by. Yein

Devise Gem

by. Yein

GEM

루비에서 지원하는 '패키지 매니저 시스템'.

다른 사람들이 만들어놓은 '패키지'들을
가져다 쓸 수 있게 도와주는 시스템!

by. Yein

젬의 무수히 많은 쓰임새

사진첨부	Carrierwave
로그인기능	Devise
자동화	Whenever
페이지네이션	Kaminari
게시판 텍스트 편집기	Tiny MCE
조회수 구현	Impressionist
권한관리	Cancancan / Rolify
게시물 삭제 기록 남기기	Paranoia
환경변수 설정	Figaro

젬 왜 쓰는고양?

~~맛있어서..~~

효율적일고양!

안정적일고양!

오늘은 디바이스!

~~이미 서강대 복커톤에서 모든 팀이 다 알아부렸지만.. ㅠㅠ~~

세션기능
세션유지
사용자인증/확인
비밀번호 보안
사용자정보 보안
.
.
.

by. Yein

https://github.com/plataformatec/devise

by. Yein

오늘의 디바이스

1. 기본 세팅
2. 컬럼추가
3. 이메일 인증

1. 기본세팅

scaffold를 이용해, 게시판을 만들어주자!

(route설정과 rake db:migrate 도 빼먹지말자!)

혹은 기존에 만들어놓았던 게시판 프로젝트를 열어줍시다!

1. 기본세팅

$ rails g scaffold post title:text content:text

routes > root 'posts#index'

by. Yein

1. 기본세팅

Gemfile > gem 'devise'

1. 기본세팅

$bundle install

1. 기본세팅

$rails g devise:install

```
bambi1859:~/workspace $ rails g devise:install
Running via Spring preloader in process 3177
      create  config/initializers/devise.rb
      create  config/locales/devise.en.yml
```

1. 기본세팅

$rails g devise user

```
bambi1859:~/workspace $ rails g devise user
Running via Spring preloader in process 3198
      invoke  active_record
      create    db/migrate/20180514102136_devise_create_users.rb
      create    app/models/user.rb
      invoke    test_unit
      create      test/models/user_test.rb
      create      test/fixtures/users.yml
      insert    app/models/user.rb
       route  devise_for :users
```

1. 기본세팅

$rails g devise:views users

```
bambi1859:~/workspace $ rails g devise:views users
Running via Spring preloader in process 3211
      invoke  Devise::Generators::SharedViewsGenerator
      create    app/views/users/shared
      create    app/views/users/shared/_links.html.erb
      invoke  form_for
      create    app/views/users/confirmations
      create    app/views/users/confirmations/new.html.erb
      create    app/views/users/passwords
      create    app/views/users/passwords/edit.html.erb
      create    app/views/users/passwords/new.html.erb
      create    app/views/users/registrations
      create    app/views/users/registrations/edit.html.erb
      create    app/views/users/registrations/new.html.erb
      create    app/views/users/sessions
      create    app/views/users/sessions/new.html.erb
      create    app/views/users/unlocks
      create    app/views/users/unlocks/new.html.erb
      invoke  erb
      create    app/views/users/mailer
```

1. 기본세팅

```
<% if user_signed_in? %>
<%= current_user.email %>
<%= link_to "로그아웃", destroy_user_session_path %>
<% else %>
<%= link_to "로그인", [redacted] %>
<%= link_to "회원가입", [redacted] %>
<% end %>
```

$rake routes

1. 기본세팅

user_signed_in?

current_user

by. Yein

1. 기본세팅

똑똑한 DEVISE

중복된 이메일 주소 알아내줌

공란 있는 주소 빠꾸먹여줌

이메일 형식만 변별해줌

1. 기본세팅

config/initialize>devise.rb

```
253   # The default HTTP method used to sign out a resource. Default is :delete.
254   config.sign_out_via = :get
```

1. 기본세팅

MISSION

user_signed_in? current_user

를 사용해서 권한 설정을 해보자!

