



Бесплатная электронная книга

УЧУСЬ

devise

Free unaffiliated eBook created from
Stack Overflow contributors.

#devise

.....	1
1:	2
.....	2
Examples.....	2
.....	2
1. Devise Gem	2
2.	2
3.	2
4.	3
5.	3
2: Devise & OmniAuth	4
Examples.....	4
.....	4
/omniauth_callbacks_controller.rb.....	4
omnia_concern.rb.....	5
authentication_provider.....	5
user_authentication.....	6
authentication_provider.rb.....	6
user_authentication.rb.....	6
user.rb.....	6
omniauth_attributes_concern.rb.....	7
/omniauth_callbacks_controller.rb routes.rb.....	8
devise.rb.....	8
Gems Gemfile OmniAuth.....	8
.....	9

Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [devise](#)

It is an unofficial and free devise ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official devise.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

глава 1: Начало работы с разработкой

замечания

В этом разделе представлен обзор того, что такое разработка, и почему разработчик может захотеть его использовать.

Следует также упомянуть о любых крупных предметах в рамках разработки, а также увязать их с соответствующими темами. Поскольку документация для разработки является новой, вам может потребоваться создать начальные версии этих связанных тем.

Examples

Установка или настройка

1. Добавить Devise Gem

Откройте свой **Gemfile** и добавьте эту строку

```
gem 'devise'
```

Затем запустите;

```
bundle install
```

2. Настройте приложение в своем приложении.

Выполните следующую команду в терминале

```
rails g devise:install
```

3. Конфигурирование

Убедитесь, что вы указали параметры URL-адреса по умолчанию в файлах среды. Откройте *config / environment / development.rb* и добавьте эту строку:

```
config.action_mailer.default_url_options = { host: 'localhost', port: 3000 }
```

Откройте *приложение* / *views* / *layouts* / *application.html.erb* и добавьте:

```
<% if notice %>
  <p class="alert alert-success"><%= notice %></p>
<% end %>
<% if alert %>
  <p class="alert alert-danger"><%= alert %></p>
<% end %>
```

прямо над

```
<%= yield %>
```

Откройте *приложение* / *views* / *ideas* / *show.html.erb* и удалите строку, которая гласит:

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

- Сделайте то же самое для *app* / *views* / *comments* / *show.html.erb* . Эти строки не нужны, поскольку мы разместили уведомление в файле *app* / *views* / *layouts* / *application.html.erb* .

4. Настройте модель пользователя

Для создания модели User мы будем использовать сценарий с объединенным генератором.

```
rails g devise user
rake db:migrate
```

Тренер: Объясните, какая модель пользователя была сгенерирована. Какие поля?

5. Создайте своего первого пользователя

Теперь, когда вы настроили все, вы можете создать своего первого пользователя. Разработчик создает все коды и маршруты, необходимые для создания учетных записей, входа в систему, выхода из системы и т. Д.

Убедитесь, что сервер rails запущен, откройте http://localhost:3000/users/sign_up и создайте свою учетную запись пользователя.

Прочитайте Начало работы с разработкой онлайн: <https://riptutorial.com/ru/devise/topic/3561/начало-работы-с-разработкой>

глава 2: Многопользовательские устройства Devise & OmniAuth

Examples

Добавьте и установите драгоценный камень

Измените свой `Gemfile` чтобы включить драгоценный камень Devise:

```
gem 'devise'
```

Затем обновите свои драгоценные камни с помощью:

```
$ bundle install
```

Запустите установщиков в вашем проекте:

```
$ rails generate devise:install
```

Затем создайте свою модель Devise (например, `User`) с помощью:

```
$ rails generate devise User
```

вам нужно настроить параметры URL-адреса по умолчанию для почтовой программы Devise в файле `config.conf.rb` среды `development.rb`:

```
config.action_mailer.default_url_options = { host: 'localhost', port: 3000 }
```

Devise вносит некоторые изменения в вашу базу данных, поэтому вам нужно запустить созданные миграции:

```
$ rake db:migrate
```

Создание пользователей контроллера / `omniauth_callbacks_controller.rb`

```
class Users::OmniauthCallbacksController < Devise::OmniauthCallbacksController
  include OmniConcern
  %w[facebook twitter gplus linkedin].each do |meth|
    define_method(meth) do
      create
    end
  end
end
```

Примечание. В части « %w[facebook twitter gplus linkedin] » вы должны перечислить все социальные учетные записи, которые вы будете использовать.

Создание Консоли контроллера `omni_concern.rb`

```
module OmniConcern
  extend ActiveSupport::Concern
  def create
    auth_params = request.env["omniauth.auth"]
    provider = AuthenticationProvider.get_provider_name(auth_params.try(:provider)).first
    authentication = provider.user_authentications.where(uid: auth_params.uid).first
    existing_user = User.where('email = ?', auth_params['info']['email']).try(:first)
    if user_signed_in?
      SocialAccount.get_provider_account(current_user.id, provider.id).first_or_create(user_id:
current_user.id , authentication_provider_id: provider.id , token:
auth_params.try(:[], "credentials").try(:[], "token") , secret:
auth_params.try(:[], "credentials").try(:[], "secret"))
      redirect_to new_user_registration_url
    elsif authentication
      create_authentication_and_sign_in(auth_params, existing_user, provider)
    else
      create_user_and_authentication_and_sign_in(auth_params, provider)
    end
  end
  def sign_in_with_existing_authentication(authentication)
    sign_in_and_redirect(:user, authentication.user)
  end
  def create_authentication_and_sign_in(auth_params, user, provider)
    UserAuthentication.create_from_omniauth(auth_params, user, provider)
    sign_in_and_redirect(:user, user)
  end
  def create_user_and_authentication_and_sign_in(auth_params, provider)
    user = User.create_from_omniauth(auth_params)
    if user.valid?
      create_authentication_and_sign_in(auth_params, user, provider)
    else
      flash[:error] = user.errors.full_messages.first
      redirect_to new_user_registration_url
    end
  end
end
```

