| Gerb-BMSTU_01 | **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**

**ОТЧЕТ**

| **по лабораторной работе №** | 11 |
| --- | --- |



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

**Дисциплина:** Языки Интернет-программирования

| Студент | ИУ6-33Б |  | 03.12.2021 | Хоанг Т.Ф. |
| --- | --- | --- | --- | --- |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2021

## ЛР 11. Добавление модели. ORM. Разработка БД, подключение, хранение и поиск данных.

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 /.

**routes.rb**

Rails.application.routes.draw do

# For details on the DSL available within this file, see https://guides.rubyonrails.org/routing.html

root 'friend\_num#home'

post '/result', to: 'friend\_num#result', as: 'friend\_num'

get '/database', to: 'friend\_num#database'

end

**friend\_num\_controller.rb**

# frozen\_string\_literal: true

# Base class controller for friend\_num

class FriendNumController < ApplicationController

attr\_accessor :res, :num, :db

def home; end;

def sum\_divisors(arg)

sum = 0

(1...arg).each do |div|

sum += div if arg % div == 0

end

sum

end

def check

arr = Array.new(0)

sum = Array.new(num)

(1..num).each {|index| sum[index] = sum\_divisors(index)}

(1..num - 1).each do |num1|

(num1..num).each do |num2|

arr.append(" #{num1} and #{num2}") if num1 != num2 && num1 == sum[num2] && sum[num1] == num2

end

end

arr

end

def result

@num = params[:num].to\_i

if !FriendNum.exists?(num: num)

str = check.join(';')

fr\_num = FriendNum.new(num: num, str\_out: str)

fr\_num.save

end

@res = FriendNum.where('num = ?', @num)

end

def database

@db = FriendNum.all

end

end

**home.html.erb**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

<title></title>

</head>

<body>

<h1>FRIEND NUMBERS</h1>

<p>Two numbers are called friend if each of them is equal to the sum of all divisors of another, except for this number itself. Find every friend numbers from 1 to n</p>

<%= form\_with(url: friend\_num\_path, method: "post", local: true) do |form|%>

<%= form.label :num, "Enter n: " %>

<%= form.text\_field :num %>

<div class="button">

<%= form.submit "Get result" %>

</div>

<% end %>

</body>

</html>

**result.html.erb**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

<title>Result</title>

</head>

<body>

<h1>RESULT TABLE - Friend numbers</h1>

<table class="table table-bordered">

<tr>

<th>Number n</th>

<th>Friend Number</th>

</tr>

<% @res.each do |x|%>

<tr>

<td><%= x.num %></td>

<td><%= x.str\_out %></td>

</tr>

<% end %>

</table>

</body>

</html>

**database.html.erb**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

<title>Result</title>

</head>

<body>

<h1>DATABASE TABLE</h1>

<table class="table table-bordered">

<tr>

<th>Number n</th>

<th>Friend Number</th>

<th>Time 1</th>

<th>Time 2</th>

</tr>

<% @db.each do |x|%>

<tr>

<td><%= x.num %></td>

<td><%= x.str\_out %></td>

<td><%= x.created\_at %></td>

<td><%= x.updated\_at %></td>

</tr>

<% end %>

</table>

</body>

</html>

**model/friend\_num.rb**

class FriendNum < ApplicationRecord

validates :num, presence: true

validates :str\_out, presence: true

end

**db/migrate/20211203104933\_create\_friend\_num.rb**

class CreateFriendNum < ActiveRecord::Migration[6.1]

def change

create\_table :friend\_nums do |t|

t.integer :num

t.string :str\_out

t.timestamps

end

end

end

**schema.rb**

# This file is auto-generated from the current state of the database. Instead

# of editing this file, please use the migrations feature of Active Record to

# incrementally modify your database, and then regenerate this schema definition.

#

# This file is the source Rails uses to define your schema when running `bin/rails

# db:schema:load`. When creating a new database, `bin/rails db:schema:load` tends to

# be faster and is potentially less error prone than running all of your

# migrations from scratch. Old migrations may fail to apply correctly if those

# migrations use external dependencies or application code.

#

# It's strongly recommended that you check this file into your version control system.

ActiveRecord::Schema.define(version: 2021\_12\_03\_104933) do

create\_table "friend\_nums", force: :cascade do |t|

t.integer "num"

t.string "str\_out"

t.datetime "created\_at", precision: 6, null: false

t.datetime "updated\_at", precision: 6, null: false

end

end

**friend\_num\_controller\_test.rb**

require "test\_helper"

class FriendNumControllerTest < ActionDispatch::IntegrationTest

# test "the truth" do

# assert true

# end

attr\_reader :num, :num2, :result

def get\_res(num)

post friend\_num\_url, params: {num:num}

assigns[:res][0].str\_out

end

setup do

@num = 3000

@num2 = 8000

@result = ' 220 and 284; 1184 and 1210; 2620 and 2924'

end

test 'should get home' do

get root\_url

assert\_response :success

end

test 'should post to result' do

post friend\_num\_url

assert\_response :success

end

test 'should get result' do

post friend\_num\_url, params: {num:@num}

assert\_response :success

assert\_equal assigns[:res][0].str\_out, @result

assert\_equal FriendNum.where('num = ?', @num)[0].str\_out, @result

end

test 'diff response' do

assert\_not\_equal get\_res(num), get\_res(num2)

end

end

**friend\_num\_test.rb**

# frozen\_string\_literal: true

require 'test\_helper'

class PalnumTest < ActiveSupport::TestCase

attr\_reader :num, :result, :fr\_num

setup do

@num = 3000

@result = ' 220 and 284; 1184 and 1210; 2620 and 2924'

@fr\_num = FriendNum.new

end

test 'should not create without data' do

assert\_not fr\_num.save

fr\_num = FriendNum.new(num: @num, str\_out: @result)

assert fr\_num.valid?

assert fr\_num.save

end

end







