



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 2

Название: Сессии. Выполнение авторизации. Интеграционные тесты.

Дисциплина: Языки интернет-программирования

Студент

ИУ6-32Б

(Группа)

(Подпись, дата)

Джафаров Э.Э.

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Петрова Я.С.

(И.О. Фамилия)

Москва, 2022

Цель работы — получить практические навыки работы с сессиями, с авторизацией и с интеграционными тестами в Rails,

Задание:

Модифицировать код приложения ЛР 8 таким образом, чтобы вычисление было невозможно без регистрации пользователя и аутентификации при помощи логина/пароля.

- Сгенерировать при помощи генератора scaffold ресурс для регистрации пользователей.
- Создать БД и выполнить миграцию соответствующим запросом rake.
- Проверить возможность добавления, редактирования информации и получения списка пользователей.
- Удалить отображение поля пароля при просмотре списка пользователей.
- Добавить контроллер сессий.
- Реализовать форму для ввода логина/пароля при обращении по адресу /. Добавить ссылку на регистрацию нового пользователя. При успешном вводе логина/пароля должно осуществляться перенаправление на страницу ввода параметров для вычисления.
- Реализовать при помощи контроллера сессий во всех действиях контроллера проверку о того, прошел ли пользователь аутентификацию или нет (с выдачей соответствующей отладочной информации).
- Вставить фильтры для запроса аутентификации.
- Подготовить интеграционный тест, позволяющий проверить регистрацию нового пользователя, вход под его именем и выполнение вычислений.
- Подготовить интеграционный тест для проверки невозможности выполнения вычислений без ввода логина/пароля.
- Проверить маршруты приложения с помощью rake routes и убрать лишние. Обеспечить доступ при обращении по адресу /.

Решение:

1. Контроллер calc_controller.rb

```
class SessionsController < ApplicationController
  attr_writer :current_user

  def new
  end

  def create
    user = User.authenticate(params[:email], params[:password])
    if user.nil?
      respond_to do |format|
        format.html { redirect_to root_path, alert: 'Леее куда прешь ☹️ неправильным паролем или логином' }
      end
    else
      sign_in user
      redirect_to :calc_input
    end
  end

  def exit
    sign_out
    redirect_to :root
  end

  def sign_in(user)
    session[:current_user_id] = user.id
    self.current_user = user
  end

  def sign_out
    session[:current_user_id] = nil
    self.current_user = nil
  end
end
```

2. Модель user.rb

```
class User < ApplicationRecord
  def has_password?(submitted_password)
    password == submitted_password
  end

  def self.authenticate(email, submitted_password)
    user = find_by_email(email)
    return nil if user.nil?
    return user if user.has_password?(submitted_password)
  end
end
```

3. input.html.erb

```
<% if not User.find_by_id(session[:current_user_id]) %>
  <%= controller.respond_to do |format| %>
    |   <%= format.html { controller.redirect_to root_path, alert: 'Леее, куда прешь брат без авторизации' } %>
  <% end %>
<% end %>
<p align="center"><strong>Анкара Месси Анкара Месси Анкара Месси Анкара Месси</strong></p>
<%= image_tag "Messi.jpeg", style: "position:fixed; right: 2%" %>
<h1>Брат, введи число от 1 до 7.</h1>
<%= form_tag("/calc/show", :method => :get) do %>
  <%= label_tag("Введите количество цифр в числе:") %>
  </br>
  <%= number_field_tag(:power, nil, in: 1..7, step:1)%>
  </br>
  <%= submit_tag("Yosh") %>
<% end %>
<%= link_to "Выйти", sessions_exit_path %>
</br>
```

4. show.html.erb

```
<%= image_tag "cot.jpeg", style: "position:fixed; right: 25%" %>
<div align="left">
<h1>Посчитали, брат</h1>
<p>Числа Армстронга:</p>
<table border=1>
  <tr>
    <td> Индекс </td>
    <td> Число </td>
  </tr>
  <% @right_numbers.each do |number| %>
    <tr>
      <td> <%= number[0] %> </td>
      <td> <%= number[1] %> </td>
    </tr>
  <% end %>
</table>
<p>Количество чисел Армстронга: <%= @right_numbers.length %></p>
<p><%= @msg %></p>
</br>
<%= link_to "Показать БД в JSON", '/users/json' %>
</br>
<%= link_to "Показать БД в XML", '/users/xml' %>
<%= button_to 'Вернуться', calc_input_url, method: :get, class: 'btn btn-primary' %>
</br>
<%= link_to "Выйти", sessions_exit_path %>
</div>
```

5. session_new.html.erb

```
<h1>Заходи брат, гостем дорогим будешь</h1>

<p style="color: green"><%= notice %></p>
<p style="color: red"><%= alert %></p>

<%= form_tag("/sessions/create", :method => "get") do %>
  <%= label_tag("Почта:") %> <br>
  <%= text_field_tag(:email) %> <br/>
  <%= label_tag("Пароль:") %> <br>
  <%= password_field_tag(:password) %> <br/>
  <br/>
  <%= submit_tag("Sign in") %>
<%= end %>
<%= link_to "Новый гость", new_user_path %>
</br>
<%= if params[:message] %>
  <p><%= params[:message] %></p>
<%= end %>

</br>
<%= image_tag "Malaxov_and_Messi.png" %>
```

