



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

ИУ6-31Б

(Группа)

(Подпись, дата)

Лазарев Е.С.

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Малахов Д.В.

(И.О. Фамилия)

Москва, 2022

Цель работы: изучить базовый принцип авторизации пользователей в Rails

ЛР 12. Сессии. Выполнение авторизации. Интеграционные тесты.

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 том, 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 /.

Index Controller:

```
# frozen_string_literal: true
require 'prime'
class IndexController < ApplicationController
  before_action :raise_error, only: [:view]
  def input
  end
  def output
    v1 = params[:v1].to_i
    k = 2 ** ((v1.to_s(2)).length - 1)
    l = (Math.log(k)/Math.log(2)).to_i
    prime_array = []
    (1..l).each do |i|
      if Prime.prime?(i) == true
        prime_array << i
      end
    end
    @result = []
    (0..prime_array.length).each do |i|
      if Prime.prime?(2**prime_array[i] - 1) == true
        @result << (2**prime_array[i] - 1)
      end
    end
  end
  def raise_error
    if params[:v1].nil? || params[:v1].empty?
      flash[:error] = 'Параметры не должны быть пустыми.'
      redirect_to root_path
    end
  end
end
```

Session Controller

```
# frozen_string_literal: false
class SessionController < ApplicationController
  skip_before_action :require_login, only: %i[login create]
  def login; end

  def create
    user = User.find_by_username(params[:username])
```

```

if user&.authenticate(params[:password])
  sign_in user
  redirect_to root_url
else
  if user.nil?
    flash[:notice] = 'Неправильный логин'
  else
    flash[:notice] = 'Неправильный пароль'
  end
  p flash[:notice]
  redirect_to session_login_url
end
end

def logout
  sign_out
  redirect_to session_login_url
end
end

```

User Controller

```

# frozen_string_literal: false
class UsersController < ApplicationController
  before_action :set_user, only: %i[show edit update destroy]
  skip_before_action :require_login, only: %i[create new]

  def index
    @users = User.all
  end

  def show; end

  def new
    @user = User.new
  end

  def edit; end

  def create
    @user = User.new(user_params)
    respond_to do |format|
      if @user.save
        sign_in @user
        format.html { redirect_to root_path, notice: 'Пользователь успешно создан!' }
      else
        format.html { render :new, status: :unprocessable_entity }
      end
    end
  end

  def update
    respond_to do |format|
      if @user.update(user_params)
        format.html { redirect_to user_url(@user), notice: 'Пользователь успешно обновлён!' }
      else
        format.html { render :edit, status: :unprocessable_entity }
      end
    end
  end

  def destroy
    @user.destroy
  end

```

```

respond_to do |format|
  format.html { redirect_to users_url, notice: 'Пользователь успешно уничтожен!' }
end
end

private

def set_user
  @user = User.find(params[:id])
end

def user_params
  params.require(:user).permit(:username, :password, :password_confirmation)
end
end

```

Input(index):

```

<%unless flash[:notice].blank?%>
<h2 style = 'color:red'><%=flash[:notice]%></h2>
<%end%>
<h1>Числа Мерсена</h1>
<%= form_with url: :output, method: :get do |form| %>
<%= form.number_field :v1, in:1..Float::INFINITY%>
<%= form.submit "Отправить", id: 'submit-btn' %>
<% end %>
<%=link_to 'Выйти из учетной записи', session_logout_path, class:"button"%>
<%=link_to 'Просмотреть базу с пользователями', :users, class:"button"%>

```

Output(index):

```

<style>
table {
border-collapse: collapse;
width: 100%;
}

th, td {
text-align: left;
padding: 8px;
}

tr:nth-child(even) {background-color: #f2f2f2;}

</style>
<h1>Числа Мерсена</h1>
<table border="1">
<th>Номер </th> <th> Число Мерсена </th>
<tr> <%(0..@result.length-1).each do |x| %>
<td> <%= x + 1 %> </td> <td> <%= @result[x] %> </td>
</tr>
<% end %>
</table>
<%=link_to 'Вернуться',root_path, class:"button"%><br/>

