



**LIKE LION**

**EWHALION**

Gem – 이미지 업로드 & mailgun

이화여대 멋사 김준선

gem

루비에서 지원하는 '패키지 매니저'

다른 사람들이 만든 '패키지'들을 가져다 쓸 수 있도록!

gem

검증된 코드들이기 때문에  
Gem 을 사용하는 것이 훨씬 더 효율적이고 안전해!

gem

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

from 서강대 devise 강의자료 ☺

gem

필요한 기능이 있을 때마다  
“스스로” 찾아서 공부하고 적용하기!  
너무 많고 다양해요!

gem

그 중에서 오늘은

Carrierwave – 이미지 업로드

Mailgun – 이메일 전송



## 원래 사진 첨부를 하려면!

- 사진 서버 저장 및 사진 경로를 DB에 저장
- 파일 확장자 검사
- 외부 파일 저장소와의 연결(AWS 등)



복잡쓰...

# Image Upload

## CarrierWave

This gem provides a simple and extremely flexible way to upload files from Ruby applications. It works well with Rack based web applications, such as Ruby on Rails.

build passing coverage not found  [git.legal](https://git.legal)

<https://rubygems.org/gems/json>

<https://github.com/carrierwaveuploader/carrierwave>

참고해주세요 ☺



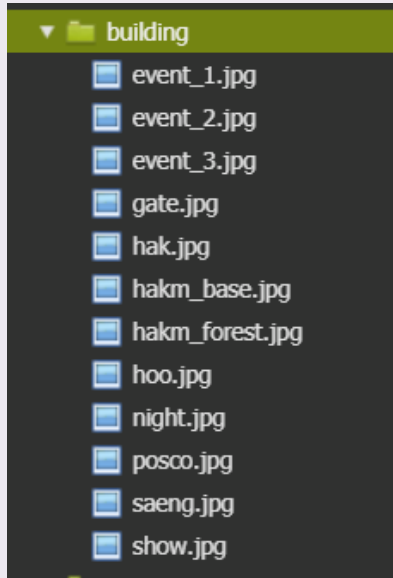
# Image Upload

사진 어떻게 올렸더라...?

```
<div class="line_height">  
  <%= image_tag ("building/#{@building}.jpg"), :class => "style_image" %>  
</div>
```

<img src= "/building/2.jpg">

# Image Upload



## <사진을 올리는 방식>

1. 올릴 사진을 서버에 저장한다
2. 서버에 저장한 사진의 경로를 가져온다.

```
<div class="line_height">  
  <%= image_tag ("building/#{@building}.jpg") :class => "style_image" %>  
</div>
```



Image upload

## 1. 기본세팅

[https://github.com/zzunvely/ewha\\_likelion\\_session.git](https://github.com/zzunvely/ewha_likelion_session.git)

클론해서 workspace 만들어주세요!

## 2. Gem 설치

```
gem 'carrierwave', '~> 1.0'
```



버전!

저장!

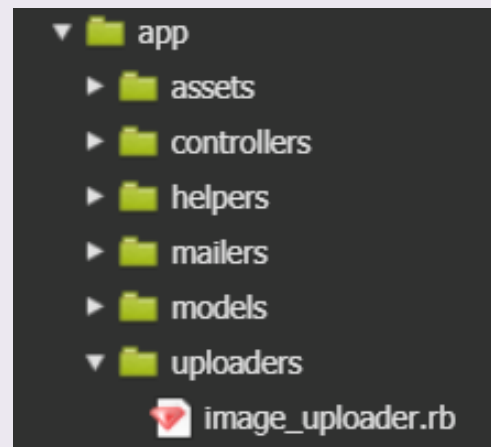
Gem 파일 수정했으면 => bundle install

# Image Upload

## 3. Uploader 만들기

```
zzunvely:~/workspace $ rails g uploader image  
Running via Spring preloader in process 2209  
create app/uploaders/image_uploader.rb
```

rails g uploader [업로더이름]



# Image Upload

## 3. Uploader 만들기

```
class ImageUploader < CarrierWave::Uploader::Base
  # Include RMagick or MiniMagick support:
  # include CarrierWave::RMagick
  # include CarrierWave::MiniMagick

  # Choose what kind of storage to use for this uploader:
  storage :file
  # storage :fog

  # Override the directory where uploaded files will be stored.
  # This is a sensible default for uploaders that are meant to be mounted:
  def store_dir
    "uploads/#{model.class.to_s.underscore}/#{mounted_as}/#{model.id}"
  end

  # Provide a default URL as a default if there hasn't been a file uploaded:
  # def default_url(*args)
  #   # For Rails 3.1+ asset pipeline compatibility:
  #   # ActionController::Base.helpers.asset_path("fallback/" + [version_name, "default.png"].compact.join('_'))
  #
  #   "/images/fallback/" + [version_name, "default.png"].compact.join('_')
  # end
```

