

プラグイン名

Attachment_fuプラグイン

このプラグインができること

1. acts_as_attachmentの機能改良版。
2. ファイル(特に画像)のアップロードや表示を効率化できる
3. ファイルアップロード時に複数サイズのサムネイルを生成できる(要RMagick)
4. ファイルの保存先はファイルシステムかDBかを選択できる。AmazonのS3サービスにも対応しているらしい。
5. 画像表示時のヘルパーもついている

対象バージョン

1.2系

ちょー簡単な使い方

```
script/plugin install http://svn.techno-weenie.net/projects/plugins/attachment\_fu/
```

でインストールして、

```
script/generate model mugshot
```

とかで普通にモデルとマイグレーションファイルを用意。

```
class CreateMugshots < ActiveRecord::Migration
```

```
  def self.up
    create_table :mugshots do |t|
      t.column :parent_id, :integer
      t.column :content_type, :string
      t.column :filename, :string
      t.column :thumbnail, :string
      t.column :size, :integer
      t.column :width, :integer
      t.column :height, :integer
    end
  end
  def self.down
    drop_table :mugshots
  end
end
```

```
end
```

こんな感じのテーブル作って、マイグレートできたらモデルに

```
class Mugshot < ActiveRecord::Base
```

```
has_attachment :content_type => :image,  
  :storage => :file_system,  
  :max_size => 3000.kilobytes,  
#   :size => 0.megabyte..10.megabytes,  
#   :resize_to => '320x200',  
  :processor => 'Rmagick',  
  :thumbnails => { :thumb => '100x100', :small => '50x50' },  
  :path_prefix => "public/system/#{table_name}"  
validates_as_attachment
```

end

てな具合にアップロード時の設定を書いておいて、

```
class MainController < ApplicationController
```

```
  def new  
    @mugshot = Mugshot.new  
  end  
  def create  
    @mugshot = Mugshot.new(params[:mugshot])  
    if @mugshot.save  
      flash[:notice] = 'Mugshot was successfully created.'  
      redirect_to '/main/index'  
    else  
      render :action => :new  
    end  
  end  
  def index  
    @mugshots = Mugshot.find(:all,  
      :conditions => ["parent_id is null"]  
    )  
  end  
end
```

end

とかってコントローラーを作って、index用のviewは

```
<h1>Sample Index</h1>  
<% for mugshot in @mugshots -%>  
  
  <%= link_to image_tag(mugshot.public_filename(:small)), mugshot.public_filename %>  
  
<% end -%>  
<%= link_to 'new', {:action => :new} %>
```

new用のviewは

```
<%= error_messages_for :mugshot %>  
<% form_for(:mugshot, :url => 'create',
```

```
      :html => { :multipart => true }) do |f| -%>
    <p>
      <label for="mugshot">Upload A Mugshot:</label>
      <%= f.file_field :uploaded_data %>
    </p>
    <p>
      <%= submit_tag 'Create' %>
    </p>
  <% end -%>
  <%= link_to 'index', {:action => :index} %>
```

で使えるはず。

(モロパクなので[cuspos diary Ruby on Railsファイルアップロードプラグイン attachment_fu](#)を参考にしてください)

公式ページ

- ▶ [どこなんでしょう?](#)

日本語解説ページ

- ▶ [cuspos diary Ruby on Railsファイルアップロードプラグイン attachment_fu](#)

基本的な使い方はこれでOK

日本にもユーザーが結構居るようで、ググればノウハウも結構溜まっているみたい。

外国語解説ページ

- ▶ [Mike Clark's Weblog File Upload Fu](#)

結構良い

のうほう

- ▶ こういうのこそポリモーフィック関連を使えば楽なのかも(未実践)
- ▶ ひとつのモデルに画像とかPDFとかExcelファイルとか入れても大丈夫なのかなあ?(実践中)
- ▶ RMagickがちゃんとインストールされているか心配なときは、以下のコードをirbなどで実行してみよう。ちなみに、カレントディレクトリにtest.jpgがある前提だ。

```
require 'rubygems'
require 'RMagick'
image = "test.jpg"
original = Magick::Image.read(image).first
resized = original.resize_to_fit(75,100)
resized.write("test_resized.jpg")
```

- ▶ 上の例で言うと、Articleモデル has_many(or has_one) :mugshotの時に、サムネイルのレコードはarticle_idが入って、parent_idがnullになるので、Articleモデル側では

```
has_many :mugshot, :condition => "parent_id is null"
```

を付けておくと幸せかも。(あまり自信無い)

▶ マイグレーションファイルの構造を載せておく

attachment_fu migrations

=====

Fields for attachment_fu metadata tables...

```
in general:
  size,          :integer # file size in bytes
  content_type, :string  # mime type, ex: application/mp3
  filename,     :string  # sanitized filename
that reference images:
  height,       :integer # in pixels
  width,        :integer # in pixels
that reference images that will be thumbnailed:
  parent_id,    :integer # id of parent image (on the same table, a self-referencing foreign-key).
                  # Only populated if the current object is a thumbnail.
  thumbnail,   :string  # the 'type' of thumbnail this attachment record describes.
                  # Only populated if the current object is a thumbnail.
                  # Usage:
                  # [ In Model 'Avatar' ]
                  #   has_attachment :content_type => :image,
                  #                   :storage => :file_system,
                  #                   :max_size => 500.kilobytes,
                  #                   :resize_to => '320x200>',
                  #                   :thumbnails => { :small => '10x10>',
                  #                                     :thumb => '100x100>' }
                  # [ Elsewhere ]
                  # @user.avatar.thumbnails.first.thumbnail #=> 'small'
that reference files stored in the database (:db_file):
  db_file_id,   :integer # id of the file in the database (foreign key)
```

Field for attachment_fu db_files table:

```
data, :binary # binary file data, for use in database file storage
```



[WWW SQL Designerファイル](#)

コメント

名前:

書き込む

トラックバック

[トラックバック表示](#)