1. 로그인한 유저만 글을 쓸 수 있게 한다.

2. 내가 쓴 글만 삭제할 수 있게 한다.

hint! hidden_field 를 사용한다.

1. 기본세팅

정답

view file

```
<% if post.userid == current_user.name %>
```

new.html

```
<%= f.hidden_field :email, value: current_user.email %>
```

column추가도!

2. 컬럼추가하기

Gemfile > gem 'rails_db'

$bundle install

2. 컬럼추가하기

$rails g migration AddUserattributeToUsers username:string admin:boolean

마이그레이션 이름 모델명

$rake db:migrate

2. 컬럼추가하기

app>views>users>registrations>new.html.erb / edit.html.erb

```
<div class="field">
  <%= f.label :이름 %><br />
  <%= f.text_field :username, autofocus: true %>
</div>
```

2. 컬럼추가하기

```
class ApplicationController < ActionController::Base
  # Prevent CSRF attacks by raising an exception.
  # For APIs, you may want to use :null_session instead.
  before_action :configure_permitted_parameters, if: :devise_controller?
  protect_from_forgery with: :exception

  protected
    def configure_permitted_parameters
      devise_parameter_sanitizer.permit(:sign_up, keys: [:username, :admin])
      devise_parameter_sanitizer.permit(:account_update, keys: [:username, :admin])
    end

end
```

3. 이메일 인증하기

어떤 단계가 더 필요할까요?

by. Yein

3. 이메일 인증하기

일반적 디바이스 로그인 단계

by. Yein

3. 이메일 인증하기

일반적 디바이스 로그인 단계

1. 유저가 회원가입 폼을 통해 가입 신청

3. 이메일 인증하기

일반적 디바이스 로그인 단계

1. 유저가 회원가입 폼을 통해 가입 신청

2. 필요한 값을 올바르게 채워 보냈다면 새로운 유저 생성

by. Yein

3. 이메일 인증하기

이메일 인증 로그인 단계

3. 이메일 인증하기

이메일 인증 로그인 단계

1. 유저가 회원가입 폼을 통해 가입 신청

3. 이메일 인증하기

이메일 인증 로그인 단계

1. 유저가 회원가입 폼을 통해 가입 신청

2. 필요한 값을 올바르게 채워 보냈다면 **인증되지 않은** 새로운 유저 생성

3. 이메일 인증하기

이메일 인증 로그인 단계

1. 유저가 회원가입 폼을 통해 가입 신청

2. 필요한 값을 올바르게 채워 보냈다면 인증되지 않은 새로운 유저 생성

3. 해당 이메일로 인증 메일을 전송

3. 이메일 인증하기

이메일 인증 로그인 단계

1. 유저가 회원가입 폼을 통해 가입 신청

2. 필요한 값을 올바르게 채워 보냈다면 인증되지 않은 새로운 유저 생성

3. 해당 이메일로 인증 메일을 전송

4. 유저에게 이메일을 확인해 달라는 메시지를 띄운다.

3. 이메일 인증하기

이메일 인증 로그인 단계

1. 유저가 회원가입 폼을 통해 가입 신청

2. 필요한 값을 올바르게 채워 보냈다면 인증되지 않은 새로운 유저 생성

3. 해당 이메일로 인증 메일을 전송

4. 유저에게 이메일을 확인해 달라는 메시지를 띄운다.

5. 유저가 이메일에 첨부되어 있는 링크를 클릭한다.

3. 이메일 인증하기

이메일 인증 로그인 단계

1. 유저가 회원가입 폼을 통해 가입 신청

2. 필요한 값을 올바르게 채워 보냈다면 인증되지 않은 새로운 유저 생성

3. 해당 이메일로 인증 메일을 전송

4. 유저에게 이메일을 확인해 달라는 메시지를 띄운다.

5. 유저가 이메일에 첨부되어 있는 링크를 클릭한다.