사진을 어느 저장소에 저장할지 선택!

file: 내 서버 / fog : 외부 클라우드 서버(AWS)

# Image Upload

## 3. Uploader 만들기

```
class ImageUploader < CarrierWave::Uploader::Base
  # Include RMagick or MiniMagick support:
  # include CarrierWave::RMagick
  # include CarrierWave::MiniMagick

  # Choose what kind of storage to use for this uploader:
  storage :file
  # storage :fog

  # Override the directory where uploaded files will be stored.
  # This is a sensible default for uploaders that are meant to be mounted:
  def store_dir
    "uploads/#{model.class.to_s.underscore}/#{mounted_as}/#{model.id}"
  end

  # Provide a default URL as a default if there hasn't been a file uploaded:
  # def default_url(*args)
  #   # For Rails 3.1+ asset pipeline compatibility:
  #   # ActionController::Base.helpers.asset_path("fallback/" + [version_name, "default.png"].compact.join('_'))
  #
  #   "/images/fallback/" + [version_name, "default.png"].compact.join('_')
  # end
```

사진 저장할 폴더 지정!

마음대로 변경 가능!

## 3. Uploader 만들기

```
def extension_whitelist
  %w(jpg jpeg gif png)
end

# Override the filename of the uploaded files:
# Avoid using model.id or version_name here, see uploader/store.rb for details.
def filename
  "something.jpg" if original_filename
end
```

- 특정한 확장자만 허가할 것인지  
(이미지파일만 업로드 된다! 다른 종류의 파일 업로드 x)
- 올라오는 파일 이름 임의로 정할 수도 있어!



## 4. post테이블 수정하기

Think About img 태그!

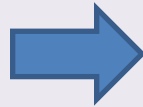
```

```

우리는 늘 경로를 적었어!  
테이블에도 '경로'를 적어주자!

## 4. post테이블 수정하기

id	title	content



id	title	content	Image

“Image : 이미지 파일의 경로” column 을 추가하자!

데이터 형식은 String!

## 4. post테이블 수정하기

### 1) Post 테이블에 사진을 위한 column추가

```
zzunvely:~/workspace $ rails g migration addImageToPosts image:string
Running via Spring preloader in process 2231
    invoke  active_record
    create  db/migrate/20180528050432_add_image_to_posts.rb
```

Rails g migration addImageToPosts image:string

Image칼럼을 Post테이블에 추가해줘!

Image칼럼의 데이터type은 string !

## 4. 모델 수정하기

### 2) db/migrate파일확인


```
class AddImageToPosts < ActiveRecord::Migration
  def change
    add_column :posts, :image, :string
  end
end
```

### 3) rake db:migrate

## 4. 모델 수정하기

4) add/model/post.rb

```
class Post < ActiveRecord::Base
  mount_uploader :image, ImageUploader
end
```



ImageUploader가

Post모델의 :image column 에서 작동할 수 있도록!

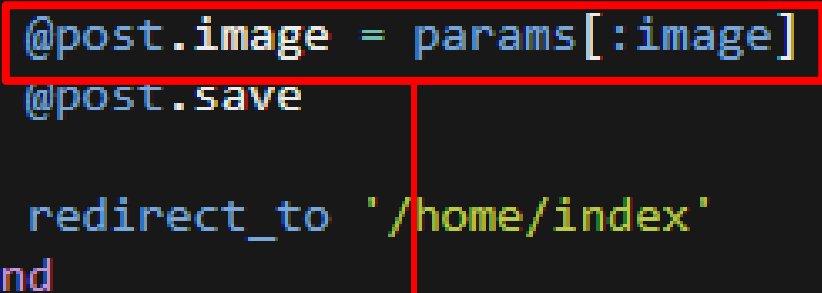
(사진 경로가 저장된 column이름 )

## 5. V&C 파일 업로드

### 1) 컨트롤러 수정하기

```
def create
  @post = Post.new
  @post.title = params[:post_title]
  @post.content = params[:post_content]
  @post.image = params[:image]
  @post.save

  redirect_to '/home/index'
end
```



title= 'image' 인 값을 받아와서 저장하기!

## 5. V&C 파일 업로드

### 2) new.html.erb

```
<h1>새로운 글을 써주세요!</h1>
<form action='/home/create' method='post' multipart="true">
  <%= hidden_field_tag :authenticity_token, form_authenticity_token %>
  제목 : <input type='text' name="post_title"><br><br>
  내용 : <textarea name = 'post_content'></textarea><br>
  사진 : <input type="file" name="image"><br>
  <input type = 'submit' value = '완료!'>
</form>
```

title= 'image' 인 값을 받아와서 저장하기!

