



EBook Gratis

APRENDIZAJE devise

Free unaffiliated eBook created from
Stack Overflow contributors.

#devise

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con el dispositivo.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
1. Añadir gema Devise.....	2
2. Configura el dispositivo en tu aplicación.....	2
3. Configure el dispositivo.....	2
4. Configurar el modelo de usuario.....	3
5. Crea tu primer usuario.....	3
Capítulo 2: Devise & OmniAuth múltiples proveedores.....	4
Examples.....	4
Agrega e instala la gema Devise.....	4
Crear usuarios de Controller / omniauth_callbacks_controller.rb.....	4
Crear problema de controlador omni_concern.rb.....	5
Añadir la autenticación de la migración del proveedor.....	5
Añadir migración de autenticación de usuario.....	6
Añadir asociaciones en authentication_provider.rb.....	6
Agregue el siguiente código a user_authentication.rb.....	6
Agregue el siguiente código a user.rb.....	6
Agregar preocupación de modelo omniauth_attributes_concern.rb.....	7
Agregar ruta para los usuarios / omniauth_callbacks_controller.rb en route.rb.....	7
Agregar claves de cuenta de redes sociales en devise.rb.....	7
Añadir gemas en Gemfile para OmniAuth.....	8
Creditos.....	9

Acerca de

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

Capítulo 1: Empezando con el dispositivo

Observaciones

Esta sección proporciona una descripción general de qué es el dispositivo y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro del dispositivo, y vincular a los temas relacionados. Dado que la Documentación para dispositivos es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Examples

Instalación o configuración

1. Añadir gema Devise

Abre tu **Gemfile** y agrega esta línea

```
gem 'devise'
```

Entonces corre;

```
bundle install
```

2. Configura el dispositivo en tu aplicación.

Ejecuta el siguiente comando en la terminal

```
rails g devise:install
```

3. Configure el dispositivo

Asegúrese de que ha definido las opciones de URL predeterminadas en los archivos de sus entornos. Abra *config / environment / development.rb* y agregue esta línea:

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

Abra la *aplicación / views / layouts / application.html.erb* y agregue:

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

justo encima

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

Abra `app / views / ideas / show.html.erb` y elimine la línea que dice:

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

- Haga lo mismo para `app / views / comments / show.html.erb` . Estas líneas no son necesarias ya que hemos puesto el aviso en el archivo `app / views / layouts / application.html.erb` .

4. Configurar el modelo de usuario

Usaremos un script de generador agrupado para crear el modelo de usuario.

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

Entrenador: Explique qué modelo de usuario se ha generado. ¿Qué son los campos?

5. Crea tu primer usuario.

Ahora que ya has configurado todo, puedes crear tu primer usuario. Devise crea todo el código y las rutas necesarias para crear cuentas, iniciar sesión, cerrar sesión, etc.

Asegúrese de que su servidor de rieles esté funcionando, abra http://localhost:3000/users/sign_up y cree su cuenta de usuario.

Lea [Empezando con el dispositivo en línea](https://riptutorial.com/es/devise/topic/3561/empezando-con-el-dispositivo):

<https://riptutorial.com/es/devise/topic/3561/empezando-con-el-dispositivo>

Capítulo 2: Devise & OmniAuth múltiples proveedores

Examples

Agrega e instala la gema Devise

Modifique su `Gemfile` para incluir la gema Devise:

```
gem 'devise'
```

Luego actualiza tus gemas con:

```
$ bundle install
```

Ejecuta los instaladores en tu proyecto:

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

Luego crea tu modelo Devise (por ejemplo, `User`) con:

```
$ rails generate devise User
```

Tendrá que configurar las opciones de URL predeterminadas para la aplicación de correo de Devise en el archivo `environment.rb` config / environment / development.rb:

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

Devise realiza algunos cambios en su base de datos, por lo que deberá ejecutar las migraciones creadas:

```
$ rake db:migrate
```

Crear usuarios de Controller / `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
```

Nota: En la parte "`%w[facebook twitter gplus linkedin]`", debe enumerar todas las

cuentas sociales que usará.

Crear problema de controlador 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
```

Añadir la autenticación de la migración del proveedor

```
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
```

Añadir migración de autenticación de usuario

```
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
```

Añadir asociaciones en authentication_provider.rb

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

Agregue el siguiente código a 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
```

Agregue el siguiente código a 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
```

Agregar preocupación de modelo omniauth_attributes_concern.rb

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

En esta preocupación, podemos crear métodos para cada red social para obtener y almacenar atributos.

```
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
```

Nota: Twitter solo devuelve una dirección de correo electrónico si el usuario ha confirmado su correo electrónico en Twitter; de lo contrario se devuelve el valor `nil`.

- Podemos agregar otras cuentas de redes sociales de la misma manera que hemos agregado Twitter anteriormente.
- La imagen de perfil de la cuenta de redes sociales también se puede recuperar, y se pasará como

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

Nota: el ejemplo anterior está destinado a la gema de [CarrierWave](#), y la `'image'` en `remote_image_url` es la columna DB. Puedes usar cualquier otra gema y pasar `params['info']['image']` para ello.

Agregar ruta para los usuarios / omniauth_callbacks_controller.rb en route.rb

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

Agregar claves de cuenta de redes sociales en devise.rb

Para facebook

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

Para twitter

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

Para LinkedIn

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

Para Google+

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

Nota: el atributo `display: "popup"` se usa cuando queremos que el registro de redes sociales se abra en una ventana del navegador por separado.

Añadir gemas en Gemfile para OmniAuth

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

Ejecute el comando `bundle install` , reinicie su servidor y listo!

Lea [Devise & OmniAuth múltiples proveedores en línea](https://riptutorial.com/es/devise/topic/5846/devise--amp--omniauth-multiples-proveedores):

<https://riptutorial.com/es/devise/topic/5846/devise--amp--omniauth-multiples-proveedores>

Creditos

S. No	Capítulos	Contributors
1	Empezando con el dispositivo	Community , utkuDAT
2	Devise & OmniAuth múltiples proveedores	Chaudhary Prakash , Fakhir Shad , Johnno , Rory O'Kane , Simone Carletti