6. 링크에 접속하면 인증이 완료되며 2번의 유저는 인증된 유저로 변경된다.

by. Yein

3. 이메일 인증하기

그래서 뭐부터 해야할까?

by. Yein

3. 이메일 인증하기

무슨 컬럼을 추가해야하지?

3. 이메일 인증하기

유저의 인증여부를 나타내는 컬럼을 추가해야한다!

3. 이메일 인증하기

그러나 친절한 디바이스님은 이미 작성해놓으심!

3. 이메일 인증하기

```
## Confirmable
t.string   :confirmation_token
t.datetime :confirmed_at
t.datetime :confirmation_sent_at
t.string   :unconfirmed_email # Only if using reconfirmable

## Lockable
t.integer  :failed_attempts, default: 0, null: false # Only if lock strategy is :failed_attempts
t.string   :unlock_token # Only if unlock strategy is :email or :both
t.datetime :locked_at
```

3. 이메일 인증하기

/app/models/user.rb

```
class User < ActiveRecord::Base
  # Include default devise modules. Others available are:
  # :confirmable, :lockable, :timeoutable and :omniauthable
  devise :database_authenticatable, :registerable,
         :recoverable, :rememberable, :trackable, :validatable, :confirmable
end
```

by. Yein

3. 이메일 인증하기

메일을 보내주는 STMP서버가 필요하다!

Simple Mail Transfer Protocol

전자 메일 전송을 위한 표준 프로토콜
이메일을 송수신하는 서버

3. 이메일 인증하기

그러나 오늘 우리는 간단하게 맛배기로!
Gmail을 우회해서 사용할거임!

3. 이메일 인증하기

/config/environments/development.rb

```
# devise : 이메일 인증 설정
  config.action_mailer.delivery_method = :smtp
  config.action_mailer.perform_deliveries = true
  config.action_mailer.default_url_options = { host: 'https://devise-silsub-bambi1859.c9users.io/' }
  ActionMailer::Base.smtp_settings = {
    :address              => 'smtp.gmail.com',
    :domain               => 'mail.google.com',
    :port                 => 587,
    :user_name            => ENV["Google_ID"],
    :password             => ENV["Google_PW"],
    :authentication       => 'login',
    :enable_starttls_auto => true
  }
```

3. 이메일 인증하기

/config/environments/development.rb

```
# devise : 이메일 인증 설정
  config.action_mailer.delivery_method = :smtp
  config.action_mailer.perform_deliveries = true
  config.action_mailer.default_url_options = { host: 'https://devise-silsub-bambi1859.c9users.io/' }
  ActionMailer::Base.smtp_settings = {
    :address              => 'smtp.gmail.com',
    :domain               => 'mail.google.com',
    :port                 => 587,
    :user_name            => ENV["Google_ID"],
    :password             => ENV["Google_PW"],
    :authentication       => 'login',
    :enable_starttls_auto => true
  }
```

"내프로젝트도메인"

3. 이메일 인증하기

/config/environments/development.rb

```
# devise : 이메일 인증 설정
  config.action_mailer.delivery_method = :smtp
  config.action_mailer.perform_deliveries = true
  config.action_mailer.default_url_options = { host: 'https://devise-silsub-bambi1859.c9users.io/' }
  ActionMailer::Base.smtp_settings = {
    :address              => 'smtp.gmail.com',
    :domain               => 'mail.google.com',
    :port                 => 587,
    :user_name            => ENV["Google_ID"],
    :password             => ENV["Google_PW"],
    :authentication       => 'login',
    :enable_starttls_auto => true
  }
```

ENV 환경변수설정

3. 이메일 인증하기

Gemfile> gem 'figrao'

발급 받은 인증 정보를 보다 안전하게 관리하기 위해!

