
(И.О. Фамилия)

Текст задания: Необходимо разработать Rails-приложение, сформировать контроллер, наполнить его соответствующей логикой и сформировать представления. Результаты проверить функциональным тестом контроллера.

Написать программу, определяющую, сколько из чисел в диапазоне от 0 до n (n вводится с клавиатуры), сами являясь палиндромами, при возведении в квадрат, образуют палиндромы. Например: $222 \rightarrow 484$. Вывести на печать числа, квадраты и их количество

Работа осуществляется на версиях Rails 7.0.4 и Ruby 3.0.2

```
timofey@timofey-ASUS:~/Desktop/RAILS_BMSTU_LAB8$ rails -v
Rails 7.0.4
timofey@timofey-ASUS:~/Desktop/RAILS_BMSTU_LAB8$ ruby -v
ruby 3.0.2p107 (2021-07-07 revision 0db68f0233) [x86_64-linux]
timofey@timofey-ASUS:~/Desktop/RAILS_BMSTU_LAB8$
```

1) Создаем Rails-приложение:

```
rails new RAILS_BMSTU_LAB8
```

2) Создаем генератор:

```
rails g controller palindroms input show
```

```
timofey@timofey-ASUS:~/Desktop/RAILS_BMSTU_LAB8$ rails g controller palindroms input show
create  app/controllers/palindroms_controller.rb
route   get 'palindroms/input'
        get 'palindroms/show'
invoke  erb
create  app/views/palindroms
create  app/views/palindroms/input.html.erb
create  app/views/palindroms/show.html.erb
invoke  test_unit
create  test/controllers/palindroms_controller_test.rb
invoke  helper
create  app/helpers/palindroms_helper.rb
invoke  test_unit
timofey@timofey-ASUS:~/Desktop/RAILS_BMSTU_LAB8$
```

3) Изменим представления:

app/views/palindroms/input.html.erb:

```
<h1 class="color-text center">Palindroms</h1>
<p class="center">For example 202 and 202 ** 2 = 40804</p>
<%= form_with url: "/palindroms/show", class: "center", method: :get do |form|
%>
  <%= form.label :n, "Input end of palindrom`s range:", class: "center" %>
  <%= form.text_field :n, class: "text-field", value: 5262 %>
  <%= form.submit "Show", class:"button" %>
<% end %>
<% unless flash[:error].nil?%>
<div class="error-text center"> <%= flash[:error] %> </div>
```

```
<% end %>
```

app/views/palindroms/show.html.erb

```
<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>
```

4) Подключим bootstrap в шаблон:

app/views/layouts/application.html.erb

```
<!DOCTYPE html>
<html>
  <head>
    <title>AirPalindroms</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-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeu0xjzrPF/
      et3URy9Bv1WTRi" crossorigin="anonymous">
    <%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>
    <%= javascript_importmap_tags %>
  </head>

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

5) Добавим стили:

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.
 *
 *= require_tree .
 *= require_self
 */
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;
}

```

6) Изменим контроллер:

app/controllers/palindroms_controller.rb

```

# frozen_string_literal: true

# Documentation
class PalindromsController < ApplicationController
  before_action :require_params, only: [:show]

  def input; end

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

  private

  # rubocop:disable Naming/MethodParameterName
  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
# rubocop:disable Naming/MethodParameterName

# rubocop:disable Metrics/AbcSize
def require_params
  if params[:n].nil? || params[:n].empty?
    flash[:error] = 'Error: Empty params'
    redirect_to palindroms_input_path
  elsif !params[:n].match(/^\d+$/)
    flash[:error] = "Error: Incorrect params '#{params[:n]}'"
    redirect_to palindroms_input_path
  end
end
end
# rubocop:enable Metrics/AbcSize
end

```

Сразу проверим его командой rubocop, предварительно внутри него отключив проверку на длину имени:

```
rubocop:disable Naming/MethodParameterName
```

И на количество ветвей:

```
rubocop:disable Metrics/AbcSize
```

Потом их включаем командой enable. Результат:

```

application_controller.rb concerns palindroms_controller.rb
• timofey@timofey-ASUS:~/Desktop/RAILS_BMSTU_LAB8/app/controllers$ rubocop palindroms_controller.rb
Inspecting 1 file
.

1 file inspected, no offenses detected
○ timofey@timofey-ASUS:~/Desktop/RAILS_BMSTU_LAB8/app/controllers$

