



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

ИУ6-33Б
(Группа)

(Подпись, дата)

И.А. Нуруллаев
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2022

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 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 /.

Решение

```
WPL_bmstu - simons_controller.rb

1 require 'nokogiri'
2
3 # api controller for simons hyp.
4 class SimonsController < ActionController::Base
5   before_action :set_simons, only: [:show, :edit, :update, :destroy]
6   XSLT_SERVER_TRANSFORM = "#{Rails.root}/public/server_transform.xslt".freeze
7   def input; end
8
9   def index
10     @simons = Simons.all
11
12     respond_to do |format|
13       format.html do
14         render 'index'
15       end
16       format.xml do
17         xml_arr = @simons.inject([]) { |acc, el| acc.append el.result }
18         doc_result = Nokogiri::XML('<db></db>')
19         xml_arr.each_with_object(doc_result) do |el, acc|
20           el = Nokogiri::XML(el).search('result')
21           acc.at('db').add_child(el)
22         end
23         render xml: doc_result
24       end
25     end
26   end
27
28   def show
29     doc = Nokogiri::XML(@simons.result)
30     xslt = Nokogiri::XSLT(File.read(XSLT_SERVER_TRANSFORM))
31     @view = xslt.transform(doc).to_html
32   end
33
34   def new
35     @simons = Simons.new
36   end
37
38   def create
39     found_id = Simons.search_id(simons_params[:max_n])
40
41     if found_id.nil?
42       @simons = Simons.new(simons_params)
43       @simons.result = make_output(simons_params[:max_n]) if @simons.valid?
44       respond_to do |format|
45         if @simons.save
46           format.html do
47             redirect_to @simons, notice: 'Расчет успешно произведен.'
48           end
49           format.json { render :show, status: :created, location: @simons }
50         else
51           format.html { render :new, status: :unprocessable_entity }
52           format.json do
53             render json: @simons.errors, status: :unprocessable_entity
54           end
55         end
56       end
57     else
58       redirect_to action: 'show', id: found_id
59     end
60   end
61
62   def destroy
63     p 'deleting'
64     @simons.destroy
65     respond_to do |format|
66       format.html do
67         redirect_to simons_url, notice: 'Расчет успешно удален.'
68       end
69       format.json { head :no_content }
70     end
71   end
end
```

```

WPL_bmstu - simons_controller.rb

72
73 private
74
75 # Use callbacks to share common setup or constraints between actions.
76 def set_simons
77   @simons = Simons.find(params[:id])
78 end
79
80 # Only allow a list of trusted parameters through.
81 def simons_params
82   params.require(:simons).permit(:max_n)
83 end
84
85 def make_output(input)
86   input = Integer(input)
87   return if input.nil?
88
89   simons_enum = Enumerator.new do |steps|
90     a1 = 1
91     a2 = 2
92     a3 = 3
93     iteration = 1
94
95     loop do
96       fact = a1 * a2 * a3
97       steps << { iteration: iteration, f: fact, a1: a1, a2: a2, a3: a3, fits: factorial?(fact) }
98       a1 = a2
99       a2 = a3
100      a3 += 1
101      iteration += 1
102    end
103  end
104
105  simons_hash = simons_enum.take_while { |step| step[:iteration] < input }.select { |step| step[:fits] }
106  result = simons_hash.length
107
108  output_hash = { result: result || 'Не найдено', simons: simons_hash }
109  output_hash.to_xml(root: 'output')
110 end
111
112 def factorial?(number)
113   i = 1
114   n = number.to_f
115   while (n % i).zero?
116     n /= i
117     i += 1
118   end
119
120   n == 1
121 end
122 end
123

```

```

WPL_bmstu - simons.rb

1 class Simons < ApplicationRecord
2   validates :max_n, uniqueness: true
3   def self.search(search)
4     if search
5       where(max_n: search)
6     else
7       all
8     end
9   end
10
11   def self.search_id(search)
12     select(:id).find_by(max_n: search) if search
13   end
14 end
15

