



НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

по лабораторной работе № 1 1

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

Дисциплина: Интернет-Программирование

Преподаватель _____

(Подпись, дата) (И.О. Фамилия)

Москва, 2020

Задание

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

Исходный код

```
#number_controller.rb
# frozen_string_literal: true

# class controller
class NumberController < ApplicationController
  def input; end

  def output
    @result = PerNum.create_with_output(params[:arr])
  end
end
```

```

#per_nem.rb /model/
class PerNum < ApplicationRecord
  validates :arr, presence: true, uniqueness: true
  validates :output, presence: true

  class << self
    def create_with_output(str)
      if where(arr: str).count().zero?
        create(arr: str, output: out(str))
      end
      @result = where(arr: str).first
    end

    private

    def out(str)
      if str == "" || str.nil?
        'input nil'
      elsif !str.nil? && str.strip.split.map(&:to_i).join(' ') != str.strip
        "don't numbers"
      else
        get_perfect_numbers(convert_input_to_numbers(str))
      end
    end

    def perfect_number?(num)
      s = (1..(num / 2)).select { |a| (num % a).zero? }
      num == s.compact.inject(:+)
    end

    def get_perfect_numbers(arr)
      out = []
      inp = []
      arr = arr.drop_while { |y| !perfect_number?(y) }
      arr.each do |x|
        if !perfect_number?(x)
          out.append(inp) if inp.any?
          inp = Array.new
        else
          inp.append(x)
        end
      end
      out.append(inp) if inp.size.positive?
      out
    end

    def convert_input_to_numbers(line)
      line.split.map(&:to_i)
    end
  end
end

```

```

#input.html.erb
<%= form_tag("/number/output", :method => "get") do %>
  <%= label_tag("search perfect number:") %>
  <%= text_field_tag(:arr, value = "6 28 3 28", required: true) %>
  <br/><br/>
  <%= submit_tag("Get result") %>
<% end %>

#output.html.erb
<h1>Number#output</h1>
<p>Find me in app/views/number/output.html.erb</p>
<table>
  <p id = "resut">
    <%if @result.output !~ /\d/ %>
      <%= @result.output%>
    <%else%>
      <% res = JSON.parse(@result.output)%>
      <tr>
        <td style="border: 1px solid grey;">array</td>
        <td style="border: 1px solid grey;">length</td>
      </tr>
      <% res.each do |x| %>
        <tr>
          <td style="border: 1px solid grey;"><%= x %></td>
          <td style="border: 1px solid grey;"><%= x.size %></td>
        </tr>
      <% end %>
      <p>Max length <%=value = res.map { |x| x.size }.max %></p>
      <p>Array with max length <%= res.find { |x| x.size == value } %> </p>
    <% end %>
  </p>
</table>

#routes.rb
Rails.application.routes.draw do
  root 'number#input'
  get 'number/output'
  get 'number/output'
  post 'number/output'
  get 'number/output'
end

#db
class CreatePerNums < ActiveRecord::Migration[6.0]
  def change
    create_table :per_nums do |t|
      t.string :arr
      t.string :output
      t.timestamps
    end
  end
end
end

```

```

#number_controller_test.rb
require 'test_helper'
require 'json'

class NumberControllerTest < ActionDispatch::IntegrationTest
  test "should get output" do
    get 'http://127.0.0.1:3000/number/output?arr=6'
    assert_response :success
  end

  test 'should get random text' do
    get 'http://127.0.0.1:3000/number/output?arr=ssfsd'
    assert_equal assigns[:result].output, 'don\'t numbers'
  end

  test 'should get numbers' do
    get number_output_url, params: { arr: '6 28 2 6' }
    assert_equal assigns[:result].output, "[[6, 28], [6]]"
  end
end

#per_nem_test.rb
require 'test_helper'

class PerNumTest < ActiveSupport::TestCase
  test 'should save data into db' do
    instance = PerNum.new(arr: 10)
    assert_not instance.save
  end

  test 'should get data from db' do
    assert PerNum.create_with_output('asd').valid?
  end

  test 'should get 1 data from db' do
    assert PerNum.create_with_output('6 1 2').valid?
  end
end

```

Number#output

Find me in app/views/number/output.html.erb

don't numbers

Number#output

Find me in app/views/number/output.html.erb

Max length 2

Array with max length [6, 28]

| array | length |
|---------|--------|
| [6, 28] | 2 |
| [28] | 1 |

```
~/Desktop/mgtu/mgtu_3_sem/IP/lab11/lab11(master*) » rake
Run options: --seed 23130

# Running:

.....

Finished in 4.243433s, 1.4139 runs/s, 1.4139 assertions/s.
6 runs, 6 assertions, 0 failures, 0 errors, 0 skips
-----
~/Desktop/mgtu/mgtu_3_sem/IP/lab11/lab11(master*) »
```

Вывод

В ходе лабораторной работы была решена 8 лабораторная работа с использованием базы данных и моделей.