



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 зачетной работе

Дисциплина: Языки интернет-программирования

Тема: Dotop – сервис для обмена дотфайлами

Студент

ИУ6-33Б

(Группа)

(Подпись, дата)

Д. И. Мироненко

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Э. Р. Маняшев

(И.О. Фамилия)

Москва, 2023

Описание

Дотфайл - это скрытый файл в операционных системах Linux и MacOS, имя которого начинается с точки. Эти файлы обычно являются файлами конфигурации для различных программ и скрыты по умолчанию для того, чтобы домашний каталог пользователя оставался чистым и чтобы избежать загромождения конфигурационными файлами. Примерами дотфайлов являются `.bashrc`, `.gitconfig` и `.vimrc`.

Мой сервис обмена такими файлами решает несколько проблем. Во-первых, часто бывает трудно написать свой конфигурационный файл для какой-нибудь программы в силу отсутствия знаний и опыта, поэтому на этом сайте пользователь может изучить конфиги других людей и на их примере написать свой. Во-вторых, бывают случаи, когда нет желания писать свой конфиг, а хочется взять уже готовый и наслаждаться всеми возможностями программы, и в этом тоже поможет мой сайт, на котором пользователь на основании оценок пользователей (голосов) может выбрать лучший или наиболее подходящий для него дотфайл.

Код программы

Файл “app/controllers/application_controller.rb”

```
# frozen_string_literal: true

# Controller for application logic
class ApplicationController < ActionController::Base
  before_action :set_locale

  private

  def after_sign_out_path_for(_resource_or_scope)
    dotfiles_path
  end

  def after_sign_in_path_for(_resource_or_scope)
    dotfiles_path
  end

  def default_url_options
    { locale: I18n.locale }
  end
end
```

```

def set_locale
  I18n.locale = extract_locale || I18n.default_locale
end

def extract_locale
  parsed_locale = params[:locale]
  return unless I18n.available_locales.map(&:to_s).include?(parsed_locale)

  parsed_locale.to_sym
end
end

```

Файл “app/controllers/bookmarks_controller.rb”

```

# frozen_string_literal: true

# Controller for bookmarks logic
class BookmarksController < ApplicationController
  include Pagy::Backend

  before_action :authenticate_user!

  def index
    search_params = params.permit(:format, :page, :commit, :locale,
                                  q: %![title_or_description_or_config_type_or_user_username_cont config_type_eq s]])
    @q = Dotfile.ransack(search_params[:q])
    order_by = get_order_(search_params)
    dotfiles = @q.result(distinct: true).order(order_by => :desc)
    @bookmarks_pagy, @bookmarks = pagy(dotfiles, items: 10)
    rescue Pagy::OverflowError
      redirect_to bookmarks_index_path(page: 1)
    end

    private

    def get_order_(search_params)
      if !search_params[:q].nil? && !search_params[:q][:s].nil?
        order_by = search_params[:q][:s].split[0]
        order_by = (order_by == 'created_at' ? :created_at : :cached_weighted_like_score)
      else
        order_by = :cached_weighted_like_score
      end
      order_by
    end
  end
end

```

Файл “app/controllers/users_controller.rb”

```

# frozen_string_literal: true

# Controller for user profile page
class UsersController < ApplicationController
  def show
    @user = User.find(params[:id])
  end
end

```

Файл “app/controllers/dotfiles_controller.rb”

```
# frozen_string_literal: true

# Controller for dotfiles logic
class DotfilesController < ApplicationController
  include Pagy::Backend

  before_action :set_dotfile, only: %i[show edit update destroy vote bookmark]
  before_action :authenticate_user!, except: %i[index show]
  before_action :correct_user, only: %i[edit update destroy]

  # GET /dotfiles or /dotfiles.json
  def index
    search_params = params.permit(:format, :page, :commit, :locale,
                                   q: %i[title_or_description_or_config_type_or_user_username_cont config_type_eq s])

    @q = Dotfile.ransack(search_params[:q])
    order_by = get_order_(search_params)
    dotfiles = @q.result(distinct: true).order(order_by => :desc)
    @dotfiles_pagy, @dotfiles = pagy(dotfiles, items: 2)
  rescue Pagy::OverflowError
    redirect_to dotfiles_path(page: 1)
  end

  def get_order_(search_params)
    if !search_params[:q].nil? && !search_params[:q][:s].nil?
      order_by = search_params[:q][:s].split[0]
      order_by = (order_by == 'created_at' ? :created_at : :cached_weighted_like_score)
    else
      order_by = :cached_weighted_like_score
    end
  end

  def bookmark
    @dotfile.bookmark!(current_user)
    respond_to do |format|
      format.html do
        redirect_to @dotfile
      end
      format.turbo_stream do
        render turbo_stream: turbo_stream.replace(
          @dotfile, partial: 'dotfiles/dotfile',
          locals: { dotfile: @dotfile, show_config: params[:show_config] }
        )
      end
    end
  end

  def vote
    case params[:type]
    when 'upvote'
      @dotfile.upvote!(current_user)
    when 'downvote'
      @dotfile.downvote!(current_user)
    else
      return redirect_to request.url, alert: 'no such vote type'
    end
  end

  respond_to do |format|
```

```

format.html do
  redirect_to @dotfile
end

format.turbo_stream do
  render turbo_stream: turbo_stream.replace(@dotfile, partial: 'dotfiles/dotfile',
                                             locals: { dotfile: @dotfile, show_config: params[:show_config] })
end
end

# GET /dotfiles/1 or /dotfiles/1.json
def show; end

# GET /dotfiles/new
def new
  @dotfile = current_user.dotfiles.build
end

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

# POST /dotfiles or /dotfiles.json
def create
  @dotfile = current_user.dotfiles.build(dotfile_params)

  respond_to do |format|
    if @dotfile.save
      format.html { redirect_to dotfile_url(@dotfile), notice: I18n.t('notices.create_dotfile') }
      format.json { render :show, status: :created, location: @dotfile }
    else
      format.html { render :new, status: :unprocessable_entity }
      format.json { render json: @dotfile.errors, status: :unprocessable_entity }
    end
  end
end

# PATCH/PUT /dotfiles/1 or /dotfiles/1.json
def update
  respond_to do |format|
    if @dotfile.update(dotfile_params)
      format.html { redirect_to dotfile_url(@dotfile), notice: I18n.t('notices.update_dotfile') }
      format.json { render :show, status: :ok, location: @dotfile }
    else
      format.html { render :edit, status: :unprocessable_entity }
      format.json { render json: @dotfile.errors, status: :unprocessable_entity }
    end
  end
end

# DELETE /dotfiles/1 or /dotfiles/1.json
def destroy
  @dotfile.destroy!

  respond_to do |format|
    format.html { redirect_to dotfiles_url, notice: I18n.t('notices.destroy_dotfile') }
    format.json { head :no_content }
  end
end

```

```

def correct_user
  @dotfile = current_user.dotfiles.find_by(id: params[:id])
  redirect_to dotfiles_path, notice: I18n.t('notices.not_authorized_to_edit') if @dotfile.nil?
end

private

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

# Only allow a list of trusted parameters through.
def dotfile_params
  params.require(:dotfile).permit(:title, :description, :config_type, :file, :user_id)
end
end

```

Файл “app/helpers/application_helper.rb”

```

module ApplicationHelper
  include Pagy::Frontend
end

```

Файл “app/helpers/dotfiles_helper.rb”

```

module DotfilesHelper
  def upvote_label(message, user)
    label_text = if user.voted_up_on? message, vote_scope: "like"
      "UNvote"
    else
      "UPvote"
    end

    tag.span do
      "#{message.cached_scoped_like_votes_up} #{label_text}"
    end
  end

  def downvote_label(message, user)
    label_text = if user.voted_down_on? message, vote_scope: "like"
      "UNvote"
    else
      "DOWNvote"
    end

    tag.span do
      "#{message.cached_scoped_like_votes_down} #{label_text}"
    end
  end

  def upvote_label_styles(dotfile, user)
    if user&.voted_up_on? dotfile, vote_scope: "like"
      "background-color: #e9ecef;"
    end
  end

  def downvote_label_styles(dotfile, user)

```

```

    if user&.voted_down_on? dotfile, vote_scope: "like"
      "background-color: #e9ecef;"
    end
  end

  def bookmark_label_styles(message, user)
    if user.voted_up_on? message, vote_scope: "bookmark"
      "background-color: #e9ecef;"
    end
  end

  def bookmark_label_styles(dotfile, user)
    if user&.voted_up_on? dotfile, vote_scope: "bookmark"
      "background-color: #e9ecef;"
    end
  end
end