6. Тест контроллера

```
require 'selenium-webdriver'
require 'json'
RSpec.describe 'Authorization and registration test' do
  before(:each) do
    @driver = Selenium::WebDriver.for :firefox
    @driver.manage.window.maximize
  end
  after(:each) do
    sleep(5)
    @driver.quit
  end
  it 'should reg user' do
    @driver.get('http://localhost:3000/')
    @driver.find_element(:link_text, 'Новый гость').click
    sleep(1)
    @driver.find_element(:id, 'user_username').send_keys('gav')
    @driver.find_element(:id, 'user_email').send_keys('gav@example.ru')
    @driver.find_element(:id, 'user_password').send_keys('897654')
    @driver.find_element(:name, 'commit').click
    sleep(3)
    expect(@driver.find_element(:xpath, '//p[2]').text).to eq("User was successfully created.")
  end
  it 'should auth user' do
    @driver.get('http://localhost:3000/')
    @driver.find_element(:id, 'email').send_keys("djafarovemil04@mail.ru")
    @driver.find_element(:id, 'password').send_keys(123456)
    @driver.find_element(:name, 'commit').click
    sleep(3)
    expect(@driver.find_element(:xpath, '//p').text).to eq("Current user:MrDzhofik")
  end
  it 'without login' do
    @driver.get('http://localhost:3000/')
    sleep(3)
    expect(@driver.find_element(:xpath, '//p').text).to eq('Not logged in')
  end
  it 'kick without login' do
    @driver.get('http://localhost:3000/calc/input')
    sleep(3)
    expect(@driver.find_element(:xpath, '//p[3]').text).to eq('Леее, куда прешь брат без авторизации')
  end
end
```

7. Результат:

```
....
Finished in 1 minute 4.87 seconds (files took 0.32491 seconds to load)
4 examples, 0 failures
```

8. Результат работы:

8.1 Страница входа:

Not logged in

Заходи брат, гостем дорогим будем

Почта:

Пароль:

[Sign in](#)

[НОВЫЙ ГОСТЬ](#)

Я с [REDACTED] после некорректной установки **Ruby**



Малаховым



8.2 Страница ввода:

Current user:MrDzhofik

Брат, введи число от 1 до 7.

Введите количество цифр в числе:

Анкара Месси Анкара Месси Анкара Месси Анкара Месси



8.3 Страница вывода:

Current user:MrDzhofik

Посчитали, брат

Числа Армстронга:

Индекс	Число
0	548834

Количество чисел Армстронга: 1

Data from DB

[Показать БД в JSON](#)

[Показать БД в XML](#)

[Выйти](#)



8.4 Страница регистрации:

Not logged in

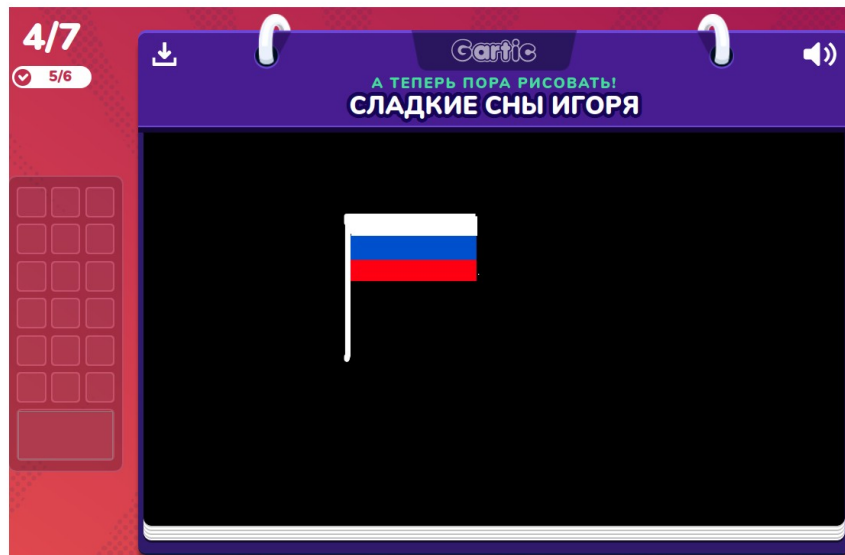
New user

Username

Email

Password

[Назад](#)



9. База данных в XML

MrDzhofik djafarovemil04@mail.ru Ksenia mevain@mail.ru Simon Maksimov sms@mail.ru Igoreha igoreha@example.mail.ru gav gav@example.ru

Вывод:

Я научился работать с сессиями, авторизацией, регистрацией и интеграционными тестами в Rails