



**«Московский государственный технический университет
имени Н.Э. Баумана»**

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети(ИУ6)

О т ч е т

по лабораторной работе № 12

Дисциплина: Языки интернет-программирования

Название лабораторной работы: Сессии. Выполнение авторизации.
Интеграционные тесты.

Студент гр. ИУ6-32Б _____ Д.В.Лабзунова
(Подпись, дата) (И.О. Фамилия)

Преподаватель _____
(Подпись, дата) (И.О. Фамилия)

Москва, 2020

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 /.

Код:

application_controller:

```
# frozen_string_literal: true
```

```
# application
```

```
# :reek:IrresponsibleModule
```

```
class ApplicationController < ActionController::Base  
  include
```

```
    before_action :require_login
```

```
  private
```

```
  def require_login
```

```
    redirect_to session_login_url unless signed_in?
```

```
  end
```

```
end
```

users_controller:

```
# frozen_string_literal: true
```

```
# Users controller
```

```
# :reek:DuplicateMethodCall
```

```
# :reek:TooManyStatements
```

```
# :reek:InstanceVariableAssumption
```

```
class UsersController <
```

```
  before_action :set_user, only: [:show, :edit, :update, :destroy]
```

```
  skip_before_action :require_login, only: %i[new create]
```

```
  # GET /users
```

```
  # GET /users.json
```

```
  def index
```

```

    @users =      .all
  end

  # GET /users/1
  # GET /users/1.json
  def show; end

  # GET /users/new
  def new
    @user =      .new
  end

  # GET /users/1/edit
  def edit; end

  # POST /users
  # POST /users.json
  def create
    @user =      .new(user_params)

    respond_to do |format|
      if @user.save
        format.html { redirect_to @user, notice: 'User was successfully created.' }
        format.json { render :show, status: :created, location: @user }
      else
        format.html { render :new }
        format.json { render json: @user.errors, status: :unprocessable_entity }
      end
    end
  end

  # PATCH/PUT /users/1
  # PATCH/PUT /users/1.json
  def update
    respond_to do |format|
      if @user.update(user_params)
        format.html { redirect_to @user, notice: 'User was successfully updated.' }
        format.json { render :show, status: :ok, location: @user }
      else
        format.html { render :edit }
        format.json { render json: @user.errors, status: :unprocessable_entity }
      end
    end
  end

  # DELETE /users/1
  # DELETE /users/1.json
  def destroy
    @user.destroy
    respond_to do |format|
      format.html { redirect_to users_url, notice: 'User was successfully destroyed.' }
      format.json { head :no_content }
    end
  end

  private

  # Use callbacks to share common setup or constraints between actions.
  def set_user
    @user =      .find(params[:id])
  end
end

```

```

# Only allow a list of trusted parameters through.
def user_params
  params.require(:user).permit(:username, :password, :password_confirmation)
end
end

```

session controller:

```
# frozen_string_literal: true
```

```
# Session controller
```

```
# :reek:TooManyStatements
```

```
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
```

```
      p 'Login'
```

```
      sign_in user
```

```
      redirect_to root_url
```

```
    else
```

```
      p 'redirect'
```

```
      redirect_to session_login_url
```

```
    end
```

```
  end
```

```
  def logout
```

```
    sign_out
```

```
    redirect_to session_login_url
```

```
  end
```

```
end
```

output controller:

```
# frozen_string_literal: true
```

```
# Output controller
```

```
# :reek:UtilityFunction
```

```
# :reek:TooManyStatements
```

```
# :reek:UncommunicativeVariableName
```

```
# :reek:IrresponsibleModule
```

```
class OutputController < ApplicationController
```

```
  def output
```

```
    sequence = params[:sequence]
```

```
    @error = 1
```

```
    if sequence == "
```

```
      @output = 'you did not write anything:('
```

```
    elsif sequence.to_i.zero?
```

```
      @output = 'please, enter a number'
```

```
    else
```

```
      @output = "Your initial sequence: #{sequence}"
```

```
      @error = 0
```

```

    res = find(sequence)
    subsequences_ = res[0]
    max = res[1]
    if subsequences_.length.zero?
      @subsequences = "There are no rising subsequences"
      @error = 1
    else
      @subsequences = subsequences_
      @max = "maximum length rising subsequence: #{max}"
    end
  end
end
end

def find(sequence)
  rising_subsequences = []
  subsequence = sequence[0]
  max = ""
  count = 1
  n = 0
  (1..sequence.length).each do |i|
    curr_seq = sequence[i]
    if curr_seq.to_i > sequence[i - 1].to_i
      count += 1
      subsequence += curr_seq
    else
      if count > 1
        rising_subsequences[n] = subsequence
        max = subsequence if subsequence.length > max.length
        n += 1
      end
      count = 1
      subsequence = curr_seq
    end
  end
  [rising_subsequences, max]
end
end

