



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 № 1 1

Название:

Добавление модели. ORM. Разработка БД, подключение,
хранение и поиск данных.

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)

Москва, 2022

Текст задания:

Модифицировать код ЛР 8 таким образом, чтобы запросы, которые были ранее выполнены, сохранялись в БД и при следующем запросе не требовали повторных вычислений.

- Сформировать модель в соответствии с потребностями хранения данных. Входные параметры являются ключами, по которым извлекается результат.
- Выполнить создание БД и миграцию соответствующими запросами rake.
- Написать тест на добавление и поиск данных с помощью модели. Проверить выполнение теста.
- Модифицировать код приложения таким образом, чтобы результат вычислений преобразовывался в строковый или бинарный формат (на выбор: json, xml, и пр.). Проверить через отладочную печать в консоль, что преобразование выполняется корректно.
- Вставить код для сохранения данных в БД и запрос на поиск предыдущего результата вычислений.
- Добавить действие в контроллер, позволяющее определить, что хранится в БД через сериализацию в XML.
- Проверить, что при выполнении запроса, данные добавляются в БД.
- При помощи консоли сообщений Puma/Webrick определить, производится ли поиск результата предыдущего запроса в БД и не повторяются ли одни и те же вычисления.
- Модифицировать модель таким образом, чтобы добавление записей с одинаковыми параметрами было невозможно.
- Реализовать тест модели, проверяющий невозможность повторного добавления одних и тех же результатов вычислений.
- Реализовать функциональный тест, проверяющий, что результаты вычислений различны при различных входных параметрах.
- Проверить маршруты приложения с помощью rake routes и убрать лишние. Обеспечить доступ при обращении по адресу /.

Последовательность выполненных команд:

```
rails new Rails_smart_lab8 -T
rails g controller palindrom input show
rails g model palindrom upper:integer result:text
```

Добавим правила добавления в модель:

Rails_smart_lab8/app/models/palindrom.rb

```

class Palindrom < ApplicationRecord
  validates :upper, {
    presence: { message: 'Error: Empty params'},
    numericality: {
      only_integer: true,
      greater_than: 0,
      message: "Error: Incorrect params '%{value}'"
    }
  }

  before_save :set_result

  def decoded_result
    ActiveSupport::JSON.decode(result)
  end

  private

  def set_result
    self.result = ActiveSupport::JSON.encode(palindroms_before)
  end

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

```

Логику контроллера и вьюх:

Rails_smart_lab8/app/controllers/palindrom_controller.rb

```

class PalindromController < ApplicationController
  def input; end

  def show
    if (query = Palindrom.find_by(upper: params[:n]))
      @notice = 'FROM DB'
      @result = query
    else
      palindrom = Palindrom.new(upper: params[:n])
      if palindrom.save
        @notice = 'SAVE TO DB'
        @result = palindrom
      else
        flash[:error] = palindrom.errors.messages[:upper]
        redirect_to root_path
      end
    end
  end

  def results
    arr = []
    Palindrom.all.each { |el| arr << { upper: el.upper, result: el.result } }
  end
end

```

```
render xml: arr
end
end
```

Rails_smart_lab8/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", method: :get do |form| %>
  <%= form.label :n, "Input end of palindrom`s range:", class: "center" %>
  <%= form.number_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 %>
```

Rails_smart_lab8/app/views/palindrom/show.html.erb

```
<h1 class="color-text center"> Palindroms from 1 to <%= @result.upper %> </h1>
<div class="table">
<div class="notice"><%= @notice %></div>
<div class="row">
<div class="col-1"> # </div>
<div class="col-5"> Number </div>
<div class="col-6"> Number ** 2</div>
</div>

<% @result.decoded_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>
```

Rails_smart_lab8/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-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/
et3URy9Bv1WTRi" crossorigin="anonymous">
<%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>
```

```
<%= javascript_importmap_tags %>
</head>

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

Rails_smart_lab8/config/routes.rb

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

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

Rails_smart_lab8/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;
}

.notice {
color: rgb(1, 46, 247);
}
```

Добавим миграцией уникальный индекс:

```
rails db:migrate
rails g migration AddUniqIndexForPalindrom
```

Rails smart_lab8/db/migrate/20221203104959_add_uniq_index_for_palindrom.rb

```
class AddUniqIndexForPalindrom < ActiveRecord::Migration[7.0]
```

```
  def change
```

```
    add_index :palindroms, :upper, unique: true
```

```
  end
```