```

WPL_bmstu - _form.html.erb

```
1 <%= form_with(model: simons, url: simons_path ) do |form| %>
2   <% if simons.errors.any? %>
3     <div id="error_explanation">
4       <h2>Пожалуйста, исправьте следующие ошибки ввода:</h2>
5       <ul>
6         <% simons.errors[:input].each do |error| %>
7           <li>
8             <%= error %>
9           </li>
10          <% end %>
11        </ul>
12      </div>
13    <% end %>
14    <div class="input-group d-flex w-25">
15      <%= form.text_field :max_n, type: 'number' , placeholder: 'Введите n:' , class: 'form-control' , required: true %>
16      <%= form.submit 'Расчитать' , type: 'submit' , class: 'btn btn-primary' %>
17    </div>
18  <% end %>
```

WPL_bmstu - index.html.erb

```
1 <p id="notice"><%= notice %></p>
2 <h1>Гипотеза Симона</h1>
3 <div>
4   <%= link_to 'Произвести новый расчет', new_simon_path, class: 'btn btn-link btnsm' %>
5 </div>
6 <br/>
7 <table>
8   <thead align="center">
9     <tr>
10      <th>Введено</th>
11      <th>Результат</th>
12      <th colspan="3"></th>
13    </tr>
14  </thead>
15  <tbody>
16    <% @simons.each do |simon| %>
17      <tr>
18        <td><%= simon.max_n %></td>
19        <td><%= simon.result %></td>
20        <td><%= link_to 'Посмотреть', simon, class: 'btn btn-link btn-sm' %></td>
21        <td><%= button_to 'Удалить', simon, method: :delete, data: { confirm: 'Вы уверены?' }, class: 'btn btn-link btn-sm' %></td>
22      </tr>
23    <% end %>
24  </tbody>
25 </table>
```

WPL_bmstu - show.html.erb

```
1 <p id="notice"><%= notice %></p>
2 <h1>Результат расчета</h1>
3 <p><strong>Вы ввели: </strong>
4 <%= @simons.max_n %></p>
5 <%= render inline: @view %>
6 <%= link_to 'На главную', simons_path, class: 'btn btn-link btn-sm' %>
7 <%= link_to 'Ввести новое число', new_simon_path, class: 'btn btn-link btnsm' %>
```

WPL_bmstu - new.html.erb

```
1 <h1>Гипотеза Симона</h1>
2 <p style="font-size: 16px">Введите n, до которого проверить гипотезу (как в прошлых ЛР)</p>
3 <%= render 'form', simons: @simons %><br/>
4 <%= link_to 'Вернуться на главную', simons_path, class: 'btn btn-link btn-sm' %>
```

WPL_bmstu - routes.rb

```
1 Rails.application.routes.draw do
2   root 'simons#new'
3   resources :simons, only: %i[index show destroy new create]
4 end
5
```

WPL_bmstu - simons_test.rb

```
1 require 'test_helper'
2
3 # simons model test
4 class SimonsTest < ActiveSupport::TestCase
5   test 'check correct input' do
6     simons = Simons.new(max_n: 12)
7     assert simons.valid?
8   end
9   test 'check duplicated input' do
10    simons1 = Simons.new(max_n: 11_122)
11    simons2 = Simons.new(max_n: 11_122)
12    assert simons1.save
13    refute simons2.save
14  end
15  test 'check that record can be saved and read' do
16    simons = Simons.new(max_n: 1234)
17    assert simons.save
18    assert Simons.find_by(max_n: 1234)
19  end
20 end
21
```