```

Файл “app/assets/stylesheets/home.scss”

```

.content {
  position: absolute;
  left: 50%;
  top: 50%;
  -webkit-transform: translate(-50%, -50%);
  transform: translate(-50%, -50%);
}

.header {
  width: 400px;
  margin-left: 100px;
}

.title {
  font-size: 100px;
  font-weight: 700;
}

.subtitle {
  font-size: 24px;
  font-weight: 500;
  color: grey;
  margin-top: -40px;
}

.actions {
  width: 100%;
  display: flex;
  justify-content: center;
  margin-top: 35px;
}

.log-in-btn {
  margin-left: 10px;
}

```

Файл “app/models/dotfile.rb”

```
class Dotfile < ApplicationRecord
  has_one_attached :file
  belongs_to :user

  validates :title, presence: true, format: { with: /\A[a-zA-Z0-9_\.]+\Z/ }
  validates :description, presence: true, format: { with: /\A[a-zA-Z0-9_\.]+\Z/ }
  validates :config_type, presence: true, format: { with: /\A[a-zA-Z_\.]+\Z/ }
  validates :file, presence: true

  validate :allowed_file_extension

  acts_as_votable

  def bookmark!(user)
    if user.voted_up_on? self, vote_scope: "bookmark"
      unvote_by user, vote_scope: "bookmark"
    else
      upvote_by user, vote_scope: "bookmark"
    end
  end

  def upvote!(user)
    if user.voted_up_on? self, vote_scope: "like"
      unvote_by user, vote_scope: "like"
    else
      upvote_by user, vote_scope: "like"
    end
  end

  def downvote!(user)
    if user.voted_down_on? self, vote_scope: "like"
      unvote_by user, vote_scope: "like"
    else
      downvote_by user, vote_scope: "like"
    end
  end

  def allowed_file_extension
    allowed_extensions = %w[fish lua conf]
    if file.present? && !allowed_extensions.include?(File.extname(file.filename.to_s).delete("."))
      errors.add(:file, I18n.t('errors.invalid_extension'))
    end
  end

  def self.ransackable_attributes(auth_object = nil)
    ["title", "description", "config_type", "username"]
  end

  def self.ransackable_associations(auth_object = nil)
    ["user"]
  end
end
```

Файл “app/models/user.rb”

```
class Dotfile < ApplicationRecord
```



```

has_one_attached :file
belongs_to :user

validates :title, presence: true, format: { with: /\A[a-zA-Z0-9_\.s]+\Z/ }
validates :description, presence: true, format: { with: /\A[a-zA-Z0-9_\.s]+\Z/ }
validates :config_type, presence: true, format: { with: /\A[a-zA-Z_\.s]+\Z/ }
validates :file, presence: true

validate :allowed_file_extension

acts_as_votable

def bookmark!(user)
  if user.voted_up_on? self, vote_scope: "bookmark"
    unvote_by user, vote_scope: "bookmark"
  else
    upvote_by user, vote_scope: "bookmark"
  end
end

def upvote!(user)
  if user.voted_up_on? self, vote_scope: "like"
    unvote_by user, vote_scope: "like"
  else
    upvote_by user, vote_scope: "like"
  end
end

def downvote!(user)
  if user.voted_down_on? self, vote_scope: "like"
    unvote_by user, vote_scope: "like"
  else
    downvote_by user, vote_scope: "like"
  end
end

def allowed_file_extension
  allowed_extensions = %w[fish lua conf]
  if file.present? && !allowed_extensions.include?(File.extname(file.filename.to_s).delete("."))
    errors.add(:file, I18n.t('errors.invalid_extension'))
  end
end

def self.ransackable_attributes(auth_object = nil)
  ["title", "description", "config_type", "username"]
end

def self.ransackable_associations(auth_object = nil)
  ["user"]
end
end

```

Файл “config/routes.rb”

```

Rails.application.routes.draw do
  scope "(:locale)", locale: /#{I18n.available_locales.join("|")}/ do
    devise_for :users, controllers: { registrations: 'users/registrations' }
  end
end

```

```
devise_scope :user do
  get '/users/sign_out' => 'devise/sessions#destroy'
end

resources :dotfiles

resources :dotfiles do
  member do
    patch :vote
    patch :bookmark
  end
end

root 'home#index'
get 'bookmarks/index'

resources :users, only: [:show]
end
end
```

Файл “config/locales/en.yml”

```
en:
  global:
    title: "Dotop"
    copyright: "© 2023 Daniil Mironenko"
  activerecord:
    models:
      dotfile: "Dotfile"
    attributes:
      dotfile:
        title: "Title"
        description: "Description"
        config_type: "Config Type"
        file: "File"
  helpers:
    submit:
      create: "Create"
      update: "Apply"
  header:
    explore: "Explore"
    saved: "Saved"
    create: "Create"
    log_in: "Log In"
    profile:
      title: "Profile"
      show: "Profile"
      edit: "Edit"
      log_out: "Log Out"
  home:
    index:
      subtitle: "Share your awesome dotfiles"
  forms:
    searching:
      search: "Search"
      search_placeholder: "Title, description, author, type"
      config_type: "Config Type"
      order_by: "Order"
```

```
votes: "Votes"
recent: "Recent"
search_button: "Search"
creating_dotfile:
  title: "Creating"
editing_dotfile:
  title: "Editing"
  delete: "Delete"
creating_user:
  title: "Creating"
editing_user:
  title: "Editing"
  change_email: "Change Email"
  new_email: "New Email"
  change_password: "Change Password"
  new_password: "New Password"
  confirm_new_password: "Confirm New Password"
  save_changes: "Save Changes"
  current_password: "Current Password"
  apply: "Apply"
  delete: "Delete"
sign_up:
  title: "Sign Up"
  username: "Username"
  email: "Email"
  password: "Password"
  confirm_password: "Confirm Password"
  submit: "Sign Up"
  log_in: "Log In"
log_in:
  title: "Log In"
  email: "Email"
  password: "Password"
  remember_me: "Remember"
  submit: "Log In"
  sign_up: "Sign Up"
user:
  username: "Username"
  email: "Email"
  edit: "Edit"
errors:
  invalid_extension: "has invalid extension"
notices:
  create_dotfile: "Dotfile was successfully created."
  update_dotfile: "Dotfile was successfully updated."
  destroy_dotfile: "Dotfile was successfully destroyed."
  not_authorized_to_edit: "Not authorized to edit this."
```

Файл “config/locales/ru.yml”

```
ru:
  global:
    title: "Dotop"
    copyright: "© 2023 Даниил Мироненко"
  activerecord:
    models:
      dotfile: "Конфиг"
```

```
attributes:
  dotfile:
    title: "Название"
    description: "Описание"
    config_type: "Тип Конфига"
    file: "Файл"
helpers:
  submit:
    create: "Создать"
    update: "Применить"
header:
  explore: "Поиск"
  saved: "Сохраненные"
  create: "Создать"
  log_in: "Войти"
  profile:
    title: "Профиль"
    show: "Профиль"
    edit: "Редактировать"
    log_out: "Выйти"
home:
  index:
    subtitle: "Делитесь ваши потрясающими конфигурами"
forms:
  searching:
    search: "Поиск"
    search_placeholder: "Название, описание, автор, тип"
    config_type: "Тип Конфига"
    order_by: "Сортировать"
    votes: "Голоса"
    recent: "Недавние"
    search_button: "Искать"
  creating_dotfile:
    title: "Создание"
  editing_dotfile:
    title: "Редактирование"
    delete: "Удалить"
  creating_user:
    title: "Создание"
  editing_user:
    title: "Редактирование"
    change_email: "Изменить почту"
    new_email: "Новая почта"
    change_password: "Изменить пароль"
    new_password: "Новый пароль"
    confirm_new_password: "Повторите новый пароль"
    save_changes: "Сохранить изменения"
    current_password: "Текущий пароль"
    apply: "Применить"
    delete: "Удалить"
  sign_up:
    title: "Регистрация"
    username: "Имя пользователя"
    email: "Email"
    password: "Пароль"
    confirm_password: "Повторите пароль"
    submit: "Зарегистрироваться"
    log_in: "Вход"
```

```
log_in:
  title: "Вход"
  email: "Email"
  password: "Пароль"
  remember_me: "Запомнить"
  submit: "Войти"
  sign_up: "Зарегистрироваться"

user:
  username: "Имя пользователя"
  email: "Почта"
  edit: "Изменить"

errors:
  invalid_extension: "имеет недопустимое расширение"

notices:
  create_dotfile: "Конфиг был успешно создан."
  update_dotfile: "Конфиг был успешно изменен."
  destroy_dotfile: "Конфиг был успешно удален."
  not_authorized_to_edit: "Вы не авторизованы для редактирования этого."
```