```

7) Изменим корневой путь:

config/routes.rb

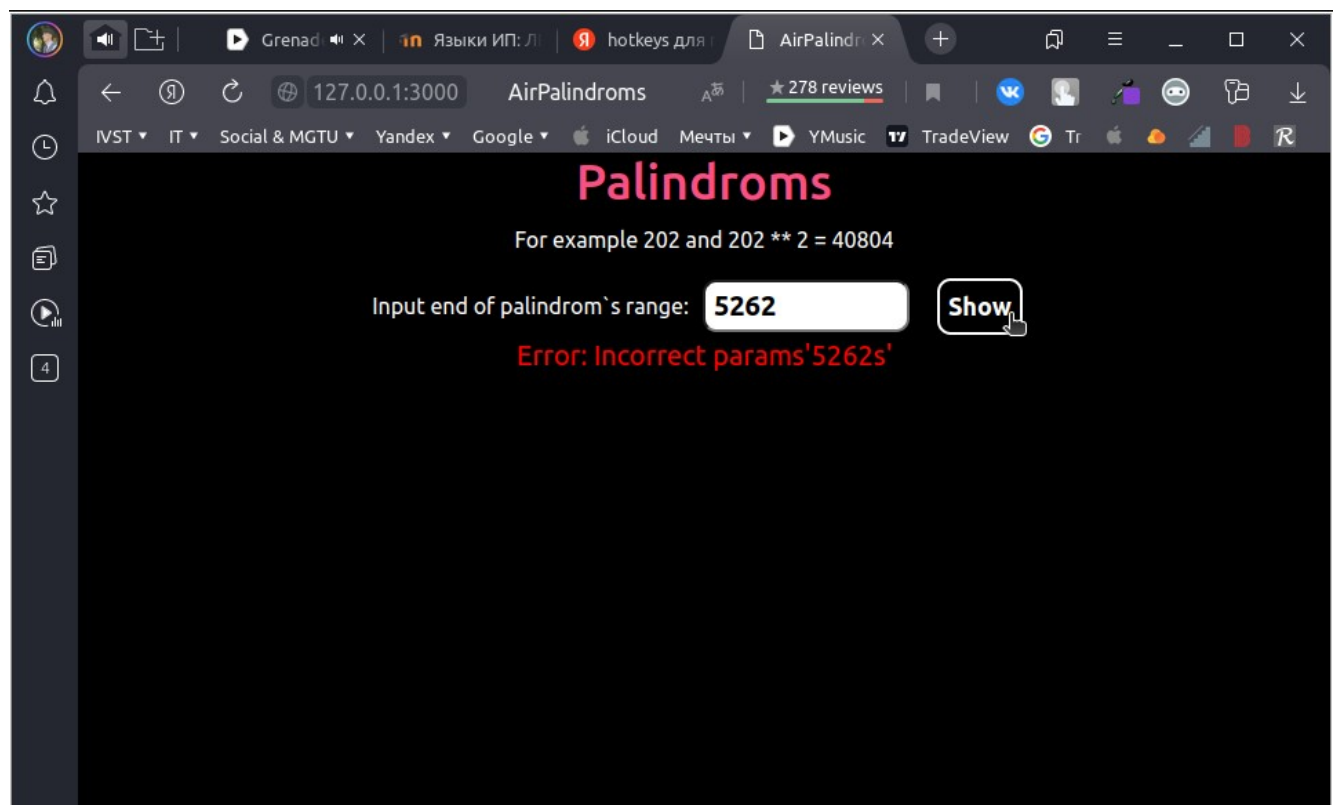
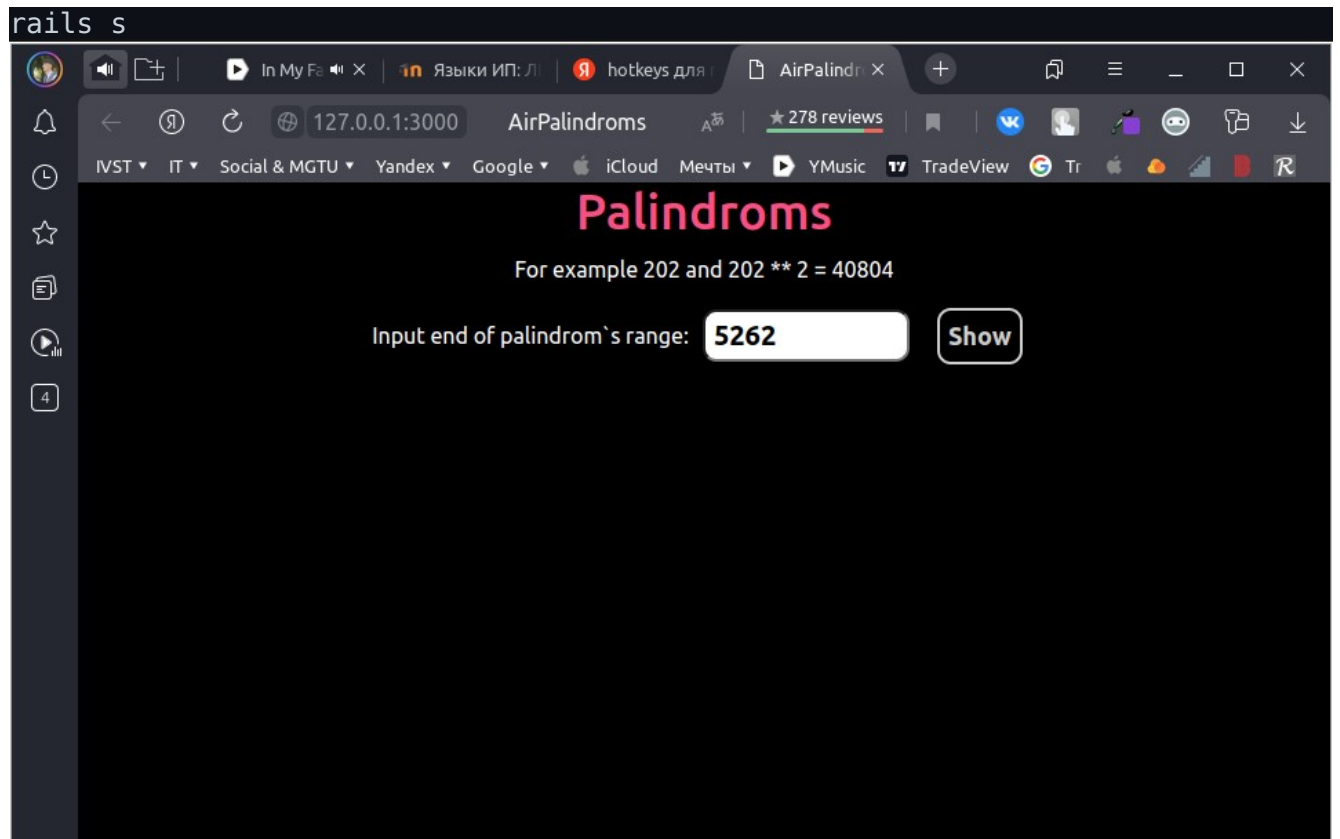
```

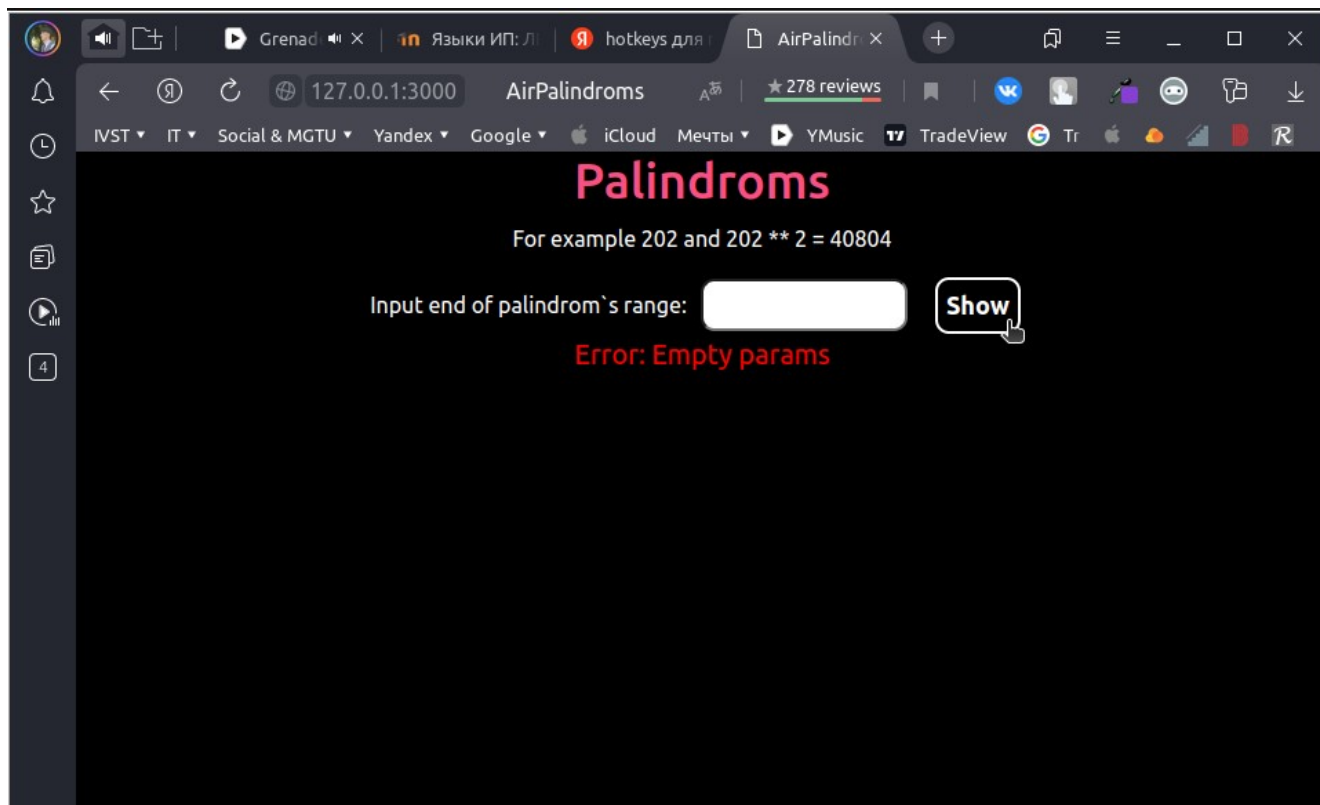
Rails.application.routes.draw do
  root 'palindroms#input'
  get 'palindroms/input', to: 'palindroms#input'
  get 'palindroms/show'
  # Define your application routes per the DSL in
  https://guides.rubyonrails.org/routing.html

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