3. 이메일 인증하기

Gemfile> gem 'figrao'

$bundle install
$bundle exec figaro install

3. 이메일 인증하기

config>application.yml

```
Google_ID: "aaa"
Google_PW: "aaa"
```

3. 이메일 인증하기

.yml은 GITHUB에 올려지지 않는다.

config>application.yml

```
Google_ID: "aaa"
Google_PW: "aaa"
```

3. 이메일 인증하기

config>application.yml

```
Google_ID: "aaa"
Google_PW: "aaa"
```

"여기에 실제 key를 적어준다"

3. 이메일 인증하기

/config/environments/development.rb

```
31    # Don't care if the mailer can't send.
32    config.action_mailer.raise_delivery_errors = true
```

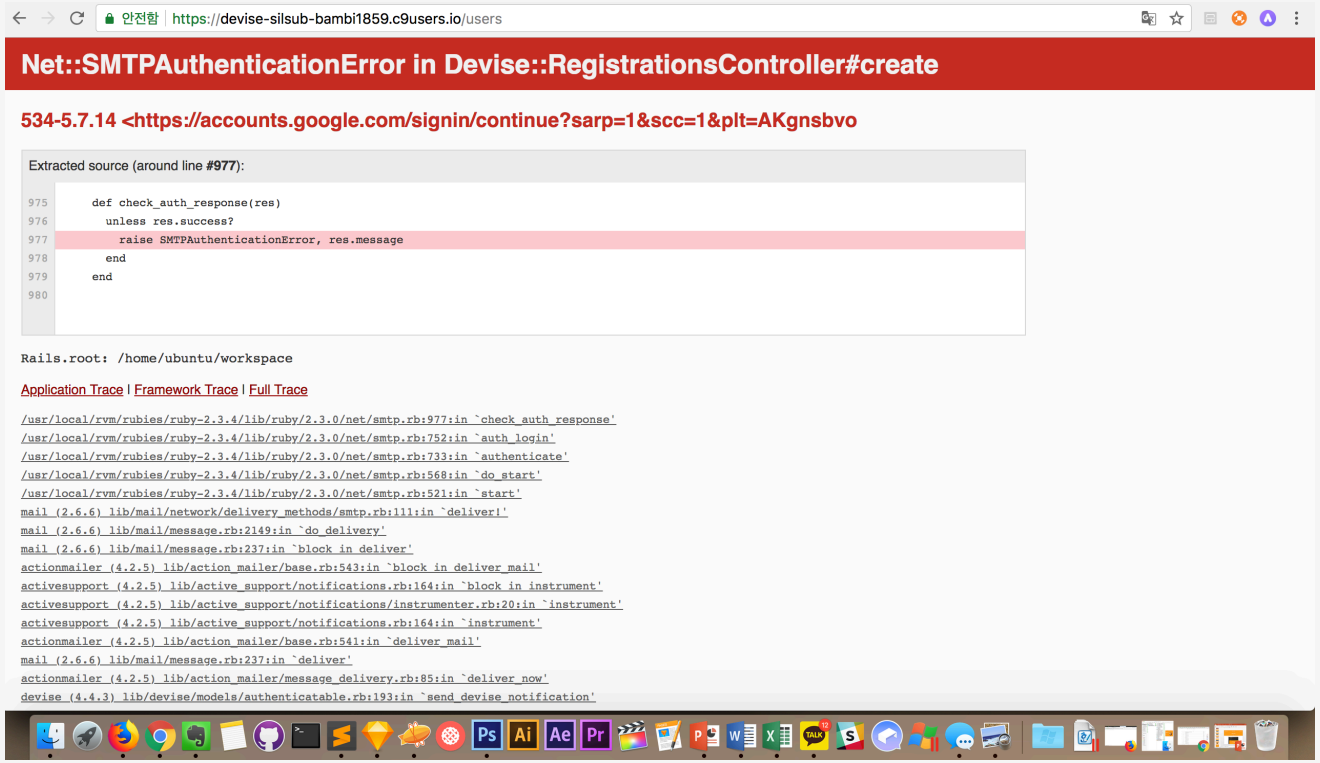
false에서 true로 바꿔준다.

by. Yein

3. 이메일 인증하기

이제 가입을 해보자!