Добавить миграцию `authentication_provider`

```
class CreateAuthenticationProviders < ActiveRecord::Migration
  def change
    create_table "authentication_providers", :force => true do |t|
      t.string "name"
      t.datetime "created_at", :null => false
      t.datetime "updated_at", :null => false
    end
    add_index "authentication_providers", ["name"], :name =>
"index_name_on_authentication_providers"
    AuthenticationProvider.create(name: 'facebook')
    AuthenticationProvider.create(name: 'twitter')
    AuthenticationProvider.create(name: 'gplus')
    AuthenticationProvider.create(name: 'linkedin')
  end
end
```

```
end
end
```

Добавить миграцию user_authentication

```
class CreateUserAuthentications < ActiveRecord::Migration
  def change
    create_table "user_authentications", :force => true do |t|
      t.integer "user_id"
      t.integer "authentication_provider_id"
      t.string "uid"
      t.string "token"
      t.datetime "token_expires_at"
      t.text "params"
      t.datetime "created_at", :null => false
      t.datetime "updated_at", :null => false
    end
    add_index "user_authentications", ["authentication_provider_id"], :name =>
      "index_user_authentications_on_authentication_provider_id"
    add_index "user_authentications", ["user_id"], :name =>
      "index_user_authentications_on_user_id"
  end
end
```

Добавить ассоциации в authentication_provider.rb

```
has_many :users
has_many :user_authentications
```

Добавьте следующий код в user_authentication.rb

```
belongs_to :user
belongs_to :authentication_provider

serialize :params

def self.create_from_omniauth(params, user, provider)
  token_expires_at = params['credentials']['expires_at'] ?
    Time.at(params['credentials']['expires_at']).to_datetime : nil
  create(
    user: user,
    authentication_provider: provider,
    uid: params['uid'],
    token: params['credentials']['token'],
    token_expires_at: token_expires_at,
    params: params,
  )
end
```

Добавьте следующий код пользователю user.rb

```
include OmniauthAttributesConcern
```



```

has_many :user_authentications

devise :omniauthable, :database_authenticatable, :registerable, :recoverable, :rememberable,
:trackable

def self.create_from_omniauth(params)
  self.send(params.provider,params)
end

```

Добавить модель Концерн `omniauth_attributes_concern.rb`

```

module OmniauthAttributesConcern
  extend ActiveSupport::Concern
  module ClassMethods
    Add Methods here
  end
end

```

В этой связи мы можем создавать методы для каждого социального медиа для извлечения и хранения атрибутов.

```

def twitter params
  (params['info']['email'] = "dummy#{SecureRandom.hex(10)}@dummy.com") if
params['info']['email'].blank?
  attributes = {
    email: params['info']['email'],
    first_name: params['info']['name'].split(' ').first,
    last_name: params['info']['name'].split(' ').last,
    username: params['info']['nickname'],
    password: Devise.friendly_token
  }
  create(attributes)
end

```

Примечание. Twitter только возвращает адрес электронной почты, если пользователь подтвердил свою электронную почту в Twitter; в противном случае возвращается значение `nil`.

- Мы можем добавить другие учетные записи социальных сетей так же, как мы добавили Twitter выше.
- Изображение профиля из учетной записи в социальных сетях также можно получить, и оно будет передано как

```
remote_image_url: params['info']['image']
```

Примечание: приведенный выше пример предназначен для драгоценного камня [CarrierWave](#), а `'image'` в `remote_image_url` - столбец DB. Вы можете использовать любой другой драгоценный камень и передать `params['info']['image']`.

Добавить маршрут для пользователей / omniauth_callbacks_controller.rb в routes.rb

```
devise_for :users, controllers: {omniauth_callbacks: 'users/omniauth_callbacks'}
```

Добавить ключи для социальных сетей в devise.rb

Для Facebook

```
config.omniauth :facebook, facebook_app_id, facebook_secret_key, :display => "popup", :scope => 'email,publish_actions', info_fields: 'email,name'
```

Для Twitter

```
config.omniauth :twitter, twitter_app_id, twitter_secret_key, :display => "popup", :scope => 'email'
```

Для LinkedIn

```
config.omniauth :linkedin, linkedin_app_id, linkedin_secret_key, :display => "popup", :scope => 'r_emailaddress,r_basicprofile'
```

Для Google+

```
config.omniauth :gplus, gplus_app_id, gplus_secret_key, :display => "popup", scope: 'userinfo.email, userinfo.profile'
```

Примечание: атрибут `display: "popup"` используется для `display: "popup"` когда мы хотим, чтобы регистрация в социальных сетях открывалась в отдельном окне браузера.

Добавить Gems в Gemfile для OmniAuth

```
gem 'omniauth-oauth2', '~> 1.3.1'
gem 'omniauth'
gem 'omniauth-facebook'
gem 'omniauth-twitter'
gem 'omniauth-gplus'
gem 'omniauth-linkedin'
```

Запустите команду `bundle install`, перезапустите сервер и отпустите!

Прочитайте Многопользовательские устройства Devise & OmniAuth онлайн:

<https://riptutorial.com/ru/devise/topic/5846/многопользовательские-устройства-devise--amp--omniauth>

кредиты

S. No	Главы	Contributors
1	Начало работы с разработкой	Community , utkuDAT
2	Многопользовательские устройства Devise & OmniAuth	Chaudhary Prakash , Fakhir Shad , Johnno , Rory O'Kane , Simone Carletti