Form 태그에게 파일형식의 데이터를  
전송할 거라고 해주자!

## 5. V&C 파일 업로드

### 3) list.html.erb

```
<%@posts.each do |p|%>
  제목 : <%=p.title%> <br>
  내용 : <%=p.content%><br>
  사진 : <img src = "<%=p.image.url%>" style="100px" ><br>
  <a href='/home/destroy/<%=p.id%>' style='text-decoration:none;'>[삭제]</a><br>
  <a href='/home/edit/<%=p.id%>' style='text-decoration:none;'>[수정]</a><br>
  -----<br>
<%end%>
```

title= 'image' 인 값을 받아와서 저장하기!



Image Upload

이미지 업로드는!  
뷰헬퍼로 바꿔주자!

# Image Upload

file_field	파일 선택 상자	<%= f.file_field :attach, size: 10, maxlength: 15 %>
		<input id="book_attach" maxlength="15" name="book[attach]" size="10" type="file" />

메소드	설명	예시
image_tag	이미지 태그 생성	<%= image_tag 'logo.png' %> 자동으로 assets 폴더 안에서 해당 사진을 불러옴
		<%= image_tag '/icons/button.png' %> icons 폴더 안에서 해당 사진을 불러옴
		<%= image_tag 'http://www.~.com/image/banner.png' %> 경로에 있는 사진 불러옴
		<%= image_tag 'sample.png', alt: 샘플, size: '120x90' %> 가로 120, 세로 90으로 사진 사이즈 지정

# Image Upload

## New.html.erb

```
<h1>새로운 글을 써주세요!</h1>
<%= form_tag("/home/create", method: "post", multipart: true) do%>
  <%= hidden_field_tag :authenticity_token, form_authenticity_token %>
  제목 : <input type='text' name="post_title"><br><br>
  내용 : <textarea name = 'post_content'></textarea><br>
  사진 <%=file_field_tag 'image', accept: 'image/*'%><br>

  <%= submit_tag "완료"%>
<%end%>
```

## Index.html.erb

```
<%@posts.each do |p|%>
  제목 : <%=p.title%> <br>
  내용 : <%=p.content%><br>
  사진 : <%= image_tag p.image.url %><br>
  <a href="/home/destroy/<%=p.id%>" style='text-decoration:none;'>[삭제]</a><br>
  <a href="/home/edit/<%=p.id%>" style='text-decoration:none;'>[수정]</a><br>
  -----<br>
<%end%>
```

나머지도 알맞게 고쳐보자!



devise

## 로그인 구현

세션 기능/ 유지

사용자 인증/확인

비밀번호/ 사용자 정보보안

...



Devise가 다 해줄거야 😊

devise

[Devise 팀스터디]

1. 로그인한 유저만 글을 쓸 수 있게
2. 내가 쓴 글만 삭제 할 수 있게
3. 칼럼추가

까지 했었지!

추가로..

하지만  
이메일인증 오류났었어ㅠㅠㅠ  
근데 사실 ...

Gem쓰면 더 쉽지롱!

# mailgun

mailgun 0.8

Mailgun library for Ruby

**VERSIONS:**

**0.11** - October 26, 2016 (32.5 KB)

**0.8** - April 09, 2014 (27.5 KB)

**0.7** - June 15, 2013 (15 KB)

**0.6** - April 09, 2013 (14 KB)

**0.5** - January 17, 2013 (12.5 KB)

*Show all versions (8 total) →*

<https://github.com/mailgun/mailgun-ruby>

<https://rubygems.org/gems/mailgun/versions/0.8>

읽어보시길 ☺



# mailgun

## 1. email 칸을 만들어주자 / 컨트롤러 설정!

New.html.erb

```
<h1>새로운 글을 써주세요!</h1>
<%= form_tag("/home/create", method: "post", multipart: true) do %>
  <%= hidden_field_tag :authenticity_token, form_authenticity_token %>
  제목 : <input type='text' name="post_title"><br><br>
  내용 : <textarea name = 'post_content'></textarea><br>
  이메일 : <input type="email" name="email" placeholder="이메일을 넣어주세요"><br>
  사진 : <%= file_field_tag 'image', accept: 'image/*' %>
  <%= submit_tag "완료"%>
<%end%>
```

Home controller

```
def create
  @post = Post.new
  @post.title = params[:post_title]
  @post.content = params[:post_content]
  @email=params[:email]
  @post.image = params[:image]
  @post.save
end
```

# mailgun

## 2. Gem install

```
gem 'mailgun-ruby', '~>1.1.6'
```

gem 'mailgun-ruby', '~>1.1.6' 추가하고

bundle install

mailgun

### 3. Mailgun 회원가입하기

<https://www.mailgun.com/> 회원가입하자!