WPL_bmstu - simons_controller_test.rb

```
1 require 'test_helper'
2
3 # simons controller test
4 class SimonsControllerTest < ActionDispatch::IntegrationTest
5   setup do
6     @controller = SimonsController.new
7   end
8
9   test 'should get index' do
10     get simons_path
11     assert_response :success
12   end
13
14   test 'should get new' do
15     get new_simon_path
16     assert_response :success
17   end
18
19   test 'should create simons' do
20     assert_difference('Simons.count', 1) do
21       post simons_path, params: { 'simons[max_n]': 100 }
22     end
23   end
24
25   test 'should show simons' do
26     max_n = 100
27     assert_difference('Simons.count', 1) do
28       post simons_path, params: { 'simons[max_n]': max_n }
29     end
30
31     @simons = Simons.all[0]
32     assert_equal @simons.max_n, max_n
33
34     get simon_path(@simons)
35     assert_response :success
36   end
37
38   test 'should destroy simons' do
39     max_n = 100
40     assert_difference('Simons.count', 1) do
41       post simons_path, params: { 'simons[max_n]': max_n }
42     end
43
44     @simons = Simons.all[0]
45     assert_equal @simons.max_n, max_n
46
47     assert_difference('Simons.count', -1) do
48       delete simon_path(@simons)
49     end
50     assert_redirected_to simons_path
51   end
52
53   test 'should get different outputs for different inputs' do
54     output1 = @controller.send :make_output, '2'
55     output2 = @controller.send :make_output, '10'
56     assert_not_equal output1, output2
57   end
58 end
59
```


←

→

↺

🏠

📄

localhost:3000

Гипотеза Симона

Введите n, до которого проверить гипотезу (как в прошлых ЛР)

100

⌵

Расчитать

[Вернуться на главную](#)

←

→

↺

🏠

📄

localhost:3000/simons/5

Расчет успешно произведен.

Результат расчета

Вы ввели: 100

Ответ: 4

Факториал Последовательность

6	1 * 2 * 3
24	2 * 3 * 4
120	4 * 5 * 6
720	8 * 9 * 10

[На главную](#) [Ввести новое число](#)

←

→

↺

🏠

📄

localhost:3000/simons

🔗

☆

Гипотеза Симона

[Произвести новый расчет](#)

Введено	Результат	
100	<pre><?xml version="1.0" encoding="UTF-8"?> <output> <result type="integer">4</result> <simons type="array"> <simon> <iteration type="integer">1</iteration> <f type="integer">6</f> <a1 type="integer">1</a1> <a2 type="integer">2</a2> <a3 type="integer">3</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration type="integer">2</iteration> <f type="integer">24</f> <a1 type="integer">2</a1> <a2 type="integer">3</a2> <a3 type="integer">4</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration type="integer">4</iteration> <f type="integer">120</f> <a1 type="integer">4</a1> <a2 type="integer">5</a2> <a3 type="integer">6</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration type="integer">8</iteration> <f type="integer">720</f> <a1 type="integer">8</a1> <a2 type="integer">9</a2> <a3 type="integer">10</a3> <fits type="boolean">true</fits> </simon> </simons> </output></pre>	Посмотреть Удалить
1000	<pre><?xml version="1.0" encoding="UTF-8"?> <output> <result type="integer">4</result> <simons type="array"> <simon> <iteration type="integer">1</iteration> <f type="integer">6</f> <a1 type="integer">1</a1> <a2 type="integer">2</a2> <a3 type="integer">3</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration type="integer">2</iteration> <f type="integer">24</f> <a1 type="integer">2</a1> <a2 type="integer">3</a2> <a3 type="integer">4</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration type="integer">4</iteration> <f type="integer">120</f> <a1 type="integer">4</a1> <a2 type="integer">5</a2> <a3 type="integer">6</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration type="integer">8</iteration> <f type="integer">720</f> <a1 type="integer">8</a1> <a2 type="integer">9</a2> <a3 type="integer">10</a3> <fits type="boolean">true</fits> </simon> </simons> </output></pre>	Посмотреть Удалить
5	<pre><?xml version="1.0" encoding="UTF-8"?> <output> <result type="integer">3</result> <simons type="array"> <simon> <iteration type="integer">1</iteration> <f type="integer">6</f> <a1 type="integer">1</a1> <a2 type="integer">2</a2> <a3 type="integer">3</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration type="integer">2</iteration> <f type="integer">24</f> <a1 type="integer">2</a1> <a2 type="integer">3</a2> <a3 type="integer">4</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration type="integer">4</iteration> <f type="integer">120</f> <a1 type="integer">4</a1> <a2 type="integer">5</a2> <a3 type="integer">6</a3> <fits type="boolean">true</fits> </simon> </simons> </output></pre>	Посмотреть Удалить