Файл “config/locales/devise.en.yml”

```
en:
  devise:
    confirmations:
      confirmed: "Your email address has been successfully confirmed."
      send_instructions: "You will receive an email with instructions for how to confirm your email address in a few minutes."
      send_paranoid_instructions: "If your email address exists in our database, you will receive an email with instructions for how to confirm your email address in a few minutes."
    failure:
      already_authenticated: "You are already signed in."
      inactive: "Your account is not activated yet."
      invalid: "Invalid %{authentication_keys} or password."
      locked: "Your account is locked."
      last_attempt: "You have one more attempt before your account is locked."
      not_found_in_database: "Invalid %{authentication_keys} or password."
      timeout: "Your session expired. Please sign in again to continue."
      unauthenticated: "You need to sign in or sign up before continuing."
      unconfirmed: "You have to confirm your email address before continuing."
    mailer:
      confirmation_instructions:
        subject: "Confirmation instructions"
      reset_password_instructions:
        subject: "Reset password instructions"
      unlock_instructions:
        subject: "Unlock instructions"
      email_changed:
        subject: "Email Changed"
      password_change:
        subject: "Password Changed"
    omniauth_callbacks:
      failure: "Could not authenticate you from %{kind} because \" %{reason}\"."
      success: "Successfully authenticated from %{kind} account."
    passwords:
      no_token: "You can't access this page without coming from a password reset email. If you do come from a password reset email, please make sure you used the full URL provided."
      send_instructions: "You will receive an email with instructions on how to reset your password in a few minutes."
      send_paranoid_instructions: "If your email address exists in our database, you will receive a password recovery link at your email address in a few minutes."
      updated: "Your password has been changed successfully. You are now signed in."
```

updated_not_active: "Your password has been changed successfully."

registrations:

destroyed: "Bye! Your account has been successfully cancelled. We hope to see you again soon."

signed_up: "Welcome! You have signed up successfully."

signed_up_but_inactive: "You have signed up successfully. However, we could not sign you in because your account is not yet activated."

signed_up_but_locked: "You have signed up successfully. However, we could not sign you in because your account is locked."

signed_up_but_unconfirmed: "A message with a confirmation link has been sent to your email address. Please follow the link to activate your account."

update_needs_confirmation: "You updated your account successfully, but we need to verify your new email address. Please check your email and follow the confirmation link to confirm your new email address."

updated: "Your account has been updated successfully."

updated_but_not_signed_in: "Your account has been updated successfully, but since your password was changed, you need to sign in again."

sessions:

signed_in: "Signed in successfully."

signed_out: "Signed out successfully."

already_signed_out: "Signed out successfully."

unlocks:

send_instructions: "You will receive an email with instructions for how to unlock your account in a few minutes."

send_paranoid_instructions: "If your account exists, you will receive an email with instructions for how to unlock it in a few minutes."

unlocked: "Your account has been unlocked successfully. Please sign in to continue."

errors:

messages:

already_confirmed: "was already confirmed, please try signing in"

confirmation_period_expired: "needs to be confirmed within %{period}, please request a new one"

expired: "has expired, please request a new one"

not_found: "not found"

not_locked: "was not locked"

not_saved:

one: "1 error prohibited this %{resource} from being saved:"

other: "%{count} errors prohibited this %{resource} from being saved:"

Файл “config/locales/devise.views.ru.yml”

ru:

activerecord:

attributes:

user:

confirmation_sent_at: Дата отправки подтверждения

confirmation_token: Токен подтверждения

confirmed_at: Дата подтверждения

created_at: Дата создания

current_password: Текущий пароль

current_sign_in_at: Дата текущего входа

current_sign_in_ip: IP текущего входа

email: Email

encrypted_password: Зашифрованный пароль

failed_attempts: Неудачные попытки

last_sign_in_at: Дата последнего входа

last_sign_in_ip: IP последнего входа

locked_at: Дата блокировки

password: Пароль

password_confirmation: Подтверждение пароля

remember_created_at: Дата создания запоминания

remember_me: Запомнить меня

reset_password_sent_at: Дата отправки сброса пароля

reset_password_token: Токен сброса пароля

sign_in_count: Количество входов

unconfirmed_email: Неподтвержденный email

unlock_token: Токен разблокировки

updated_at: Дата обновления

models:

user:

few: Пользователя

many: Пользователей

one: Пользователь

other: Пользователи

devise:

confirmations:

confirmed: Ваш адрес эл. почты успешно подтвержден.

new:

resend_confirmation_instructions: Выслать повторно письмо с активацией

send_instructions: В течение нескольких минут вы получите письмо с инструкциями по подтверждению адреса эл. почты.

send_paranoid_instructions: Если ваш адрес эл. почты есть в системе, то в течение нескольких минут вы получите письмо с инструкциями по его подтверждению.

failure:

already_authenticated: Вы уже вошли в систему.

inactive: Ваша учетная запись еще не активирована.

invalid: Неправильный %{authentication_keys} или пароль.

last_attempt: У вас осталась последняя попытка ввода пароля до блокировки учетной записи.

locked: Ваша учетная запись заблокирована.

not_found_in_database: Неправильный %{authentication_keys} или пароль.

timeout: Ваш сеанс закончился. Пожалуйста, войдите в систему снова.

unauthenticated: Вам необходимо войти в систему или зарегистрироваться.

unconfirmed: Вы должны подтвердить вашу учетную запись.

mailer:

confirmation_instructions:

action: Активировать

greeting: Здравствуйте, %{recipient}!

instruction: 'Вы можете активировать свою учетную запись, нажав ссылку снизу:'

subject: Инструкции по подтверждению учетной записи

email_changed:

greeting: Здравствуйте, %{recipient}!

message: Мы пытаемся связаться с вами, что бы сообщить, что ваш email был изменен на %{email}.

message_unconfirmed:

subject: Email изменен

password_change:

greeting: Приветствуем, %{recipient}!

message: Мы пытаемся связаться с вами, что бы сообщить, что ваш пароль был изменен.

subject: Пароль изменен

reset_password_instructions:

action: Изменить пароль

greeting: Здравствуйте, %{recipient}!

instruction: 'Вы (или кто-то еще) запросили изменение пароля. Для изменения пароля нажмите ссылку ниже:'

instruction_2: Если вы не запрашивали изменение пароля - проигнорируйте это сообщение

instruction_3: Ваш пароль не изменится пока вы не нажмете на ссылку и не введете новый.

subject: Инструкции по восстановлению пароля

unlock_instructions:

action: Разблокировать учетную запись

greeting: Здравствуйте, %{recipient}!

instruction: 'Нажмите ссылку для активации учетной записи:'

message: Ваша учетная запись была заблокирована в связи с превышением лимита неудачных попыток входа.

subject: Инструкции по разблокировке учетной записи

omniauth_callbacks:

failure: Вы не можете войти в систему с учетной записью из %{kind}, т.к. "%{reason}".

success: Вход в систему выполнен с учетной записью из %{kind}.

passwords:

edit:

change_my_password: Изменить мой пароль

change_your_password: Изменить пароль

confirm_new_password: Повторите новый пароль

new_password: Новый пароль

new:

forgot_your_password: Забыли пароль?

send_me_reset_password_instructions: Выслать новый пароль

no_token: Доступ к этой странице возможен только по ссылке из письма о восстановлении пароля. Если вы пришли по такой ссылке, пожалуйста, убедитесь, что Вы скопировали всю ссылку целиком.

send_instructions: В течение нескольких минут вы получите письмо с инструкциями по восстановлению пароля.

send_paranoid_instructions: Если ваш адрес эл. почты есть в системе, то в течение нескольких минут вы получите письмо с инструкциями по восстановлению пароля.

updated: Ваш пароль изменен. Теперь вы вошли в систему.

updated_not_active: Ваш пароль изменен.

registrations:

destroyed: До свидания! Ваша учетная запись удалена. Надеемся снова увидеть вас.

edit:

are_you_sure: Вы уверены?

cancel_my_account: Удалить учетную запись

currently_waiting_confirmation_for_email: 'Ожидается подтверждение адреса E-mail: %{email}'

leave_blank_if_you_don_t_want_to_change_it: оставьте поле пустым, если не хотите его менять

title: Редактировать %{resource}

unhappy: Разочарованы?

update: Обновить

we_need_your_current_password_to_confirm_your_changes: введите текущий пароль для подтверждения изменений

new:

sign_up: Регистрация

signed_up: Добро пожаловать! Вы успешно зарегистрировались.

signed_up_but_inactive: Вы успешно зарегистрированы. Однако, вы не можете войти в систему, потому что ваша учетная запись не активирована.

signed_up_but_locked: Вы успешно зарегистрированы. Однако, вы не можете войти в систему, потому что ваша учетная запись заблокирована.

signed_up_but_unconfirmed: Письмо со ссылкой для подтверждения было отправлено на ваш адрес эл. почты. Пожалуйста, перейдите по ссылке, чтобы подтвердить учетную запись.

update_needs_confirmation: Вы успешно обновили данные учетной записи, но необходимо подтвердить новый адрес эл. почты. Пожалуйста, проверьте почтовый ящик и перейдите по ссылке, чтобы закончить процедуру проверки нового адреса эл. почты.

updated: Ваша учетная запись изменена.

updated_but_not_signed_in: Ваш аккаунт успешно обновлен, но так как пароль был изменен, необходимо выполнить вход еще раз

sessions:

already_signed_out: Выход из системы уже выполнен.