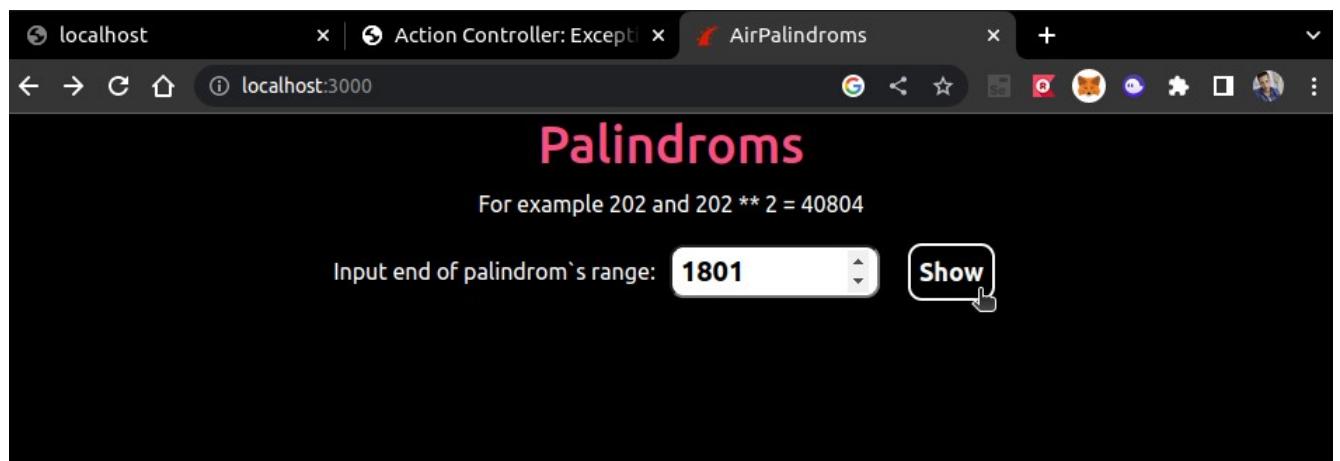
```
end
```

```
rails db:migrate
```

Запустим приложение:

```
rails s
```

Введем отсутствующее в базе данных значение:



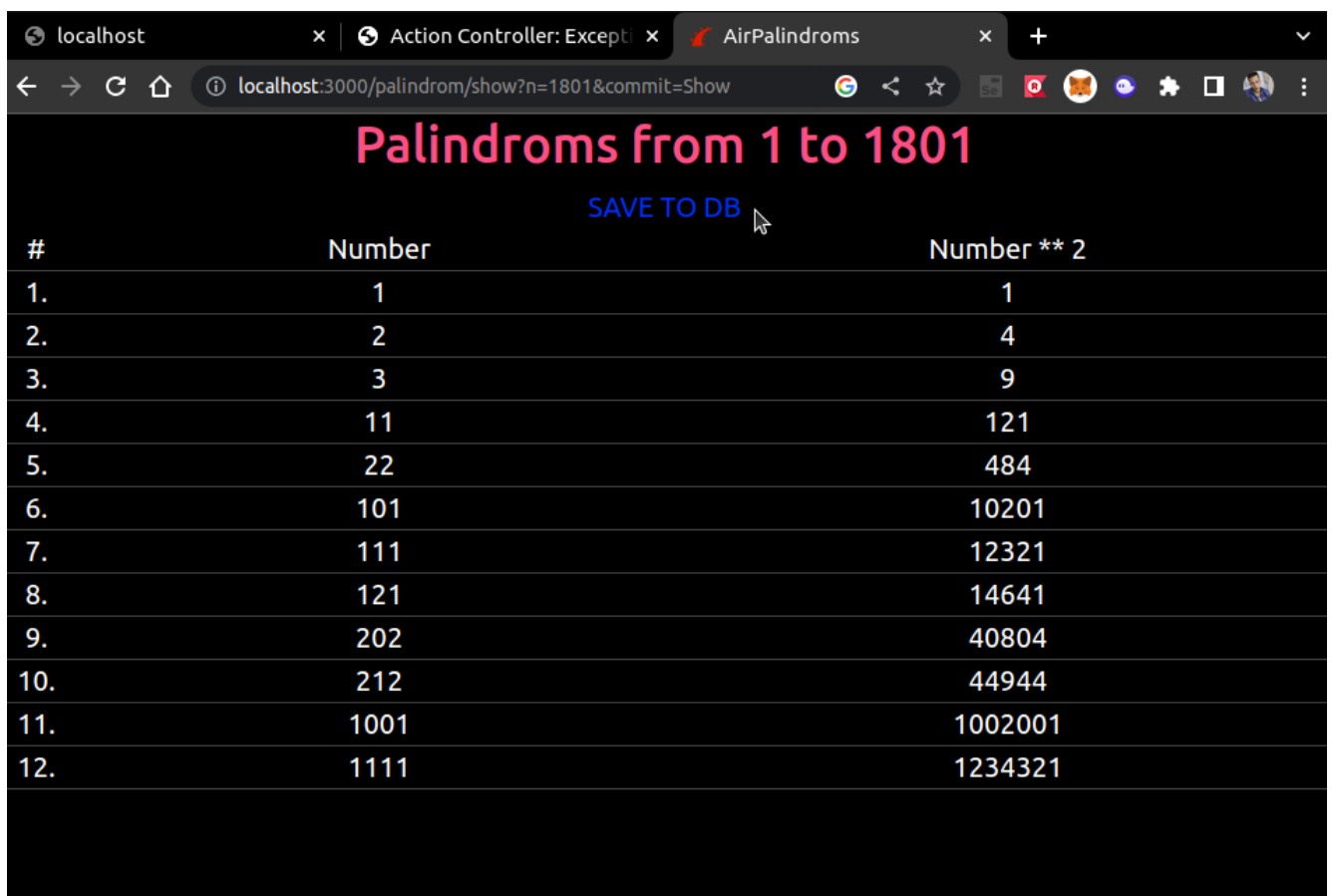
localhost x | Action Controller: Excepti x | AirPalindroms x +

