FREE eBook

LEARNING devise

Free unaffiliated eBook created from
Stack Overflow contributors.

#devise

Table of Contents

About

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.

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

Chapter 1: Getting started with devise

Remarks

This section provides an overview of what devise is, and why a developer might want to use it.

It should also mention any large subjects within devise, and link out to the related topics. Since the Documentation for devise is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

1. Add Devise Gem

Open up your **Gemfile** and add this line

```
gem 'devise'
```

Then run;

```
bundle install
```

2. Set up devise in your app

Run the following command in the terminal

```
rails g devise:install
```

3. Configure Devise

Ensure you have defined default url options in your environments files. Open up *config/environments/development.rb* and add this line:

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

Open up *app/views/layouts/application.html.erb* and add:

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

right above

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

Open up *app/views/ideas/show.html.erb* and remove the line that says:

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

- Do the same for *app/views/comments/show.html.erb*. These lines are not necessary as we’ve put the notice in the *app/views/layouts/application.html.erb* file.

4. Setup the User model

We’ll use a bundled generator script to create the User model.

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

Coach: Explain what user model has been generated. What are the fields?

5. Create your first user

Now that you have set everything up you can create your first user. Devise creates all the code and routes required to create accounts, log in, log out, etc.

Make sure your rails server is running, open **http://localhost:3000/users/sign_up** and create your user account.

Read Getting started with devise online: https://riptutorial.com/devise/topic/3561/getting-started-with-devise

Chapter 2: Devise & OmniAuth Multiple Providers

Examples

Add and install the Devise gem

Modify your `Gemfile` to include the Devise gem:

```
gem 'devise'
```

Then update your gems with:

```
$ bundle install
```

Run the installers in your project:

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

Then create your Devise model (e.g. `User`) with:

```
$ rails generate devise User
```

you'll need to set up the default URL options for the Devise mailer in environment.rb file config/environments/development.rb:

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

Devise makes some changes to your database, so you'll need to run the created migrations:

```
$ rake db:migrate
```

Create Controller users/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
```

> Note: In the part “`%w[facebook twitter gplus linkedin]`”, you should list all the social accounts you will use.

Create Controller Concern 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
```

Add authentication_provider migration

```
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
```

Add user_authentication migration

```
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
```

Add Associations in authentication_provider.rb

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

Add the following code to 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
```

Add the following code to 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
```

Add Model Concern omniauth_attributes_concern.rb

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

In this concern we can create methods for each social media to fetch and store attributes.

```
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
```

> Note: Twitter only returns an email address if the user has confirmed his/her email at Twitter; otherwise the `nil` value is returned.

- We can add other social media accounts the same way we have added Twitter above.
- The profile image from the social media account can also be fetched, and will be passed as

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

  > Note: the above example is meant for the CarrierWave gem, and `'image'` in `remote_image_url` is the DB column. You can use any other gem and pass `params['info']['image']` to it.

Add route for users/omniauth_callbacks_controller.rb in routes.rb

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

Add Social Media Account Keys in devise.rb

For Facebook

```
config.omniauth :facebook, facebook_app_id, facebook_secret_key, :display => "popup", :scope
```

```
=> 'email,publish_actions', info_fields: 'email,name'
```

For Twitter

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

For Linkedin

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

For Google+

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

Note: the `display: "popup"` attribute is used when we want social media signup to open in a separate browser window.

Add Gems in Gemfile for OmniAuth

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

Run the `bundle install` command, restart your server and off you go!

Read Devise & OmniAuth Multiple Providers online: https://riptutorial.com/devise/topic/5846/devise---omniauth-multiple-providers

Credits

S. No	Chapters	Contributors
1	Getting started with devise	Community, utkuDAT
2	Devise & OmniAuth Multiple Providers	Chaudhary Prakash, Fakhir Shad, Johno, Rory O'Kane, Simone Carletti