new:

sign_in: Войти

signed_in: Вход в систему выполнен.

signed_out: Выход из системы выполнен.

shared:

links:

back: Назад

didn_t_receive_confirmation_instructions: Не получили подтверждение?

didn_t_receive_unlock_instructions: Не получили код разблокировки?

forgot_your_password: Забыли пароль?

sign_in: Войти

sign_in_with_provider: 'Войти с помощью: %{provider}'

sign_up: Регистрация

minimum_password_length:

few: "(минимум %{count} знака)"

many: "(минимум %{count} знаков)"

one: "(минимум %{count} знак)"

other: "(минимум %{count} знаков)"

unlocks:

new:

resend_unlock_instructions: Выслать подтверждение заново

send_instructions: В течение нескольких минут вы получите письмо с инструкциями по разблокировке учетной записи.

send_paranoid_instructions: Если ваша учетная запись существует, то в течение нескольких минут вы получите письмо с инструкциями по ее разблокировке.

unlocked: Ваша учетная запись разблокирована. Теперь вы можете войти в систему.

errors:

messages:

already_confirmed: уже подтверждён. Пожалуйста, попробуйте войти в систему

confirmation_period_expired: должен быть подтвержден в течение %{period}, пожалуйста, повторите запрос на подтверждение

expired: устарела. Пожалуйста, запросите новую

not_found: не найден

not_locked: не заблокирован

not_saved:

few: "Сохранение не удалось из-за %{count} ошибок:"

many: "%{resource}: сохранение не удалось из-за %{count} ошибок:"

one: "%{resource}: сохранение не удалось из-за %{count} ошибки:"

other: "%{resource}: сохранение не удалось из-за %{count} ошибки:"

Файл “db/schema.rb”

```
ActiveRecord::Schema[7.1].define(version: 2023_12_08_161150) do
  create_table "active_storage_attachments", force: :cascade do |t|
    t.string "name", null: false
    t.string "record_type", null: false
    t.bigint "record_id", null: false
    t.bigint "blob_id", null: false
    t.datetime "created_at", null: false
    t.index ["blob_id"], name: "index_active_storage_attachments_on_blob_id"
    t.index ["record_type", "record_id", "name", "blob_id"], name: "index_active_storage_attachments_uniqueness", unique: true
  end

  create_table "active_storage_blobs", force: :cascade do |t|
    t.string "key", null: false
    t.string "filename", null: false
    t.string "content_type"
    t.text "metadata"
    t.string "service_name", null: false
    t.bigint "byte_size", null: false
    t.string "checksum"
    t.datetime "created_at", null: false
    t.index ["key"], name: "index_active_storage_blobs_on_key", unique: true
  end

  create_table "active_storage_variant_records", force: :cascade do |t|
    t.bigint "blob_id", null: false
    t.string "variation_digest", null: false
    t.index ["blob_id", "variation_digest"], name: "index_active_storage_variant_records_uniqueness", unique: true
  end

  create_table "dotfiles", force: :cascade do |t|
    t.string "title"
    t.text "description"
    t.string "config_type"
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
    t.integer "user_id"
    t.integer "cached_scoped_like_votes_total", default: 0
    t.integer "cached_scoped_like_votes_score", default: 0
  end
```

```

t.integer "cached_scoped_like_votes_up", default: 0
t.integer "cached_scoped_like_votes_down", default: 0
t.integer "cached_weighted_like_score", default: 0
t.integer "cached_weighted_like_total", default: 0
t.float "cached_weighted_like_average", default: 0.0
t.integer "cached_scoped_bookmark_votes_total", default: 0
t.integer "cached_scoped_bookmark_votes_score", default: 0
t.integer "cached_scoped_bookmark_votes_up", default: 0
t.integer "cached_scoped_bookmark_votes_down", default: 0
t.integer "cached_weighted_bookmark_score", default: 0
t.integer "cached_weighted_bookmark_total", default: 0
t.float "cached_weighted_bookmark_average", default: 0.0
t.index ["user_id"], name: "index_dotfiles_on_user_id"
end

create_table "users", force: :cascade do |t|
  t.string "email", default: "", null: false
  t.string "encrypted_password", default: "", null: false
  t.string "reset_password_token"
  t.datetime "reset_password_sent_at"
  t.datetime "remember_created_at"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.string "username"
  t.index ["email"], name: "index_users_on_email", unique: true
  t.index ["reset_password_token"], name: "index_users_on_reset_password_token", unique: true
  t.index ["username"], name: "index_users_on_username", unique: true
end

create_table "votes", force: :cascade do |t|
  t.string "votable_type"
  t.integer "votable_id"
  t.string "voter_type"
  t.integer "voter_id"
  t.boolean "vote_flag"
  t.string "vote_scope"
  t.integer "vote_weight"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.index ["votable_id", "votable_type", "vote_scope"], name: "index_votes_on_votable_id_and_votable_type_and_vote_scope"
  t.index ["votable_type", "votable_id"], name: "index_votes_on_votable"
  t.index ["voter_id", "voter_type", "vote_scope"], name: "index_votes_on_voter_id_and_voter_type_and_vote_scope"
  t.index ["voter_type", "voter_id"], name: "index_votes_on_voter"
end

add_foreign_key "active_storage_attachments", "active_storage_blobs", column: "blob_id"
add_foreign_key "active_storage_variant_records", "active_storage_blobs", column: "blob_id"
end

```

Файл “spec/shared_contexts.rb”

```

RSpec.shared_context "api request authentication helper methods" do
  def sign_in(user)
    login_as(user, scope: :user)
  end

  def sign_out
    logout(:user)
  end
end

```

```
end
end
```

Файл “spec/factories/dotfile.rb”

```
FactoryBot.define do
  factory :dotfile do
    title { Faker::Lorem.sentence(word_count: 3) }
    description { Faker::Lorem.paragraph(sentence_count: 3) }
    config_type { "fish" }
    user { FactoryBot.create(:user) }
    file { Rack::Test::UploadedFile.new('spec/fixtures/config.fish', 'text/fish') }
  end
end
```

Файл “spec/factories/user.rb”

```
FactoryBot.define do
  password = Faker::Internet.password

  factory :user do
    username { Faker::Internet.username(specifier: 6..10) }
    email { Faker::Internet.email }
    password { password }
    password_confirmation { password }
  end
end
```

Файл “spec/unit/dotfile_spec.rb”

```
require 'rails_helper'

RSpec.describe Dotfile, type: :model do
  before(:each) do
    @user = User.create!(
      username: Faker::Internet.username(specifier: 6..10),
      email: Faker::Internet.email,
      password: "password",
      encrypted_password: "password"
    )
  end

  subject do
    described_class.new(
      title: Faker::Lorem.sentence(word_count: 3),
      description: Faker::Lorem.paragraph(sentence_count: 3),
      config_type: "fish",
      user_id: @user.id,
      file: fixture_file_upload(Rails.root.join('spec', 'fixtures', 'config.fish'), 'text/plain')
    )
  end

  it "is valid with valid attributes" do
    expect(subject).to be_valid
  end

  it "is not valid when no title is entered" do

```

```

    subject.title = nil
    expect(subject).not_to be_valid
  end

  it "is not valid when no description is entered" do
    subject.description = nil
    expect(subject).not_to be_valid
  end

  it "is not valid when no config_type is provided" do
    subject.config_type = nil
    expect(subject).not_to be_valid
  end

  it "is not valid when no user_id is given" do
    subject.user_id = nil
    expect(subject).not_to be_valid
  end

  it "should belong to a user" do
    expect(subject.user).to eq(@user)
  end

  it "is not valid when title has invalid symbols" do
    subject.title = "title&"
    expect(subject).not_to be_valid
  end

  it "is not valid when description has invalid symbols" do
    subject.description = "description*"
    expect(subject).not_to be_valid
  end
end

```

Файл "spec/unit/user_spec.rb"

```

require 'rails_helper'

RSpec.describe User, type: :model do
  subject do
    described_class.new(
      username: Faker::Internet.username(specifier: 6..10),
      email: Faker::Internet.email,
      password: "password",
      encrypted_password: "password"
    )
  end

  it "is valid with valid attributes" do
    expect(subject).to be_valid
  end

  it "is not valid when email is not entered" do
    subject.email = nil
    expect(subject).not_to be_valid
  end

  it "is not valid when username is not entered" do

```

```

    subject.username = nil
    expect(subject).not_to be_valid
end

it "is valid when encrypted password is not entered" do
    subject.encrypted_password = nil
    expect(subject).to be_valid
end

it "is not valid when username has spaces" do
    subject.username = "user name"
    expect(subject).not_to be_valid
end

it "is not valid when username has invalid symbols" do
    subject.username = "username^"
    expect(subject).not_to be_valid
end

it "is not valid when email has invalid format" do
    subject.email = "email"
    expect(subject).not_to be_valid
end

it "is not valid when email has invalid symbols" do
    subject.email = "ma&l@gmail.com"
    expect(subject).not_to be_valid
end
end