3. 이메일 인증하기


안전함 | https://devise-silsub-bambi1859.c9users.io/users
Net::SMTPAuthenticationError in Devise::RegistrationsController#create
534-5.7.14 <https://accounts.google.com/signin/continue?sarp=1&scc=1&plt=AKgnsbvo
Extracted source (around line #977):
975 def check_auth_response(res)
976 unless res.success?
977 raise SMTPAuthenticationError, res.message
978 end
979 end
980
Rails.root: /home/ubuntu/workspace
Application Trace | Framework Trace | Full Trace
/usr/local/rvm/rubies/ruby-2.3.4/lib/ruby/2.3.0/net/smtp.rb:977:in `check_auth_response'
/usr/local/rvm/rubies/ruby-2.3.4/lib/ruby/2.3.0/net/smtp.rb:752:in `auth_login'
/usr/local/rvm/rubies/ruby-2.3.4/lib/ruby/2.3.0/net/smtp.rb:733:in `authenticate'
/usr/local/rvm/rubies/ruby-2.3.4/lib/ruby/2.3.0/net/smtp.rb:568:in `do_start'
/usr/local/rvm/rubies/ruby-2.3.4/lib/ruby/2.3.0/net/smtp.rb:521:in `start'
mail (2.6.6) lib/mail/network/delivery_methods/smtp.rb:111:in `deliver!'
mail (2.6.6) lib/mail/message.rb:2149:in `do_delivery'
mail (2.6.6) lib/mail/message.rb:237:in `block in deliver'
actionmailer (4.2.5) lib/action_mailer/base.rb:543:in `block in deliver_mail'
activesupport (4.2.5) lib/active_support/notifications.rb:164:in `block in instrument'
activesupport (4.2.5) lib/active_support/notifications/instrumenter.rb:20:in `instrument'
activesupport (4.2.5) lib/active_support/notifications.rb:164:in `instrument'
actionmailer (4.2.5) lib/action_mailer/base.rb:541:in `deliver_mail'
mail (2.6.6) lib/mail/message.rb:237:in `deliver'
actionmailer (4.2.5) lib/action_mailer/message_delivery.rb:85:in `deliver_now'
devise (4.4.3) lib/devise/models/authenticatable.rb:193:in `send_devise_notification'

3. 이메일 인증하기

1) 구글 Access 허용 : https://myaccount.google.com/u/0/lesssecureapps?pageId=none

2) 보안수준앱 접근 허용 : https://accounts.google.com/DisplayUnlockCaptcha

3. 이메일 인증하기

/app/views/devise/mailer/confirmation_instructions.html

승인 이메일 문구 바꾸기!

3. 이메일 인증하기

특정 이메일 주소만 가입시키기

/config/initializers/devise.rb

config.email_regexp = /₩A([₩w₩.%₩+₩-]+)@naver₩.com₩z/i

3. 이메일 인증하기

특정 이메일 주소만 가입시키기

이메일 작성 form.html

<%= f.text_field :email, pattern: "[A-Za-z0-9]{1,20}@naver.com" %>

3. 이메일 인증하기

특정 이메일 주소만 가입시키기

이메일 작성 form.html

<%= f.text_field :email, pattern: "[A-Za-z0-9]{1,20}@naver.com" %>

정규식이라는 녀석!

A~Z, a~z, 0~9까지 20자리 내로 문자입력

3. 이메일 인증하기

여기까지는 Development 모드 기준이다!

3. 이메일 인증하기

실제 서비스를 배포 (Production모드) 할 때!

1. 메일 보내는 양이 많아지면 정식(유료) STMP서버를 사용해야 한다.
2. 백그라운드로 처리할 수 있는 gem (Sidekiq, Resque)의 사용을 병행해야 메일을 보내는 동안 서버가 멈추지 않는다.
3. Figaro Gem 매우매우 중요하다 ! production mode설정도 해주어야 한다.
4. Devise를 한글로 번역해주는 devise-i18n 젬도 사용해 본다!