

 eBook Gratuit

# APPRENEZ deviser

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#deviser

# Table des matières

À propos.....	1
<b>Chapitre 1: Commencer à utiliser.....</b>	<b>2</b>
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
<b>1. Ajouter Devise Gem.....</b>	<b>2</b>
<b>2. Configurez votre appareil dans votre application.....</b>	<b>2</b>
<b>3. Configurez Devise.....</b>	<b>2</b>
<b>4. Configurez le modèle d'utilisateur.....</b>	<b>3</b>
<b>5. Créez votre premier utilisateur.....</b>	<b>3</b>
<b>Chapitre 2: Devise &amp; OmniAuth Multiple Providers.....</b>	<b>4</b>
Exemples.....	4
Ajouter et installer le bijou Devise.....	4
Créer des utilisateurs du contrôleur / omniauth_callbacks_controller.rb.....	4
Créer un problème de contrôleur omni_concern.rb.....	5
Ajouter la migration authentication_provider.....	5
Ajouter la migration user_authentication.....	6
Ajouter des associations dans authentication_provider.rb.....	6
Ajoutez le code suivant à user_authentication.rb.....	6
Ajoutez le code suivant à user.rb.....	6
Ajouter un problème concernant le modèle omniauth_attributes_concern.rb.....	7
Ajouter une route pour les utilisateurs / omniauth_callbacks_controller.rb dans routes.rb.....	7
Ajouter des clés de compte de média social dans.....	8
Ajouter des gemmes dans Gemfile pour OmniAuth.....	8
<b>Crédits.....</b>	<b>9</b>

---

# À propos

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](mailto:info@zzzprojects.com)

---

# Chapitre 1: Commencer à utiliser

## Remarques

Cette section fournit une vue d'ensemble de ce qu'est le concept et de la raison pour laquelle un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les sujets importants dans le cadre du projet, et établir un lien avec les sujets connexes. Étant donné que la documentation pour le périphérique est nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

## Exemples

### Installation ou configuration

---

## 1. Ajouter Devise Gem

Ouvrez votre **Gemfile** et ajoutez cette ligne

```
gem 'devise'
```

Puis cours;

```
bundle install
```

---

## 2. Configurez votre appareil dans votre application

Exécutez la commande suivante dans le terminal

```
rails g devise:install
```

---

## 3. Configurez Devise

Assurez-vous d'avoir défini les options d'url par défaut dans les fichiers de votre environnement. Ouvrez *config / environnements / development.rb* et ajoutez cette ligne:

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

Ouvrez `app / views / layouts / application.html.erb` et ajoutez:

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

juste au dessus

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

Ouvrez `app / views / ideas / show.html.erb` et supprimez la ligne indiquant:

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

- Faites de même pour `app / views / comments / show.html.erb` . Ces lignes ne sont pas nécessaires car nous avons mis l'avis dans le fichier `app / views / layouts / application.html.erb` .

---

## 4. Configurez le modèle d'utilisateur

Nous utiliserons un script de générateur groupé pour créer le modèle d'utilisateur.

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

**Coach:** Expliquez quel modèle d'utilisateur a été généré. Quels sont les champs?

---

## 5. Créez votre premier utilisateur

Maintenant que vous avez tout configuré, vous pouvez créer votre premier utilisateur. Devise crée tout le code et les itinéraires nécessaires pour créer des comptes, se connecter, se déconnecter, etc.

Assurez-vous que votre serveur rails est en marche, ouvrez [http://localhost:3000/users/sign\\_up](http://localhost:3000/users/sign_up) et créez votre compte utilisateur.

Lire Commencer à utiliser en ligne: <https://riptutorial.com/fr/devise/topic/3561/commencer-a-utiliser>

---

# Chapitre 2: Devise & OmniAuth Multiple Providers

## Exemples

### Ajouter et installer le bijou Devise

Modifiez votre `Gemfile` pour inclure le bijou Devise:

```
gem 'devise'
```

Puis mettez à jour vos gemmes avec:

```
$ bundle install
```

Exécutez les installateurs dans votre projet:

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

Créez ensuite votre modèle Devise (par ex. `User` ) avec:

```
$ rails generate devise User
```

Vous devez configurer les options d'URL par défaut pour le logiciel de messagerie Devise dans le fichier `environment.rb` config / environnements / `development.rb`:

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

Devise apporte des modifications à votre base de données, vous devrez donc exécuter les migrations créées:

```
$ rake db:migrate
```

### Créer des utilisateurs du contrôleur / `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
```

Remarque: Dans la partie « `%w[facebook twitter gplus linkedin]` », vous devez lister

tous les comptes sociaux que vous utiliserez.

## Créer un problème de contrôleur 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
```

## Ajouter la migration 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
```

## Ajouter la migration 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
```

## Ajouter des associations dans authentication\_provider.rb

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

## Ajoutez le code suivant à 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
```

## Ajoutez le code suivant à 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
```

## Ajouter un problème concernant le modèle omniauth\_attributes\_concern.rb

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

Dans cette préoccupation, nous pouvons créer des méthodes pour chaque média social pour récupérer et stocker des attributs.

```
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
```

Remarque: Twitter ne renvoie une adresse électronique que si l'utilisateur a confirmé son adresse électronique sur Twitter. sinon la valeur `nil` est renvoyée.

- Nous pouvons ajouter d'autres comptes de médias sociaux de la même manière que nous avons ajouté Twitter ci-dessus.
- L'image de profil du compte de média social peut également être récupérée et transmise comme

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

Note: l'exemple ci-dessus est destiné à la gemme [CarrierWave](#) , et 'image' dans `remote_image_url` est la colonne DB. Vous pouvez utiliser n'importe quelle autre gem et y passer les `params['info']['image']` .

## Ajouter une route pour les utilisateurs / omniauth\_callbacks\_controller.rb dans routes.rb

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

## Ajouter des clés de compte de média social dans

### Pour Facebook

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

### Pour Twitter

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

### Pour LinkedIn

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

### Pour Google+

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

Remarque: l'attribut `display: "popup"` est utilisé lorsque nous souhaitons que l'inscription aux réseaux sociaux s'ouvre dans une fenêtre de navigateur distincte.

## Ajouter des gemmes dans Gemfile pour OmniAuth

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

Exécutez la commande `d bundle install` , redémarrez votre serveur et c'est parti!

Lire [Devise & OmniAuth Multiple Providers en ligne](https://riptutorial.com/fr/devise/topic/5846/devise--amp--omniauth-multiple-providers):

<https://riptutorial.com/fr/devise/topic/5846/devise--amp--omniauth-multiple-providers>

# Crédits

S. No	Chapitres	Contributeurs
1	Commencer à utiliser	<a href="#">Community</a> , <a href="#">utkuDAT</a>
2	Devise & OmniAuth Multiple Providers	<a href="#">Chaudhary Prakash</a> , <a href="#">Fakhir Shad</a> , <a href="#">Johnno</a> , <a href="#">Rory O'Kane</a> , <a href="#">Simone Carletti</a>