```

Файл “spec/integration/dotfile_spec.rb”

```

require 'rails_helper'
require 'shared_contexts'

RSpec.describe "Dotfile Tests", type: :feature do
    include_context "api request authentication helper methods"

    let(:user) { FactoryBot.create(:user) }
    let(:dotfile) { FactoryBot.create(:dotfile, user: user) }

    before do
        sign_in user
    end

    scenario "create new dotfile" do
        visit new_dotfile_path

        expect(page).to have_content("Creating")

        title = Faker::Lorem.sentence(word_count: 3)
        description = Faker::Lorem.paragraph(sentence_count: 3)

        find("#title").set(title)
        find("#description").set(description)
        find("#config_type").select("fish")
        find("#file").attach_file("spec/fixtures/config.fish")
    end
end

```

```

find("#create_button").click

expect(page).to have_content("Dotfile was successfully created.")
expect(page).to have_content(title)
expect(page).to have_content(description)
expect(page).to have_content("fish")
end

scenario "edit title" do
  visit edit_dotfile_path(id: dotfile.id)
  expect(page).to have_content("Editing")

  title = Faker::Lorem.sentence(word_count: 3)
  find("#title").set(title)

  click_on "Apply"

  expect(page).to have_content("Dotfile was successfully updated.")
  expect(page).to have_content(title)
end

scenario "edit description" do
  visit edit_dotfile_path(id: dotfile.id)
  expect(page).to have_content("Editing")

  description = Faker::Lorem.paragraph(sentence_count: 3)
  find("#description").set(description)

  click_on "Apply"

  expect(page).to have_content("Dotfile was successfully updated.")
  expect(page).to have_content(description)
end

scenario "edit config type" do
  visit edit_dotfile_path(id: dotfile.id)
  expect(page).to have_content("Editing")

  config_type = "tmux"
  find("#config_type").select(config_type)

  click_on "Apply"

  expect(page).to have_content("Dotfile was successfully updated.")
  expect(page).to have_content(config_type)
end
end

```

Файл "spec/integration/user_spec.rb"

```

require 'rails_helper'
require 'shared_contexts'

RSpec.describe "User Tests", type: :feature do
  include_context "api request authentication helper methods"

  let(:user) { FactoryBot.create(:user) }

```

```
scenario "valid sign up" do
  visit new_user_registration_path

  fill_in "Username", with: "#{Faker::Internet.username(specifier: 6..10)}"
  fill_in "Email", with: "#{Faker::Internet.email}"

  password = Faker::Internet.password
  fill_in "Password", with: "#{password}"
  fill_in "Confirm Password", with: "#{password}"

  click_on "Sign Up"

  expect(page).to have_content("Welcome! You have signed up successfully.")
end

scenario "valid log in" do
  visit new_user_session_path

  fill_in "Email", with: user.email
  fill_in "Password", with: user.password

  click_on "Log In"

  expect(page).to have_content("Signed in successfully.")
end

scenario "change user email" do
  sign_in user

  visit edit_user_registration_path

  fill_in "New Email", with: "#{Faker::Internet.email}"
  fill_in "Current Password", with: "#{user.password}"

  click_on "Apply"

  visit dotfiles_path

  expect(page).to have_content("Your account has been updated successfully.")
end

scenario "change user password" do
  sign_in user

  visit edit_user_registration_path

  new_password = Faker::Internet.password
  fill_in "New Password", with: "#{new_password}"
  fill_in "Confirm New Password", with: "#{new_password}"
  fill_in "Current Password", with: "#{user.password}"

  click_on "Apply"

  visit dotfiles_path

  expect(page).to have_content("Your account has been updated successfully.")
end
```

Файл “Gemfile”

```
source "https://rubygems.org"

ruby "3.2.2"

gem 'sassc-rails'

# Bundle edge Rails instead: gem "rails", github: "rails/rails", branch: "main"
gem "rails", "~> 7.1.1"

gem 'rails-i18n'

# The original asset pipeline for Rails [https://github.com/rails/sprockets-rails]
gem "sprockets-rails"

# Use sqlite3 as the database for Active Record
gem "sqlite3", "~> 1.4"

# Use the Puma web server [https://github.com/puma/puma]
gem "puma", ">= 5.0"

# Use JavaScript with ESM import maps [https://github.com/rails/importmap-rails]
gem "importmap-rails"

# Hotwire's SPA-like page accelerator [https://turbo.hotwired.dev]
gem "turbo-rails"

# Hotwire's modest JavaScript framework [https://stimulus.hotwired.dev]
gem "stimulus-rails"

# Build JSON APIs with ease [https://github.com/rails/jbuilder]
gem "jbuilder"

# Use Redis adapter to run Action Cable in production
# gem "redis", ">= 4.0.1"

# Use Kredis to get higher-level data types in Redis [https://github.com/rails/kredis]
# gem "kredis"

# Use Active Model has_secure_password [https://guides.rubyonrails.org/active_model_basics.html#securepassword]
# gem "bcrypt", "~> 3.1.7"

# Windows does not include zoneinfo files, so bundle the tzinfo-data gem
gem "tzinfo-data", platforms: %i[ windows jruby ]

# Reduces boot times through caching; required in config/boot.rb
gem "bootsnap", require: false

# Use Active Storage variants [https://guides.rubyonrails.org/active_storage_overview.html#transforming-images]
# gem "image_processing", "~> 1.2"

gem 'activestorage', '~> 7.1', '>= 7.1.1'

gem 'devise', '~> 4.9', '>= 4.9.3'
gem 'devise-i18n', '~> 1.12.0', ">= 1.12.0"
```



```
group :development, :test do
  # See https://guides.rubyonrails.org/debugging_rails_applications.html#debugging-with-the-debug-gem
  gem "debug", platforms: %i[ mri windows ]
end

group :development do
  # Use console on exceptions pages [https://github.com/rails/web-console]
  gem "web-console"

  # Add speed badges [https://github.com/MiniProfiler/rack-mini-profiler]
  # gem "rack-mini-profiler"

  # Speed up commands on slow machines / big apps [https://github.com/rails/spring]
  # gem "spring"
end

group :test do
  # Use system testing [https://guides.rubyonrails.org/testing.html#system-testing]
  gem "capybara"
  gem "selenium-webdriver"
  gem "rspec-rails"
  gem "factory_bot_rails"
  gem "faker"
end

gem "ransack", "~> 4.1"

gem "pagy", "~> 6.1"

gem "acts_as_votable", "~> 0.14.0"
```