localhost:3000

Palindroms

For example 202 and 202 ** 2 = 40804

Input end of palindrom`s range: Show



localhost x | Action Controller: Excepti x | AirPalindroms x +

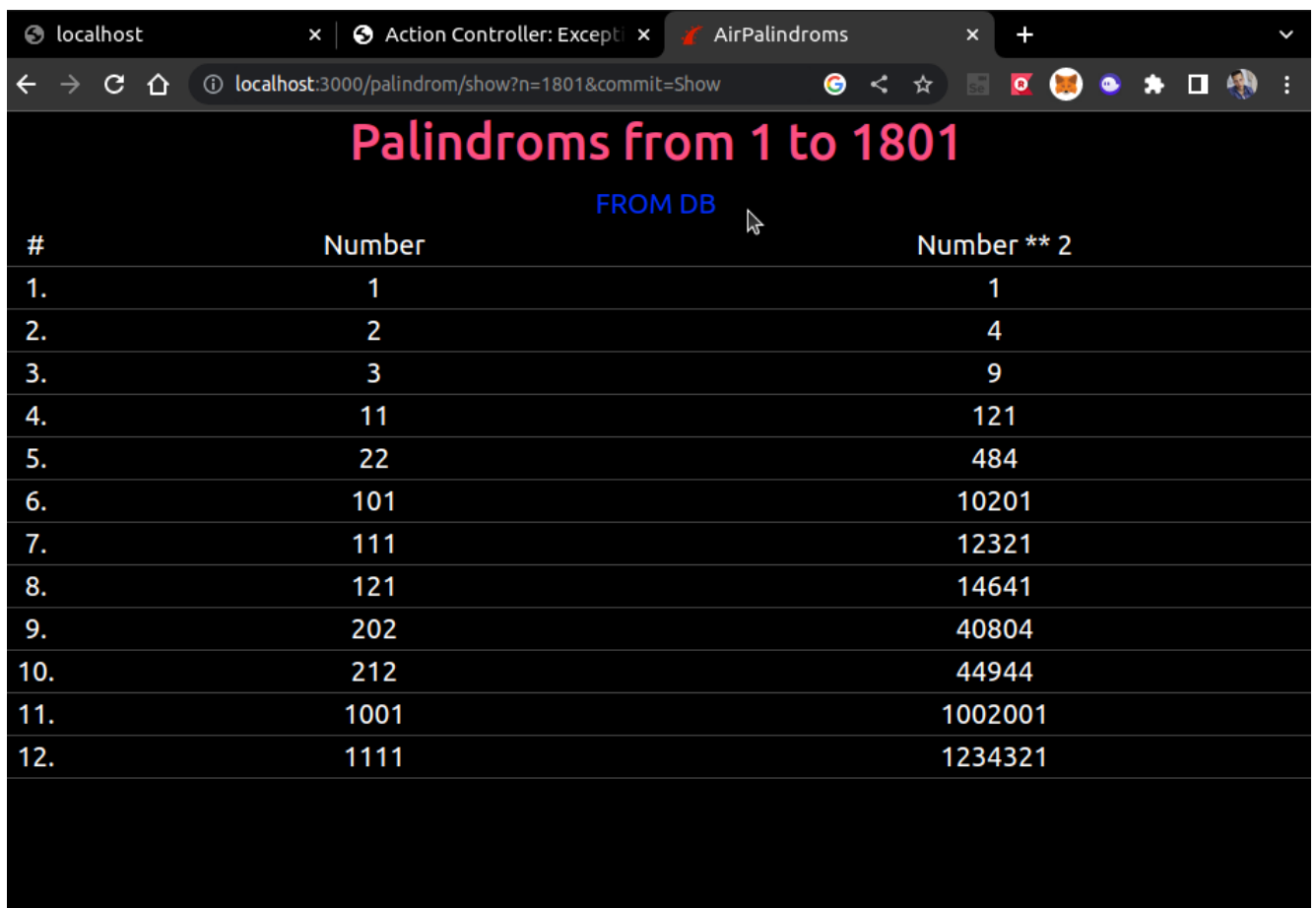
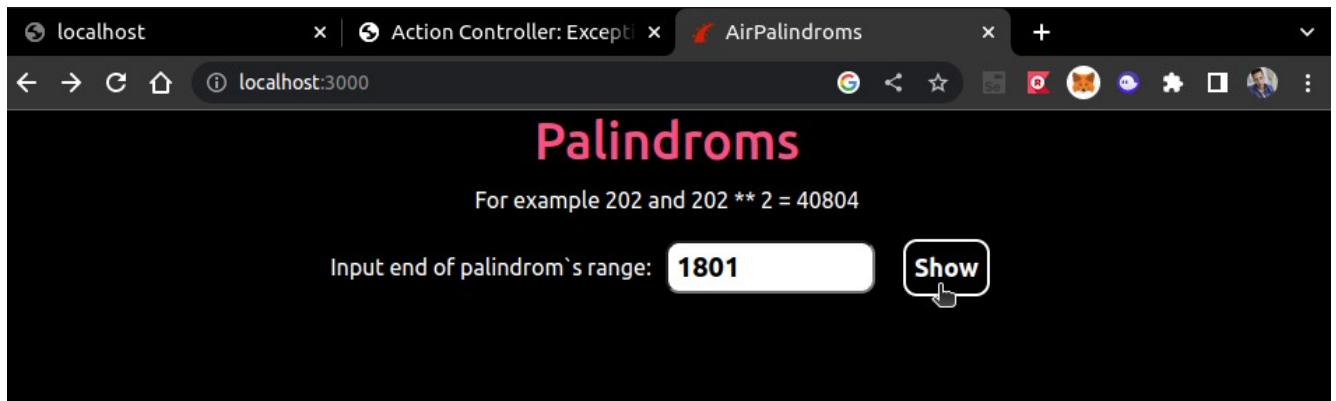
localhost:3000/palindrom/show?n=1801&commit=Show

