

 免费电子书

学习

devise

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

#devise

.....	1
<b>1:</b> .....	<b>2</b>
.....	2
Examples.....	2
.....	2
<b>1.Devise Gem</b> .....	<b>2</b>
<b>2.</b> .....	<b>2</b>
<b>3.</b> .....	<b>2</b>
<b>4.</b> .....	<b>3</b>
<b>5.</b> .....	<b>3</b>
<b>2: DeviseOmniAuth</b> .....	<b>4</b>
Examples.....	4
Devise gem.....	4
Controller/ omniauth_callbacks_controller.rb.....	4
omniauth_concern.rb.....	4
authentication_provider.....	5
user_authentication.....	5
authentication_provider.rb.....	6
user_authentication.rb.....	6
user.rb.....	6
omniauth_attributes_concern.rb.....	6
routes.rbusers / omniauth_callbacks_controller.rb.....	7
devise.rb.....	7
GemfileOmniAuthGems.....	8
.....	<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)

---

# 1:

- 
- ◦

## Examples

---

# 1.Devise Gem

### Gemfile

```
gem 'devise'
```

```
;
```

```
bundle install
```

---

# 2.

```
rails g devise:install
```

---

# 3.

URL。 *config / environments / development.rb*

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

*app / views / layouts / application.html.erb*

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

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

*app / views / ideas / show.html.erb*

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

- `app / views / comments / show.html.erb` ◦ `app / views / layouts / application.html.erb`◦

---

## 4.

◦

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

### Coach◦

---

## 5.

- Devise◦

rails [http// localhost3000 / users / sign\\_up](http://localhost:3000/users/sign_up)◦

<https://riptutorial.com/zh-CN/devise/topic/3561/>

---

## 2: DeviseOmniAuth

### Examples

#### Devise gem

Gemfile Devise gem

```
gem 'devise'
```

```
$ bundle install
```

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

#### DeviseUser

```
$ rails generate devise User
```

environment.rb config / environments / development.rb DeviseURL

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

#### Devise

```
$ rake db:migrate
```

#### 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
```

“ %w[facebook twitter gplus linkedin] ”。

#### omniauth\_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
  end
end
```

```

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
end
end

```

## 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
end

```

## 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"
    end
  end
end

```

```

      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
end

```

## authentication\_provider.rb

```

has_many :users
has_many :user_authentications

```

## 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

```

## 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

```

## omniauth\_attributes\_concern.rb



```

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

```

◦

```

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

```

Twitter/Twitter;nil◦

- Twitter◦

- remote\_image\_url: params['info']['image']

[CarrierWave](#) gem remote\_image\_url 'image' DB◦ params['info']['image']◦

## routes.rb users / omniauth\_callbacks\_controller.rb

```

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

```

## devise.rb

### Facebook

```

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

```

### Twitter

```

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

```

### Linkedin

```

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

```

## Google+

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

```
display: "popup"◦
```

## GemfileOmniAuthGems

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

```
bundle install
```

**DeviseOmniAuth** <https://riptutorial.com/zh-CN/devise/topic/5846/devise-omniauth>

---

S. No		Contributors
1		<a href="#">Community</a> , <a href="#">utkuDAT</a>
2	DeviseOmniAuth	<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>