Работа программы

Dotop

Share your awesome dotfiles

Explore

Log In

Рисунок 1 — домашняя страница

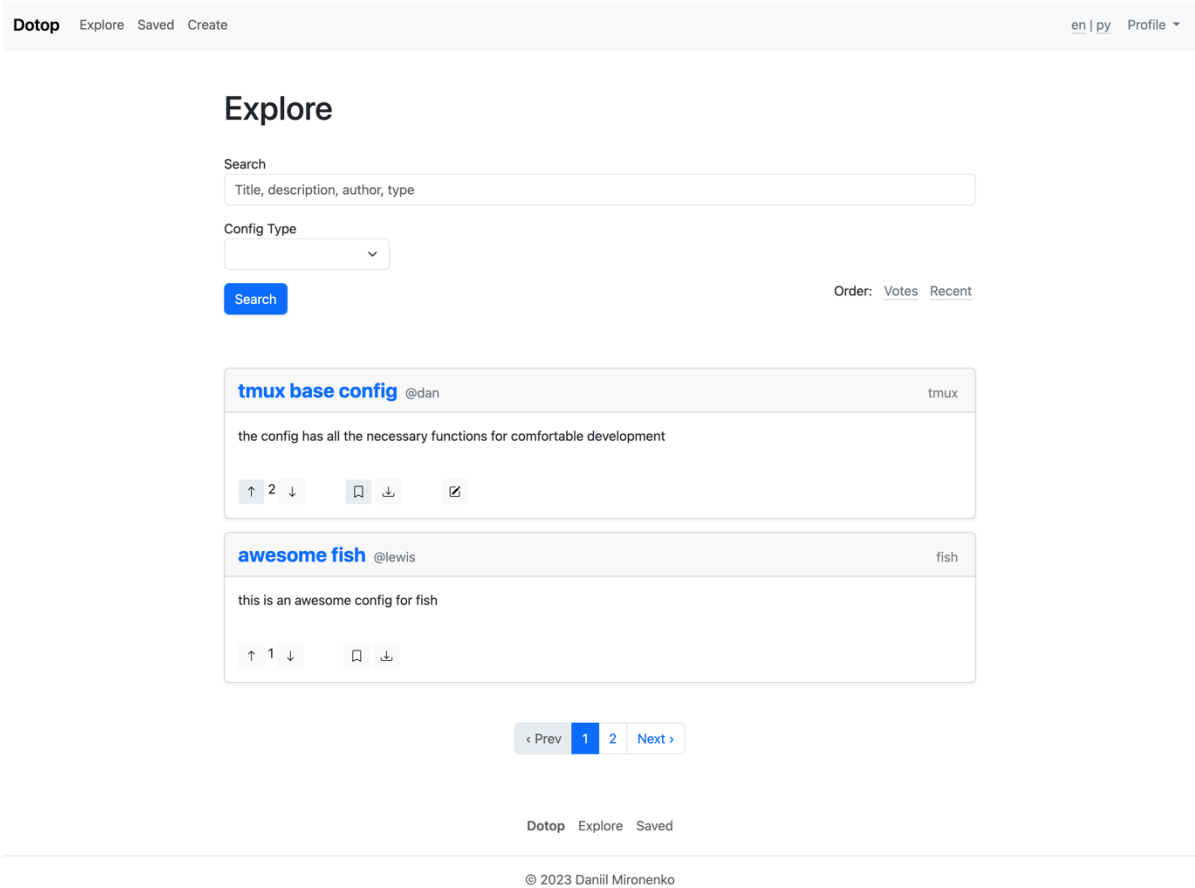


Рисунок 2 – страница просмотра конфигов

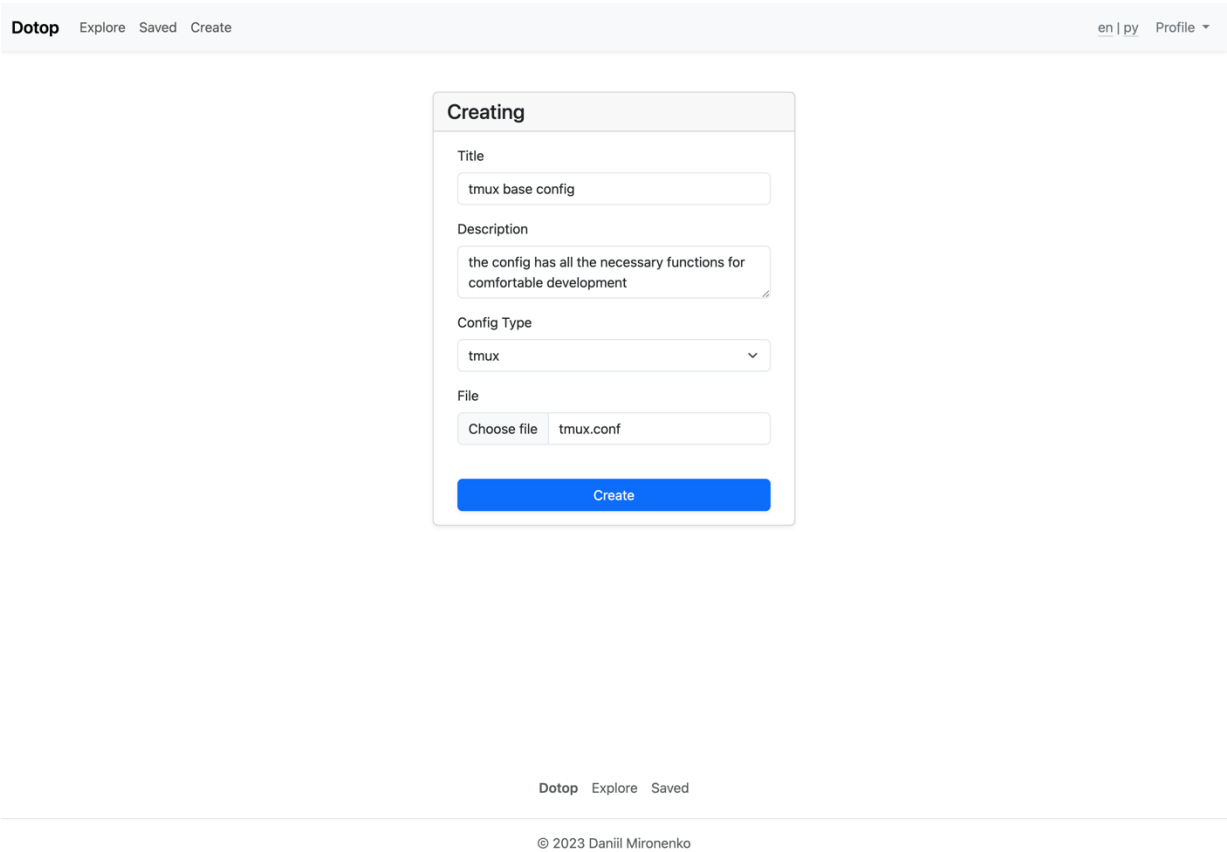


Рисунок 3 – создание конфига

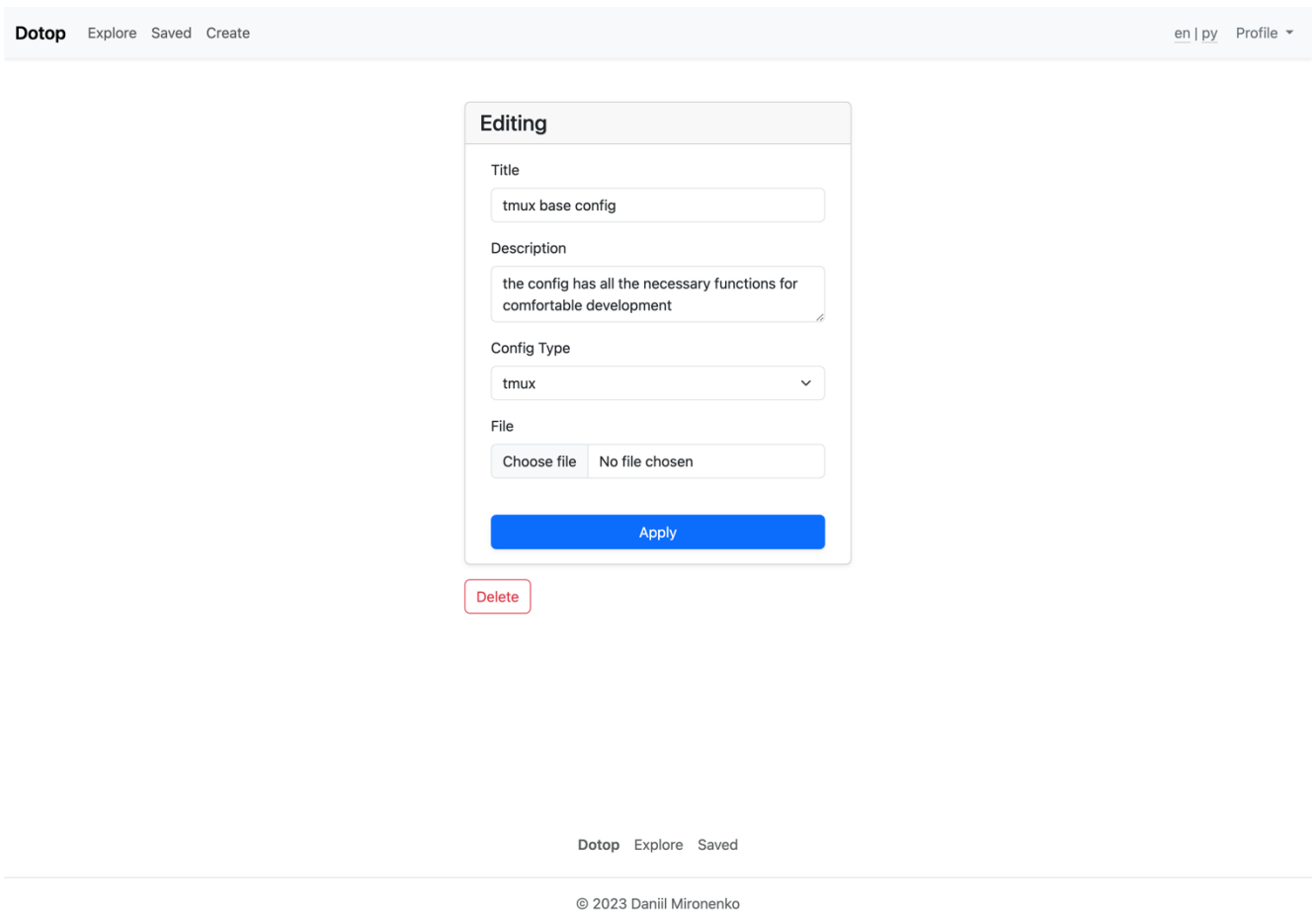


Рисунок 4 – редактирование конфига

Sign Up

Username
dan

Email
daronenko@yandex.ru

Password

Confirm Password

Sign Up

[Log In](#)

Рисунок 5 – регистрация аккаунта

Log In

Email

daronenko@yandex.ru

Password

.....