Palindroms from 1 to 1801

[SAVE TO DB](#)

#	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

Видим вывод нотиса SAVE TO DB, повторим операцию с тем же значением, только теперь оно уже есть в базе данных:



#	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

Как видим, в ветке контроллера цикл явно пошел в ту сторону, в которой нотис FROM DB, из чего мы понимаем, что заново нагрузка на сервер по расчету значения не шла, и результат был получен из базы данных запросом.

Тестирование:

Добавим в Gemfile зависимости:

```
group :test do
  gem 'rspec-rails'
end
```



```
rails g rspec install
```

```
rails g rspec:model palindrom
```

Rails_smart_lab8/spec/models/palindrom_spec.rb

```
require 'rails_helper'

RSpec.describe Palindrom, type: :model do
  describe 'If adding with similar params' do
    before do
      Palindrom.destroy_all
      Palindrom.create!(upper: 17)
    end
    it 'should return error if value isn`t unique' do
      expect { Palindrom.create!(upper: 17) }.to raise_error(ActiveRecord::RecordNotUnique)
    end
    context 'should return correct result' do
      it 'when upper 17' do
        expect(Palindrom.create!(upper: 12).result).to eq('[1,2,3,11]')
      end
      it 'when upper 35' do
        expect(Palindrom.create!(upper: 35).result).to eq('[1,2,3,11,22]')
      end
    end
  end
end
end
end
```

Запустим тесты:

```
rspec
```

```
timofey@timofey-ASUS:~/Projects/git bmstu/lab11/Rails_smart_lab8$ rspec
...
Finished in 0.03509 seconds (files took 0.80352 seconds to load)
3 examples, 0 failures
```

Вывод: Научился работать с моделью Rails приложения, добавлять валидацию данных для нее и создавать миграции по добавлению различных индексов в модель, а также писать тесты для нее на RSpec.