```

Login(session):

```

<%unless flash[:notice].blank?%>

```

```

<h2 style = 'color:red'><%=flash[:notice]%></h2>
<%=end%>

<%=form_with url: session_create_url, method: :post, local:true do |f|%>
<table id='login'>
<tr>
<td><label>Имя пользователя</label></td>
<td><%=f.text_field :username, require: true %><br/></td>
</tr>
<tr>
<td><label>Пароль</label></td>
<td><%=f.password_field :password, require: true%><br/></td>
</tr>
<tr>
<td><%=f.submit id: 'submit-btn', value: 'Вход',require: true%></td>
</tr>
</table>
<%=end%>

<%=link_to 'Регистрация', new_user_path, class:"button"%>

```

_Form(user)

```

<%= form_with(model: user) do |form| %>
<%= if user.errors.any? %>
<div style="color: red">
<h2>Пользователь не был создан:</h2>

<ul>
<%= user.errors.each do |error| %>
<%= p error%>
<li><%= error.full_message%></li>
<%= end %>
</ul>
</div>
<%= end %>

<div>
<%= form.label :username, value: 'Имя пользователя',style: "display: block" %>
<%= form.text_field :username %>
</div>

<div>
<%= form.label :password, value: 'Пароль', style: "display: block" %>
<%= form.password_field :password %>
</div>

<div>
<%= form.label :lbl, value: 'Подтвердите пароль', style: "display: block" %>
<%= form.password_field :password_confirmation %>
</div>

<div>
<%= form.submit 'Создать', class:"button", id: 'submit-btn'%>
</div>
<%= end %>

```

User(user)

```

<div id="<%= dom_id user %>">
<p>
<strong>Имя:</strong>

```

```

<%= user.username %>
</p>

<p>
<strong>Пароль:</strong>
<%= user.password_digest %>
</p>

</div>

```

Index(user)

```

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

```

```

<h1>Пользователи</h1>

```

```

<div id="users">
<% @users.each do |user| %>
<p><%= (user.username) %>
<%= link_to "Показать данные пользователя", user %>
</p>
<% end %>
</div>

```

```

<%= link_to "Новый пользователь", new_user_path, class:"button" %>
<%= link_to "Войти в систему", session_logout_path, class:"button"%>
<%=link_to 'Найти числа Мерсена', root_path, class:"button"%>

```

Show(user)

```

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

```

```

<b>Имя: </b><%= @user.username %><br>
<b>Пароль: </b> Confidential

```

```

<div>
<%= link_to "Изменить", edit_user_path(@user), class:"button" %>
<%= link_to "Вернуться", users_path, class:"button" %>
<%= button_to "Удалить", @user, class:"button", method: :delete %>
</div>

```

Test:

```

require 'rails_helper'
require 'spec_helper'

```

```

RSpec.describe 'Static content', type: :system do
  scenario 'Register, login, calculate' do
    visit new_user_path
    fill_in :user_username, with: 'Lololoshka'
    fill_in :user_password, with: 'ILoveMinecraft'
    fill_in :user_password_confirmation, with: 'ILoveMinecraft'

```

```

    find('#submit-btn').click
    expect(current_path).to eq root_path
    visit session_login_path
    fill_in :username, with: 'Lololoshka'
    fill_in :password, with: 'ILoveMinecraft'
    find('#submit-btn').click

```

```

    fill_in :v1, with: 600
    find('#submit-btn').click
    expect(current_path).to eq '/index/output'
  end

```

```
def to_path(path)
  visit path
  expect(current_path).to eq session_login_path
end

scenario 'Check without register' do
  User.create(username: 'NotHim', password_digest: 'Pass')
  routes = ['/users/1', '/users/1/edit', '/index/output', root_path, '/users']
  routes.each do |path|
    to_path(path)
  end
end
```

← → ↻ ⓘ localhost:3000/session/login

YouTube Карты 3-ий семестр... Перевести Install Ruby O...

Имя пользователя

Пароль

Вход

Регистрация

Новый пользователь

Имя пользователя

Egor

Пароль

••

Подтвердите пароль

••

Создать

Вернуться

Числа Мерсена

Номер	Число Мерсена
1	3
2	7
3	31
4	127

[Вернуться](#)

Вывод: я изучил базовый принцип авторизации пользователей в Rails