☒ Remember

Log In

[Sign Up](#)

Рисунок 6 – вход в аккаунт

Dotop Explore Saved Create

en | py Profile

@dan

Email: daronenko@yandex.ru

Edit

Dotop Explore Saved

© 2023 Daniil Mironenko

Рисунок 7 – просмотр профиля

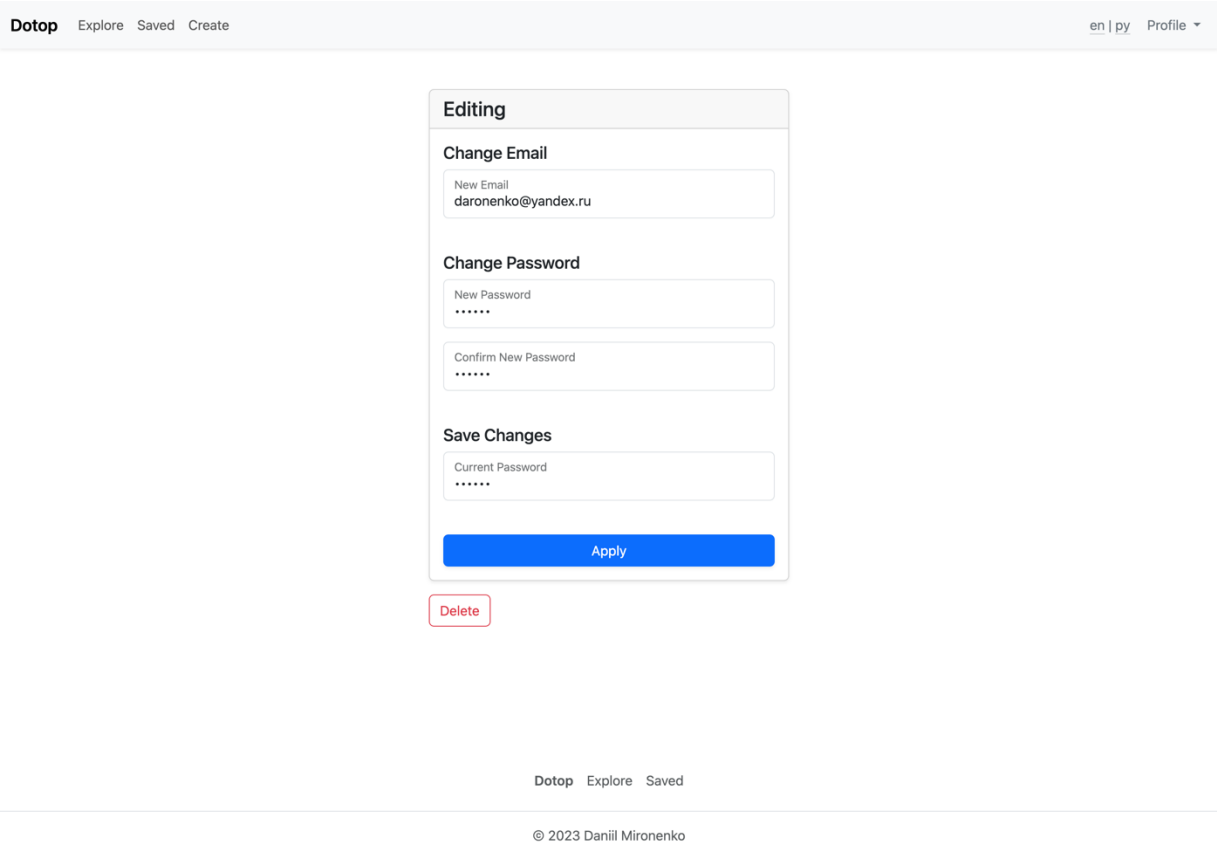


Рисунок 8 – редактирование профиля

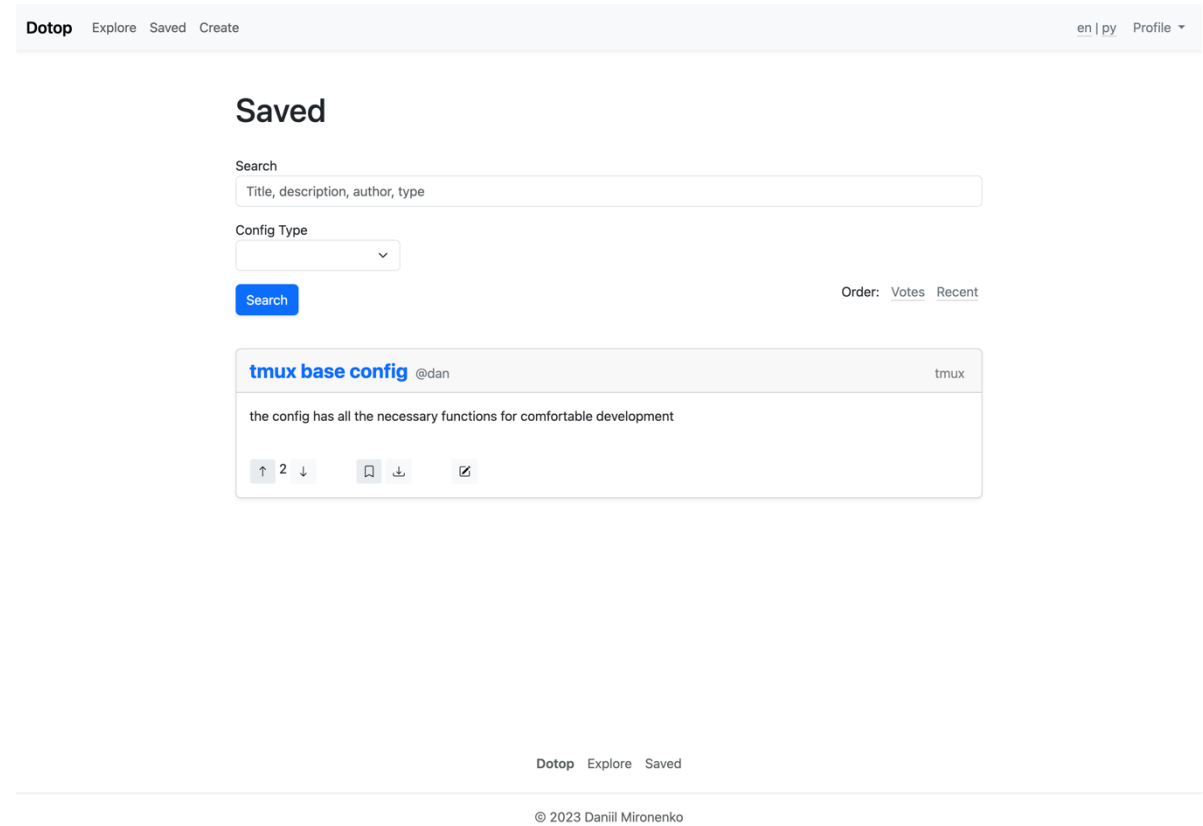


Рисунок 9 – страница сохраненных конфигов

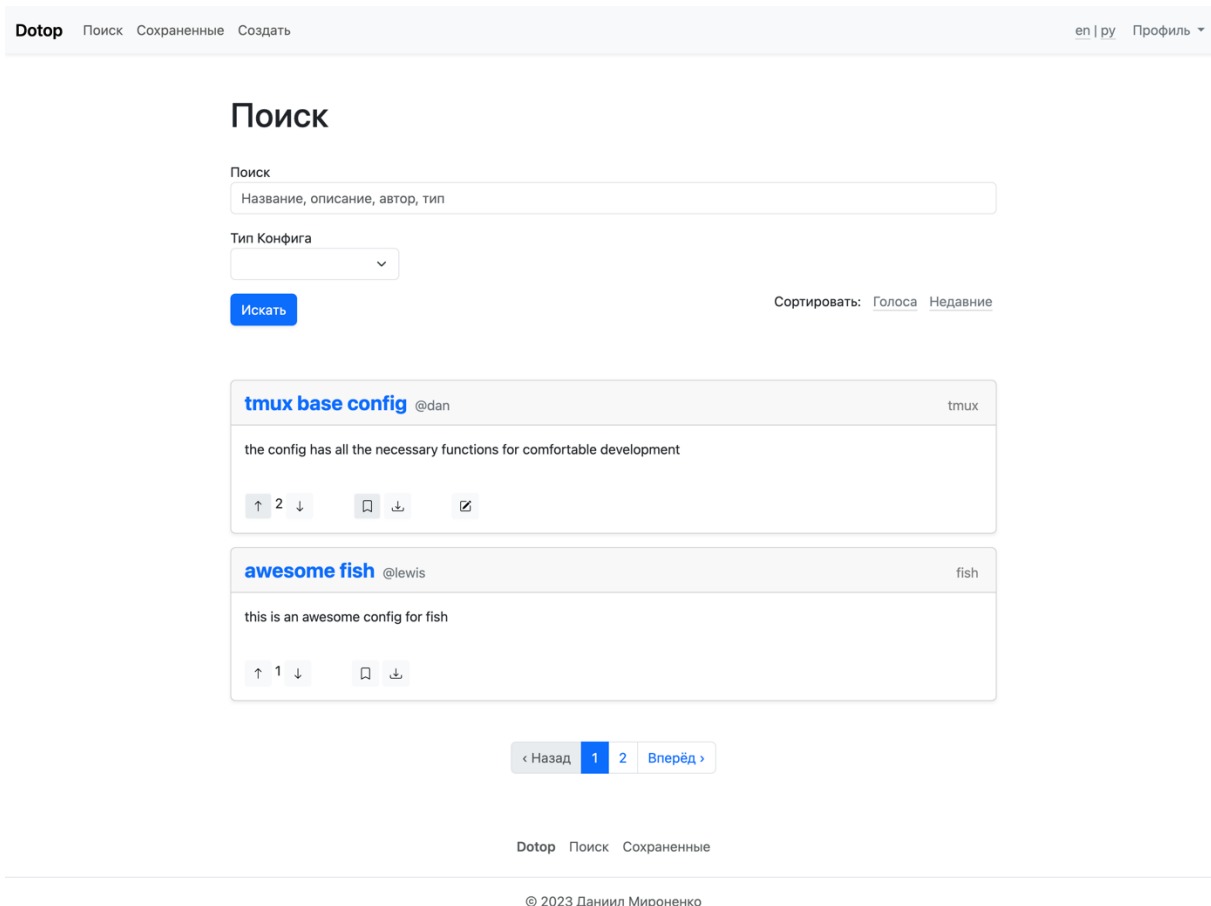


Рисунок 10 – смена локали сайта

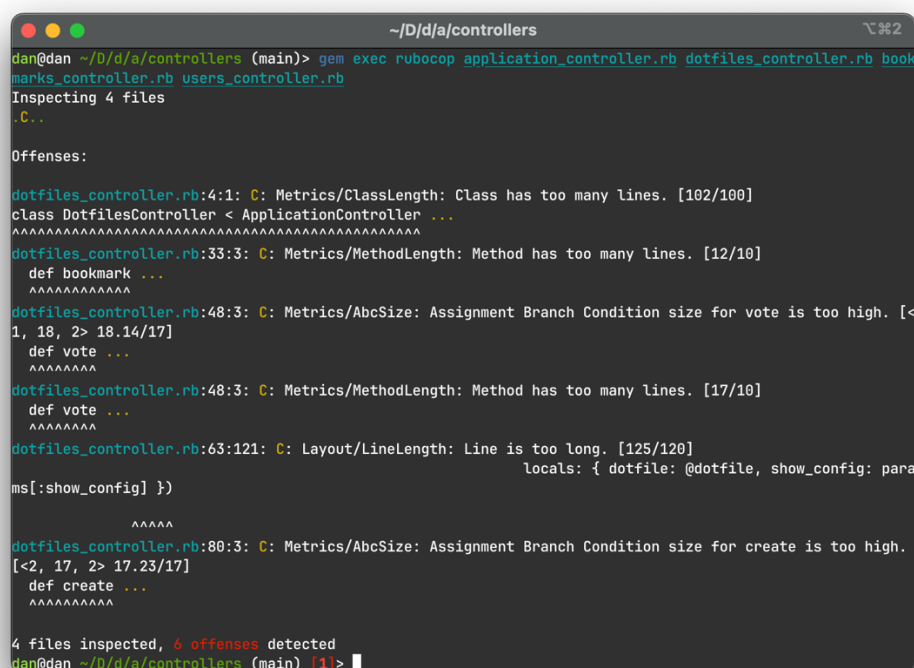