```

login.html.erb:

```

<h1>Session#login</h1>
<p>Find me in app/views/session/login.html.erb</p>
<%= form_with url: session_create_url, method: :post, local:true do |f| %>
  <%= f.text_field :username, required: true, label: 'username' %>
  <%= f.password_field :password, required: true, label: 'password' %>
  <%= f.submit 'log in' %>
<% end %>

<%= link_to 'sign up', new_user_path %>

```

form.html.erb:

```

<%= form_with(model: user, local: true) do |form| %>
  <% if user.errors.any? %>
    <div id="error_explanation">
      <h2><%= pluralize(user.errors.count, "error") %> prohibited this user from being saved:</h2>

      <ul>
        <% user.errors.full_messages.each do |message| %>
          <li><%= message %></li>
        <% end %>
      </ul>
    </div>
  <% end %>

```

```

<div class="field">
  <%= form.label :username %>
  <%= form.text_field :username %>
</div>

<div class="field">
  <%= form.label :password %>
  <%= form.password_field :password %>
</div>

<div class="field">
  <%= form.label :password_confirmation %>
  <%= form.password_field :password_confirmation %>
</div>

<div class="actions">
  <%= form.submit %>
</div>
<% end %>

```

edit.html.erb:

```

<h1>Editing User</h1>

<%= render 'form', user: @user %>

<%= link_to 'Show', @user %> |
<%= link_to 'Back', users_path %>

```

index.html.erb:

```

<p id="notice"><%= notice %></p>

<h1>Users</h1>

<table>
  <thead>
    <tr>
      <th>Username</th>
      <th colspan="3"></th>
    </tr>
  </thead>

  <tbody>
    <% @users.each do |user| %>
      <tr>
        <td><%= user.username %></td>
        <td><%= link_to 'Show', user %></td>
        <td><%= link_to 'Edit', edit_user_path(user) %></td>
        <td><%= link_to 'Destroy', user, method: :delete, data: { confirm: 'Are you sure?' } %></td>
      </tr>
    <% end %>
  </tbody>
</table>

<br>

<%= link_to 'New User', new_user_path %>
<hr>
<%= link_to 'START CALCULATING', root_url(@user) %>

```

new.html.erb:

```

<h1>New User</h1>

<%= render 'form', user: @user %>

```

```
<%= link_to 'Back', session_login_url %>
```

show.html.erb:

```
<p id="notice"><%= notice %></p>

<p>
  <strong>Username:</strong>
  <%= @user.username %>
</p>

<p>
  <strong>Password digest:</strong>
  <%= @user.password_digest %>
</p>

<p>
  <strong>Last login at:</strong>
  <%= @user.last_login_at %>
</p>
<hr>
<%= link_to 'Edit', edit_user_path(@user) %> |
<%= link_to 'Back', users_path %>
<hr>
<%= link_to 'START CALCULATING', root_url(@user) %>
```

session_helper:

```
module SessionHelper
```

```
  def sign_in(user)
    cookies.signed[:user_id] = { value: user.id, expires: 10.days }
    self.current_user = user
  end
```

```
  def signed_in?
    !current_user.nil?
  end
```

```
  def sign_out
    cookies.signed[:user_id] = nil
    self.current_user = nil
  end
```

```
  def current_user=(user)
    @current_user = user
  end
```

```
  def current_user
    @current_user ||= User.find_by(id: cookies.signed[:user_id]) # "если current_user на данном этапе nil, то нужно
выполнить..."
    # иначе - "
  end
end
```

routes.rb:

```
  .application.routes.draw do
    get 'session/login'
    post 'session/create'
    get 'session/logout'
    get '/logout', to: 'session#logout'
    resources :users
```

```

# For details on the DSL available within this file, see https://guides.rubyonrails.org/routing.html
root 'input#index'
get 'output' => 'output#output'
end

```

session_flows_test.rb:

```
require 'test_helper'
```

```

class SessionFlowsTest < ActionDispatch::IntegrationTest
  test 'unauthorized user will be redirected to login page' do
    get root_url
    assert_redirected_to controller: :session, action: :login
  end

  test 'user with incorrect credentials will be redirected to login page' do
    post session_create_url, params: { username: "hello", password: "1111" }
    assert_redirected_to controller: :session, action: :login
  end

  test 'user with correct credentials will see the root' do
    password = "1111"
    user = .create(username: "hello", password: password, password_confirmation: password)
    post session_create_url, params: { username: user.username, password: password }
    assert_redirected_to root_url
  end

  test 'user can logout' do
    password = "1111"
    user = .create(username: "hello", password: password, password_confirmation: password)
    post session_create_url, params: { login: user.username, password: password }
    get session_logout_url
    assert_redirected_to controller: :session, action: :login
  end
end

```

session_controller_test.rb:

```
require 'test_helper'
```

```

class SessionControllerTest < ActionDispatch::IntegrationTest
  test "should get login" do
    get session_login_url
    assert_response :success
  end

  # при выходе редиректит на страницу входу
  test "should get logout" do
    get '/logout'
    assert_redirected_to session_login_url
  end

  # не можем перейти на главную страницу без авторизации
  test "can't calculate without login" do
    get '/'
    assert_redirected_to session_login_url
  end
end

