***Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования***

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ \_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_

КАФЕДРА\_\_\_\_\_\_\_ КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по лабораторной работе № 11**

**Дисциплина: Языки интернет-программирования**

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

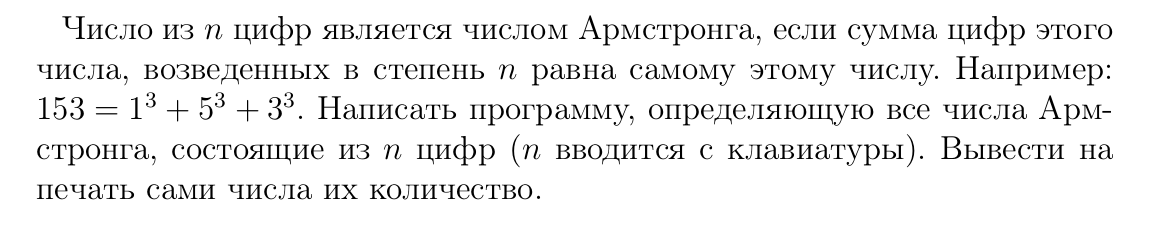
Студент гр. \_\_ИУ6-34 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лавренов И.С.**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2017

Задание

Код модели armstrong\_number.rb:

class ArmstrongNumber < ApplicationRecord

validates\_uniqueness\_of :input\_value

end

Код созданной миграции 20171219062712\_create\_armstrong\_numbers.rb:

class CreateArmstrongNumbers < ActiveRecord::Migration[5.1]

def change

create\_table :armstrong\_numbers do |t|

t.string :input\_value

t.string :output\_value

t.timestamps

end

end

end

Код контроллера armstrong\_number\_controller.rb:

require 'uri'

class ArmstrongNumberController < ApplicationController

before\_action :parse\_params, only: :output

def input

end

def output

if res = ArmstrongNumber.find\_by\_input\_value( @number )

@result = ActiveSupport::JSON::decode( res.output\_value )

else

find\_arm\_num

if @result[0].to\_i != 0

res = ArmstrongNumber.create :input\_value => @number, :output\_value => ActiveSupport::JSON::encode( @result )

res.save

@flag = true

end

end

respond\_to do |format|

format.html

format.json do

render json:

{input\_value: @number, result: @result}

end

end

end

def get\_last

last = ArmstrongNumber.last

@number = ActiveSupport::JSON::decode( last.input\_value )

@result = ActiveSupport::JSON::decode( last.output\_value )

render json:

{input: @number, result: @result}

end

def result\_in\_xml

@result = ArmstrongNumber.all

render xml:

{input: @number, result: @result}

end

protected

def find\_arm\_num

@numbers=Array.new

count = 0

if @number.to\_s.empty?

@result = :blank #"Заполните пустое поле"

elsif @number.to\_i == 0

@result = :nan #"Введите числовое значение"

else

num = @number.to\_i

(10\*\*(num-1)..10\*\*num - 1).to\_a.each do |ii|

wanted = ii

sum = 0

num.times do

sum += (ii % 10)\*\*num

ii /= 10

end

if wanted == sum

@numbers[count] = sum

count += 1

end

#return @numbers.any? ? @numbers : :not\_found #numbers : "Для чисел с #{number} цифрами нет чисел Армстронга"

if @numbers.any?

@result = @numbers

else

@result = :not\_found

end

end

end

end

def parse\_params

@number = params[:input\_value]

end

end

Код представления input.html.erb:

class CreateArmstrongNumbers < ActiveRecord::Migration[5.1]

def change

create\_table :armstrong\_numbers do |t|

t.string :input\_value

t.string :output\_value

t.timestamps

end

end

end

Код представления output.html.erb:

<!--%= render inline: @flag %-->

<div> <%= "Результат для '#{@number}':" %> </div>

<% if @result[1].to\_i == 0 %>

<div> <%= @result %> </div>

<% else %>

<style>

table{ width: 300px;

border: 2px solid;

text-align: center; }

td{border: 1px solid;}

</style>

<table >

<tr><th >№</th><th >Полученное число</th></tr>

<%count = 0%>

<% @result.each do |i|%>

<tr><td ><%=@result.index(i) + 1%></td><td><%=i.to\_s%></td></tr>

<%count += 1%>

<%end%>

</table>

<% end %>

<br/>

<%= form\_tag('/armstrong\_number/get\_last', :method => 'get') do %>

<%= submit\_tag("Показать последние данные") %>

<% end %>

<br/>

<%= form\_tag('/armstrong\_number/result\_in\_xml', :method => 'get') do %>

<%= submit\_tag("Вывести в .xml") %>

<% end %>

<br/>

<%= link\_to "Повторить вычисление", :root %>

Код для тестирования контроллера (armstrong\_number\_controller\_test.rb):

require 'test\_helper'

class ArmstrongNumberControllerTest < ActionDispatch::IntegrationTest

test "should get input" do

get armstrong\_number\_input\_url

assert\_response :success

end

test "should get output" do

get armstrong\_number\_output\_url

assert\_response :success

end

test 'should return xml' do

get armstrong\_number\_result\_in\_xml\_url

assert\_includes @response.headers['Content-Type'], 'application/xml'

end

test 'should return different results' do

get armstrong\_number\_output\_url, params: {input\_value: '3'}

result1 = response.parsed\_body

get armstrong\_number\_output\_url, params: {input\_value: '1'}

result2 = response.parsed\_body

assert\_not\_equal result1, result2

end

test "should return result" do

get armstrong\_number\_output\_url, params: {input\_value: '1'}

assigns[@result] = '[1, 2, 3, 4, 5, 6, 7, 8, 9]'

end

end

Код для тестирования модели (armstrong\_number\_test.rb):

require 'test\_helper'

class ArmstrongNumberTest < ActiveSupport::TestCase

test "the truth" do

assert true

end

=begin

test "not\_found" do

ArmstrongNumber.create(input\_value: '1', result\_value: '[1, 2, 3, 4, 5, 6, 7, 8, 9]')

assert\_equal 'not\_found', ArmstrongNumber.find\_by\_input\_value('2').output\_value

end

=end

test 'no to second' do

res1=ArmstrongNumber.create(input\_value: '1', output\_value: '[1, 2, 3, 4, 5, 6, 7, 8, 9]')

assert res1.save

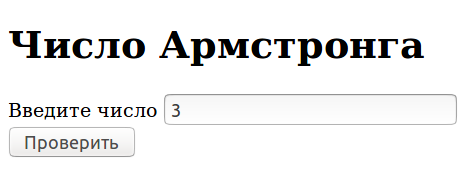
res2=ArmstrongNumber.create(input\_value: '1', output\_value: '[1, 2, 3, 4, 5, 6, 7, 8, 9]')

assert !res2.save

end

end

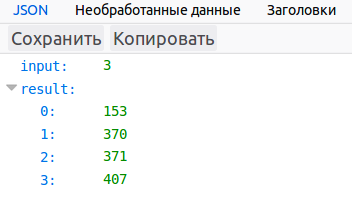
Результаты:



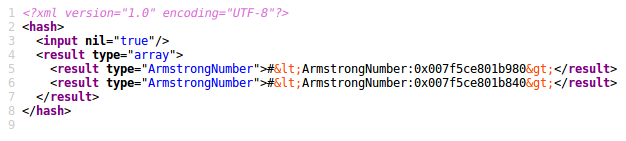
Ввод

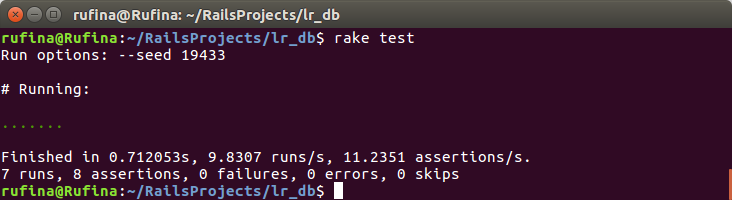


Вывод при нажатии на кнопку «Проверить»



Вывод последних введенных данных в базу в формате json

Вывод в xml



Результаты тестирования