Рисунок 11 – проверка исходного кода контроллеров rubocop'ом

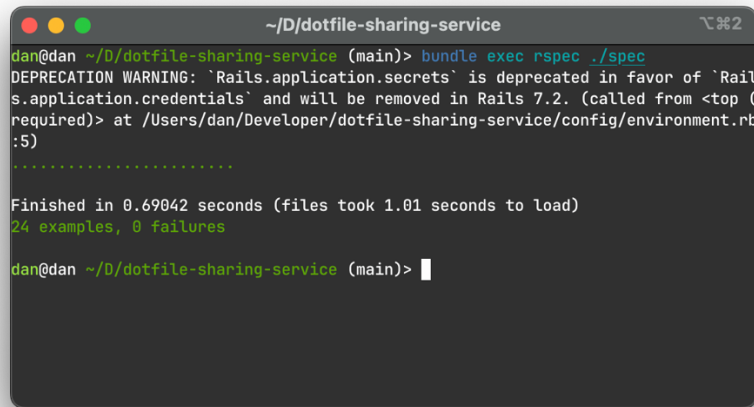


Рисунок 12 — выполнение тестов

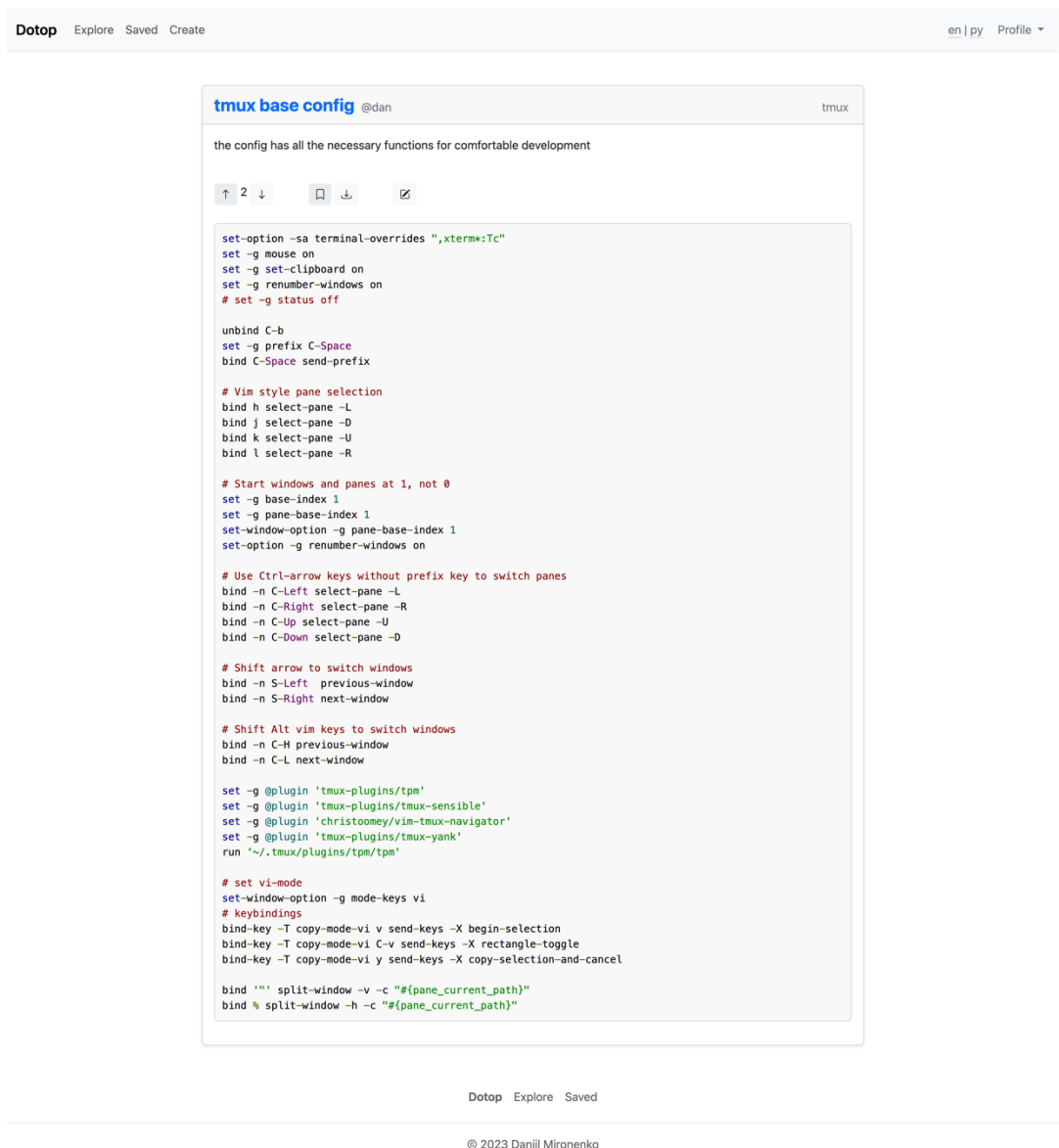


Рисунок 13 — просмотр конфига

Вывод

Разработан и реализован сайт, удовлетворяющий условиям задания, а также написаны тесты (юнит-тесты и интеграционные тесты) для проверки функционала и корректной работы приложения; проведена проверка исходного кода с использованием линтера “rubocop” для обеспечения соблюдения стандартов кода.