Содержимое базы данных:

Reset Filters Records: 3					
id	max_n	result	created_at	updated_at	
1	5	100	<?xml version="1.0" encoding="UTF-8"?> <output> <result type="integer">4</result> <simons...	2022-11-24 20:02:11...	2022-11-24 20:02:11...
2	6	1000	<?xml version="1.0" encoding="UTF-8"?> <output> <result type="integer">4</result> <simons...	2022-11-24 20:02:55...	2022-11-24 20:02:55...
3	7	5	<?xml version="1.0" encoding="UTF-8"?> <output> <result type="integer">3</result> <simons...	2022-11-24 20:03:06...	2022-11-24 20:03:06...

```
<?xml version="1.0" encoding="UTF-8"?> <output> <result type="integer">4</result> <simons type="array">
<simon> <iteration type="integer">1</iteration> <f type="integer">6</f> <a1 type="integer">1</a1> <a2
type="integer">2</a2> <a3 type="integer">3</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration
type="integer">2</iteration> <f type="integer">24</f> <a1 type="integer">2</a1> <a2 type="integer">3</a2>
<a3 type="integer">4</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration
type="integer">4</iteration> <f type="integer">120</f> <a1 type="integer">4</a1> <a2 type="integer">5</a2>
<a3 type="integer">6</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration
type="integer">8</iteration> <f type="integer">720</f> <a1 type="integer">8</a1> <a2 type="integer">9</a2>
<a3 type="integer">10</a3> <fits type="boolean">true</fits> </simon> </simons> </output>
<?xml version="1.0" encoding="UTF-8"?> <output> <result type="integer">4</result> <simons type="array">
<simon> <iteration type="integer">1</iteration> <f type="integer">6</f> <a1 type="integer">1</a1> <a2
type="integer">2</a2> <a3 type="integer">3</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration
type="integer">2</iteration> <f type="integer">24</f> <a1 type="integer">2</a1> <a2 type="integer">3</a2>
<a3 type="integer">4</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration
type="integer">4</iteration> <f type="integer">120</f> <a1 type="integer">4</a1> <a2 type="integer">5</a2>
<a3 type="integer">6</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration
type="integer">8</iteration> <f type="integer">720</f> <a1 type="integer">8</a1> <a2 type="integer">9</a2>
<a3 type="integer">10</a3> <fits type="boolean">true</fits> </simon> </simons> </output>
<?xml version="1.0" encoding="UTF-8"?> <output> <result type="integer">3</result> <simons type="array">
<simon> <iteration type="integer">1</iteration> <f type="integer">6</f> <a1 type="integer">1</a1> <a2
type="integer">2</a2> <a3 type="integer">3</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration
type="integer">2</iteration> <f type="integer">24</f> <a1 type="integer">2</a1> <a2 type="integer">3</a2>
<a3 type="integer">4</a3> <fits type="boolean">true</fits> </simon> <simon> <iteration
type="integer">4</iteration> <f type="integer">120</f> <a1 type="integer">4</a1> <a2 type="integer">5</a2>
<a3 type="integer">6</a3> <fits type="boolean">true</fits> </simon> </simons> </output>
```

Примеры запросов:

```
SELECT "simons".* FROM "simons"
```

Выбор всех расчетов

```
SELECT "simons".* FROM "simons" WHERE "simons"."max_n" < 200 LIMIT 10
```

Выбор 10 расчетов где максимальное количество итераций меньше 200

```
INSERT INTO "simons" ("max_n", "result", "created_at", "updated_at") VALUES (?, ?, ?, ?)
```

Добавление новой записи вручную

Итоговый код данной лабораторной работы доступен по ссылке:

https://github.com/tenessinum/WPL_bmstu/tree/main/Lab11