Application\_controller

class ApplicationController < ActionController::Base

before\_filter :set\_session\_locale

before\_action :require\_login

include SessionsHelper

def require\_login

if not current\_register

redirect\_to '/signin'

end

end

def set\_session\_locale

I18n.locale = session[:locale] || I18n.default\_locale

session[:locale] = I18n.locale

end

end

Main\_controller

class MainController < ApplicationController

def set\_locale

session[:locale] = params[:locale]

redirect\_to '/'

end

def index

end

def view

stars = []

if params[:stars]['3'] != '0' then

stars.push(3)

end

if params[:stars]['4'] != '0' then

stars.push(4)

end

if params[:stars]['5'] != '0' then

stars.push(5)

end

if stars == [] then

stars = [3, 4, 5]

end

@mas = Hotel.where(country: params[:country], city: params[:city])

if params[:date1] != '' then

@mas = @mas.where('date1 < ?', params[:date1])

end

if params[:date2] != '' then

@mas = @mas.where('date2 > ?', params[:date2])

end

@mas = @mas.where(stars: stars)

end

end

Registers\_controller

class RegistersController < ApplicationController

before\_action :set\_register, only: [:show, :edit, :update, :destroy]

skip\_before\_action :require\_login, [:new]

def index

require\_login

@registers = Register.all

end

def show

end

def new

@register = Register.new

end

# GET /registers/1/edit

def edit

end

# POST /registers

# POST /registers.json

def create

params = register\_params()

@register = Register.new()

@register.login = params[:login]

@register.password = params[:password]

if @register.save

sign\_in @register

redirect\_to '/'

else

render 'new'

end

end

# PATCH/PUT /registers/1

# PATCH/PUT /registers/1.json

def update

respond\_to do |format|

if @register.update(register\_params)

format.html { redirect\_to @register, notice: 'Register was successfully updated.' }

format.json { head :no\_content }

else

format.html { render action: 'edit' }

format.json { render json: @register.errors, status: :unprocessable\_entity }

end

end

end

# DELETE /registers/1

def destroy

if current\_register

shoot\_foot = (@register.id == current\_register.id)

else

shoot\_foot = false

end

@register.destroy()

if shoot\_foot

sign\_out

redirect\_to '/signin'

else

redirect\_to '/registers'

end

end

private

# Use callbacks to share common setup or constraints between actions.

def set\_register

@register = Register.find(params[:id])

end

# Never trust parameters from the scary internet, only allow the white list through.

def register\_params

params.require(:register).permit(:login, :password)

end

end

Sessions\_controller

class SessionsController < ApplicationController

skip\_before\_action :require\_login, [:new, :create]

def new

if current\_register

redirect\_to '/main/index'

end

end

def create

register = Register.find\_by\_login\_and\_password(params[:session][:login], params[:session][:password])

if register != nil

sign\_in register

redirect\_to register

else

flash[:alert] = 'Invalid email/password combination'

render 'new'

flash[:alert] = ''

end

end

def destroy

sign\_out

redirect\_to '/signin'

end

end

Файл Sessions\_helper.rb

module SessionsHelper

def sign\_in(register)

cookies.permanent[:remember\_token] = register.remember\_token

session[:user] = register

end

def signed\_in?

session[:user] != nil

end

def sign\_out

session[:user] = nil

cookies.delete(:remember\_token)

end

def current\_register

if cookies[:remember\_token]

session[:user] ||= Register.find\_by\_remember\_token(cookies[:remember\_token])

end

session[:user]

end

end