```

8) Запустим приложение, чтобы проверить логику:





При корректных параметрах:

The screenshot shows the same web browser window, but the page now displays a list of palindromes. The title is "Palindroms from 1 to 5262". Below the title is a table with three columns: "#", "Number", and "Number ** 2". The table contains 13 rows of data, showing the first 13 palindromes and their squares.

#	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
9.	202	40804
10.	212	44944
11.	1001	1002001
12.	1111	1234321
13.	2002	4008004

Написание тестов

1) Будем пользоваться Rspec и парсер Nokogiri, для этого добавим в Gemfile их установку:

```
group :test do
  # Use system testing [https://guides.rubyonrails.org/testing.html#system-testing]
  # gem "capybara"
  # gem "selenium-webdriver"
  # gem "webdrivers"
  # Adding by me
  gem 'rspec-rails'
  gem 'nokogiri'
end
```

2) Далее выполним последовательно команды:

```
rails g rspec:install
rails g rspec:controller palindroms
```

3) Опишем тесты:

spec/requests/palindroms_spec.rb

```
require 'rails_helper'
require 'nokogiri'

RSpec.describe 'Palindroms', type: :request do
  describe 'path' do
    context '/input' do
      it 'should return http success' do
        get root_path
        expect(response).to have_http_status(:success)
      end
    end

    context '/show' do
      it 'should return success if input correct params' do
        get palindroms_show_path, params: { n: 25 }
        expect(response).to have_http_status(:success)
      end

      context 'shouldn't redirect and return http Error if' do
        it 'redirecting was without params' do
          get palindroms_show_path
          expect(response).to have_http_status(302)
        end

        it 'redirecting was with empty params' do
          get palindroms_show_path, params: {}
        end
      end
    end
  end
end
```

```

expect(response).to have_http_status(302)
end

it 'input \'25s\'' do
  get palindroms_show_path, params: { n: '25s' }
  expect(response).to have_http_status(302)
end

it 'input \'h25\'' do
  get palindroms_show_path, params: { n: 'h25' }
  expect(response).to have_http_status(302)
end

it 'input words' do
  get palindroms_show_path, params: { n: 'khxuizv' }
  expect(response).to have_http_status(302)
end
end

context 'should return body contains' do
  it 'all palindroms from 1 to 145' do
    get palindroms_show_path, params: { n: '145' }
    html = Nokogiri::HTML(response.body)
    expects = [' 1 ', ' 4 ', ' 9 ', ' 121 ', ' 484 ', ' 10201 ', ' 12321 ', ' 14641 ']
    expects.each_with_index do |el, idx|
      div = html.search('div')[8 + idx * 4].text
      expect(div).to eq(el)
    end
  end
end
end
end
end
end
end
end

```

Тестируем отсутствие параметров при передаче, передачу пустых или некорректных параметров, наличие необходимых элементов на странице при корректных параметрах.

4) Запустим тесты командой rspec

```

• timofey@timofey-ASUS:~/Desktop/RAILS_BMSTU_LAB8$ rspec
.....

Finished in 0.11389 seconds (files took 0.6976 seconds to load)
8 examples, 0 failures

• timofey@timofey-ASUS:~/Desktop/RAILS_BMSTU_LAB8$

```

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