



Kostenloses eBook

LERNEN

devise

Free unaffiliated eBook created from
Stack Overflow contributors.

#devise

Inhaltsverzeichnis

Über.....	1
Kapitel 1: Erste Schritte mit Devise.....	2
Bemerkungen.....	2
Examples.....	2
Installation oder Setup.....	2
1. Devise Gem hinzufügen.....	2
2. Richten Sie das Gerät in Ihrer App ein.....	2
3. Konfigurieren Sie Devise.....	2
4. Richten Sie das Benutzermodell ein.....	3
5. Erstellen Sie Ihren ersten Benutzer.....	3
Kapitel 2: Devise & OmniAuth mehrere Anbieter.....	4
Examples.....	4
Fügen Sie den Devise-Edelstein hinzu und installieren Sie ihn.....	4
Erstellen Sie Controller-Benutzer / omniauth_callbacks_controller.rb.....	4
Controller Concern erstellen omni_concern.rb.....	5
Fügen Sie eine authentication_provider-Migration hinzu.....	5
Fügen Sie die Benutzerauthentifizierungsmigration hinzu.....	6
Fügen Sie Assoziationen in authentication_provider.rb hinzu.....	6
Fügen Sie den folgenden Code zu user_authentication.rb hinzu.....	6
Fügen Sie den folgenden Code zu user.rb hinzu.....	6
Modellanliegen hinzufügen omniauth_attributes_concern.rb.....	7
Fügen Sie die Route für Benutzer / omniauth_callbacks_controller.rb in routes.rb hinzu.....	7
Fügen Sie Schlüssel für Social Media-Konten in devise.rb hinzu.....	8
Fügen Sie Edelsteine in Gemfile für OmniAuth hinzu.....	8
Credits.....	9



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

Kapitel 1: Erste Schritte mit Devise

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was ein Gerät ist und warum ein Entwickler es verwenden möchte.

Es sollte auch alle großen Themen innerhalb des Projekts erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für devise neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Examples

Installation oder Setup

1. Devise Gem hinzufügen

Öffne dein **Gemfile** und füge diese Zeile hinzu

```
gem 'devise'
```

Dann renne;

```
bundle install
```

2. Richten Sie das Gerät in Ihrer App ein

Führen Sie den folgenden Befehl im Terminal aus

```
rails g devise:install
```

3. Konfigurieren Sie Devise

Stellen Sie sicher, dass Sie in den Umgebungsdateien Standard-URL-Optionen definiert haben. Öffnen Sie *config / virones / development.rb* und fügen Sie diese Zeile hinzu:

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

Öffnen Sie *app / views / layouts / application.html.erb* und fügen Sie *Folgendes* hinzu:

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

direkt darüber

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

Öffnen Sie `app / views / ideas / show.html.erb` und entfernen Sie die folgende Zeile:

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

- Machen Sie dasselbe für `app / views / comments / show.html.erb`. Diese Zeilen sind nicht erforderlich, da wir die Mitteilung in die Datei `app / views / layouts / application.html.erb` aufgenommen haben.

4. Richten Sie das Benutzermodell ein

Wir verwenden ein gebündeltes Generator-Skript, um das Benutzermodell zu erstellen.

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

Coach: Erläutern Sie, welches Benutzermodell erstellt wurde. Was sind die Felder?

5. Erstellen Sie Ihren ersten Benutzer

Nachdem Sie alles eingerichtet haben, können Sie Ihren ersten Benutzer erstellen. Devise erstellt alle Codes und Routen, die zum Erstellen von Konten, Anmelden und Abmelden erforderlich sind.

Stellen Sie sicher, dass Ihr Schienen-Server ausgeführt wird, öffnen Sie [http://localhost:3000 / users / sign_up](http://localhost:3000/users/sign_up) und erstellen Sie Ihr Benutzerkonto.

Erste Schritte mit Devise online lesen: <https://riptutorial.com/de/devise/topic/3561/erste-schritte-mit-devise>

Kapitel 2: Devise & OmniAuth mehrere Anbieter

Examples

Fügen Sie den Devise-Edelstein hinzu und installieren Sie ihn

Gemfile Ihr Gemfile , dass es den Devise-Edelstein enthält:

```
gem 'devise'
```

Dann aktualisiere deine Edelsteine mit:

```
$ bundle install
```

Führen Sie die Installationsprogramme in Ihrem Projekt aus:

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

Dann erstellen Sie Ihr Devise-Modell (zB User) mit:

```
$ rails generate devise User
```

Sie müssen die Standard-URL-Optionen für den Devise-Mailer in der Datei environment.rb config / umwelts / entwicklung.rb einrichten:

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

Devise nimmt einige Änderungen an Ihrer Datenbank vor, daher müssen Sie die erstellten Migrationen ausführen:

```
$ rake db:migrate
```

Erstellen Sie Controller-Benutzer / 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
```

Hinweis: Im Teil „%w[facebook twitter gplus linkedin]“ sollten Sie alle Social Accounts

%w[facebook twitter gplus linkedin] , die Sie verwenden werden.

Controller Concern erstellen 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
```

Fügen Sie eine authentication_provider-Migration hinzu

```
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
```

Fügen Sie die Benutzerauthentifizierungsmigration hinzu

```
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
```

Fügen Sie Assoziationen in authentication_provider.rb hinzu

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

Fügen Sie den folgenden Code zu user_authentication.rb hinzu

```
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
```

Fügen Sie den folgenden Code zu user.rb hinzu

```
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
```

Modellanliegen hinzufügen omniauth_attributes_concern.rb

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

In diesem Zusammenhang können wir für jedes soziale Medium Methoden erstellen, um Attribute abzurufen und zu speichern.

```
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
```

Hinweis: Twitter gibt nur eine E-Mail-Adresse zurück, wenn der Benutzer seine E-Mail-Adresse bei Twitter bestätigt hat. Andernfalls wird der `nil` zurückgegeben.

- Andere Social-Media-Konten können auf dieselbe Weise hinzugefügt werden, auf die wir Twitter oben hinzugefügt haben.
- Das Profilbild aus dem Social Media-Konto kann auch abgerufen werden und wird als weitergegeben

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

Hinweis: Das obige Beispiel ist für den [CarrierWave-Gem](#) gedacht und `'image'` in `remote_image_url` ist die DB-Spalte. Sie können jeden anderen Edelstein verwenden und ihm `params['info']['image']` .

Fügen Sie die Route für Benutzer / omniauth_callbacks_controller.rb in routes.rb hinzu

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

Fügen Sie Schlüssel für Social Media-Konten in devise.rb hinzu

Für Facebook

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

Für Twitter

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

Für LinkedIn

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

Für Google+

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

Hinweis: Das Attribut `display: "popup"` wird verwendet, wenn die Social-Media-Anmeldung in einem separaten Browserfenster geöffnet werden soll.

Fügen Sie Edelsteine in Gemfile für OmniAuth hinzu

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

Führen Sie den Befehl zum `bundle install` des `bundle install` , starten Sie den Server neu und los geht's!

Devise & OmniAuth mehrere Anbieter online lesen:

<https://riptutorial.com/de/devise/topic/5846/devise--amp--omniauth-mehrere-anbieter>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit Devise	Community , utkuDAT
2	Devise & OmniAuth mehrere Anbieter	Chaudhary Prakash , Fakhir Shad , Johnno , Rory O'Kane , Simone Carletti