



**EBook Gratuito**

# APPENDIMENTO

## devise

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#devise**

# Sommario

Di.....	1
<b>Capitolo 1: Iniziare con Idea.....</b>	<b>2</b>
Osservazioni.....	2
Examples.....	2
Installazione o configurazione.....	2
<b>1. Aggiungi Devise Gem.....</b>	<b>2</b>
<b>2. Installa idea nella tua app.....</b>	<b>2</b>
<b>3. Configura Devise.....</b>	<b>2</b>
<b>4. Impostare il modello utente.....</b>	<b>3</b>
<b>5. Crea il tuo primo utente.....</b>	<b>3</b>
<b>Capitolo 2: Devise &amp; OmniAuth Provider multipli.....</b>	<b>4</b>
Examples.....	4
Aggiungi e installa la gemma Devise.....	4
Crea utenti Controller / omniauth_callbacks_controller.rb.....	4
Crea Controller Concern omni_concern.rb.....	5
Aggiungi la migrazione authentication_provider.....	5
Aggiungi migrazione user_authentication.....	6
Aggiungi associazioni in authentication_provider.rb.....	6
Aggiungere il seguente codice a user_authentication.rb.....	6
Aggiungi il seguente codice a user.rb.....	6
Aggiungi Model Concern omniauth_attributes_concern.rb.....	7
Aggiungi percorso per utenti / omniauth_callbacks_controller.rb in routes.rb.....	7
Aggiungi le chiavi dell'account dei social media in devise.rb.....	7
Aggiungi gemme in Gemfile per OmniAuth.....	8
<b>Titoli di coda.....</b>	<b>9</b>

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)

---

# Capitolo 1: Iniziare con Idea

## Osservazioni

Questa sezione fornisce una panoramica di ciò che idea è, e perché uno sviluppatore potrebbe voler usarlo.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno del progetto e collegarsi agli argomenti correlati. Poiché la Documentazione per ideare è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

## Examples

### Installazione o configurazione

---

## 1. Aggiungi Devise Gem

Apri il tuo **Gemfile** e aggiungi questa linea

```
gem 'devise'
```

Quindi corri;

```
bundle install
```

---

## 2. Installa idea nella tua app

Esegui il seguente comando nel terminale

```
rails g devise:install
```

---

## 3. Configura Devise

Assicurati di aver definito le opzioni di URL di default nei tuoi file di ambiente. Apri *config / environments / development.rb* e aggiungi questa riga:

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

Apri *app / views / layouts / application.html.erb* e aggiungi:

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

proprio sopra

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

Apri `app / views / ideas / show.html.erb` e rimuovi la riga che dice:

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

- Fai lo stesso per `app / views / comments / show.html.erb` . Queste righe non sono necessarie poiché abbiamo inserito la notifica nel file `app / views / layouts / application.html.erb` .

---

## 4. Impostare il modello utente

Utilizzeremo uno script di generatore in bundle per creare il modello utente.

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

**Allenatore:** spiega quale modello utente è stato generato. Quali sono i campi?

---

## 5. Crea il tuo primo utente

Ora che hai impostato tutto, puoi creare il tuo primo utente. Devise crea tutto il codice e le rotte necessarie per creare account, accedere, disconnettersi, ecc.

Assicurati che il tuo server delle rotaie sia in esecuzione, apri [http://localhost:3000/users/sign\\_up](http://localhost:3000/users/sign_up) e crea il tuo account utente.

Leggi [Iniziare con Idea online](https://riptutorial.com/it/devise/topic/3561/iniziare-con-idea): <https://riptutorial.com/it/devise/topic/3561/iniziare-con-idea>

---

# Capitolo 2: Devise & OmniAuth Provider multipli

## Examples

### Aggiungi e installa la gemma Devise

Modifica il tuo `Gemfile` per includere la gemma Devise:

```
gem 'devise'
```

Quindi aggiorna le tue gemme con:

```
$ bundle install
```

Esegui gli installer nel tuo progetto:

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

Quindi crea il tuo modello Devise (ad es. `User` ) con:

```
$ rails generate devise User
```

dovrai configurare le opzioni URL predefinite per il mailer Devise nel file `environment.rb` config / `environments / development.rb`:

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

Devise apporta alcune modifiche al tuo database, quindi dovrai eseguire le migrazioni create:

```
$ rake db:migrate
```

### Crea utenti 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:** nella parte "`%w[facebook twitter gplus linkedin]`", dovresti elencare tutti gli account social che utilizzerai.

## Crea 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
```

## Aggiungi la migrazione 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
```

## Aggiungi migrazione 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
```

## Aggiungi associazioni in authentication\_provider.rb

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

## Aggiungere il seguente codice 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
```

## Aggiungi il seguente codice 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
```

## Aggiungi Model Concern omniauth\_attributes\_concern.rb

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

In questa preoccupazione possiamo creare metodi per ogni social media per recuperare e memorizzare gli attributi.

```
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 restituisce solo un indirizzo e-mail se l'utente ha confermato la sua e-mail su Twitter; altrimenti viene restituito il valore `nil`.

- Possiamo aggiungere altri account di social media nello stesso modo in cui abbiamo aggiunto Twitter sopra.
- L'immagine del profilo dall'account dei social media può anche essere recuperata e passata come

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

Nota: l'esempio precedente è inteso per la gemma [CarrierWave](#) e `'image'` in `remote_image_url` è la colonna DB. Puoi usare qualsiasi altra gemma e passare i `params['info']['image']` su di essa.

## Aggiungi percorso per utenti / omniauth\_callbacks\_controller.rb in routes.rb

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

## Aggiungi le chiavi dell'account dei social media in devise.rb

## Per Facebook

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

## Per Twitter

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

## Per LinkedIn

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

## Per Google+

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

**Nota:** il `display: "popup"` attributo `display: "popup"` è usato quando vogliamo che la registrazione sui social media si apra in una finestra separata del browser.

## Aggiungi gemme in Gemfile per OmniAuth

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

Esegui il comando di `bundle install` del `bundle install` , riavvia il server e via!

Leggi [Devise & OmniAuth Provider multipli online](https://riptutorial.com/it/devise/topic/5846/devise--amp--omniauth-provider-multipli):

<https://riptutorial.com/it/devise/topic/5846/devise--amp--omniauth-provider-multipli>

---

## Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con Idea	<a href="#">Community</a> , <a href="#">utkuDAT</a>
2	Devise & OmniAuth Provider multipli	<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>