```

user.rb:


```
class User < ActiveRecord::Base
  has_secure_password
  validates :username, presence: true, uniqueness: true
  validates :password_digest, presence: true
end
```

schema.rb:

```
ActiveRecord::Schema[4.2].define(version: 2020_12_12_221716) do
  create_table "users", force: :cascade do |t|
    t.string "username"
    t.string "password_digest"
    t.datetime "last_login_at"
    t.datetime "created_at", precision: 6, null: false
    t.datetime "updated_at", precision: 6, null: false
  end
```

end

Результаты выполнения:

Авторизация:

Session#login

Find me in app/views/session/login.html.erb

<input type="text"/>	<input type="password"/>	<input type="button" value="log in"/>
sign up		

Авторизовались, можно начинать вычисления:

Input

<input type="text"/>	<input type="button" value="Определить наиболее длинную монотонно возрастающую последовательность"/>
Users	
log out	

Вычисления:

Output

Your initial sequence: 123451209346

0: 12345

1: 12

2: 09

3: 346

maximum length rising subsequence: 12345 [Назад](#)

Просмотр пользователей:

Users

Username

userrr [Show](#) [Edit](#) [Destroy](#)

cat [Show](#) [Edit](#) [Destroy](#)

labzuonova [Show](#) [Edit](#) [Destroy](#)

ssssss [Show](#) [Edit](#) [Destroy](#)

user1 [Show](#) [Edit](#) [Destroy](#)

[New User](#)

[START CALCULATING](#)

Настройка данных пользователя:

Editing User

Username

Password

Password confirmation

[Show](#) | [Back](#)

Содержимое бд:

```
[{"id":1,"username":"userrr","password_digest":"$2a$12$1oIjXQU48DBlvxWmDCaP8.RnI/KH0cbKrLwRcXtN0tox0XL.t9Hha","last_login_at":null,"created_at":"2020-12-12T23:20:44.645Z","updated_at":"2020-12-12T23:20:44.645Z"}, {"id":2,"username":"cat","password_digest":"$2a$12$e3dz7n8HbxuC4hs.KpiU8upI4CjKI/YiLYrxm.TYkh3Uv2ws6o40","last_login_at":null,"created_at":"2020-12-13T11:07:18.501Z","updated_at":"2020-12-13T11:11:24.410Z"}, {"id":3,"username":"labzunova","password_digest":"$2a$12$LB0o.eSnLb5Kthnw.GjWGu8LVJeIt5b0SFHT5zT5l2H.meQBtChIO","last_login_at":null,"created_at":"2020-12-13T11:41:57.714Z","updated_at":"2020-12-15T02:28:49.154Z"}, {"id":5,"username":"ssssss","password_digest":"$2a$12$Ru973IgUc6cX63TNkxufFeh2fqqt0ocPCU6p06/V1Tz0iKeQhoSWG","last_login_at":null,"created_at":"2020-12-15T06:03:03.512Z","updated_at":"2020-12-15T06:03:03.512Z"}, {"id":6,"username":"user1","password_digest":"$2a$12$IHPjXNEeD3Y7lmiZKCytNuScXytSBQRe8wMtovyBsmkHZ4fKa5Pi","last_login_at":null,"created_at":"2020-12-15T06:03:26.667Z","updated_at":"2020-12-15T06:03:26.667Z"}]
```

Выведено с помощью добавления в output контроллер

```
def users
  respond_to do |format|
    res = .all
    format.xml { render xml: res }
    format.json { render json: res }
  end
end
```

добавления в пути `get 'usersbd' => 'output#users'`

по запросу `http://127.0.0.1:3000/usersbd.json`

Тесты:

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR12/test$ rake
(in /home/labzunova/Рабочий стол/labs BMSTU/LR12)
Run options: --seed 6387

# Running:

... "redirect"
"redirect"
"Login"
....

Finished in 1.143233s, 6.1230 runs/s, 8.7471 assertions/s.
7 runs, 10 assertions, 0 failures, 0 errors, 0 skips
```

Проверка rubocop и reek:

Rubocop:

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR12/app/controllers$ rubocop
NOTE: Gem.gunzip is deprecated; use Gem::Util.gunzip instead. It will be removed on or after 2018-12-01.
Gem.gunzip called from /usr/lib/ruby/vendor_ruby/unicode/display_width/index.rb:5.
Inspecting 5 files
.....

5 files inspected, no offenses detected
```

Reek:

```
labzunova@labzunova:~/Рабочий стол/labs ВМSTU/LR12/app/controllers$ reek
Inspecting 5 file(s):
.....

0 total warnings
labzunova@labzunova:~/Рабочий стол/labs ВМSTU/LR12/app/controllers$
```