Mailgun => 메일을 대신 보내주는 서비스

# mailgun



## Getting Started With Mailgun Is Free And Takes 57 Seconds

Company/Account Name

zzun

Your Name

zzun

Work Email

zzunvely@likelion.org

Password

.....

Confirm Password

.....

☐ Add payment info now

체크하지 마세요!

## 4. Git 에서 코드 가져오기

```
mg_client = Mailgun::Client.new("your-api-key")

message_params = {
  from: 'bob@example.com',
  to:   'sally@example.com',
  subject: 'The Ruby SDK is awesome!',
  text:   'It is really easy to send a message!'
}

result = mg_client.send_message('example.com', message_params).to_h!

message_id = result['id']
message = result['message']
```

# mailgun

```
require 'mailgun'
class HomeController < ApplicationController
  def index
    @posts = Post.all
  end

  def new
  end

  def create
    @post = Post.new
    @post.title = params[:post_title]
    @post.content = params[:post_content]
    @email = params[:email]
    @post.image = params[:image]
    @post.save

    mg_client = Mailgun::Client.new("your-api-key")

    message_params = {
      from: 'bob@example.com',
      to: 'sally@example.com',
      subject: 'The Ruby SDK is awesome!',
      text: 'It is really easy to send a message!'
    }

    result = mg_client.send_message('example.com', message_params).to_h!

    message_id = result['id']
    message = result['message']

    redirect_to '/home/index'
  end
end
```

# mailgun

[Domains](#)[Mailing Lists](#)[Logs](#)[Routes](#)[Analytics](#)[Suppressions](#)[Email Validation](#)

[Domains](#) / sandbox97f88776a0184775a1a51c2f7c4112d4.mailgun.org

DOMAIN

sandbox97f88776a0184775a1a51c2f7c4112d4.mailgun.org

## Domain Information

=> Result 'example.com' 부분에!

State	Active
IP Address	184.173.153.194 • <a href="#">Manage IPs</a>
SMTP Hostname	smtp.mailgun.org
Default SMTP Login	postmaster@sandbox97f88776a0184775a1a51c2f7c4112d4.mailgun.org
API Base URL	https://api.mailgun.net/v3/sandbox97f88776a0184775a1a51c2f7c4112d4.mailgun.org
Default Password	19f4559cd4b76145100e2d210486ac4a-b6183ad4-7cf4d351 • <a href="#">Manage SMTP credentials</a>
API Key	2869628a198d23ce3da022a442d420cf-b6183ad4-edd11ab3

=> Your 'api-key' 부분에!

# mailgun

```
require 'mailgun'

class HomeController < ApplicationController
  def index
    @posts = Post.all
  end

  def new
  end

  def create
    @post = Post.new
    @post.title = params[:post_title]
    @post.content = params[:post_content]
    @email = params[:email]
    @post.image = params[:image]
    @post.save

    mg_client = Mailgun::Client.new("2869628a198d23ce3da022a442d420cf-b6183ad4-edd11ab3")

    message_params = {
      from: 'ewha11@gmail.com',
      to: @email,
      subject: @post.title,
      text: @post.content
    }

    result = mg_client.send_message('sandbox97f88776a0184775a1a51c2f7c4112d4.mailgun.org', message_params).to_h!

    message_id = result['id']
    message = result['message']

    redirect_to '/home/index'
  end
end
```



mailgun

## 5. 메일을 보내자!

새로운 글을 써주세요!

제목 : 안녕하세요

내용 : 가주세요~

이메일 : zzunvely@naver.com

사진 : 파일 선택 선택된 파일 없음

완료

mailgun

성!공!



ewha11@gmail.com

안녕하세요



안녕하세요



보낸사람 <ewha11@gmail.com>

받는사람 <zzunvely@naver.com>

가주세여~

## Mailgun::CommunicationError in HomeController#create

**400 Bad Request: Please activate your Mailgun account. Check your inbox or log in to your control panel to resend the activation email.**

Extracted source (around line #27):

```
25         }  
26  
27         result = mg_client.send_message('sandbox97f88776a0184775a1a51c2f7c4112d4@mailgun.org', message_params).to_h!  
28  
29         message_id = result['id']  
30         message = result['message']
```

Rails.root: /home/ubuntu/workspace

[Application Trace](#) | [Framework Trace](#) | [Full Trace](#)

[app/controllers/home\\_controller.rb:27:in `create'](#)

혹시 이런 에러가 뜬다면!

메일 인증관련 오류니, 다른 메일로 보내주세요!

수고하셨습니다 😊

