File Uploads / Carrier Wave

intro

we've used images in our apps before for things like backgrounds, but what if we want to attach an image to a specific instance of a class

Q: ask for examples of when we might need to upload an image and attach it to a model

- eg: adding profile picture to a User's profile
- rental site - adding images of houses etc

Objectives:

1. upload images and attach to a model using Carrierwave

- from our computer
- from a URL

2. edit images as we upload them using RMagick

Carrier Wave is a 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.

Wiki page is available here.

Create a simple rails app with a Recipe model

```
rails new carrierwave_app -d postgresql -B -T
cd carrierwave_app
rails g scaffold Recipe name
rake db:migrate
subl .
```

Change routes: app/config/routes.rb

```
Rails.Application.routes.draw do
  root 'recipes#index'
  resources :recipes
end
```

Then:

```
rails s
```

We will need to add the gem to our gemfile.

```
gem 'carrierwave', '0.9.0'
```

Now we will need to run bundle:

```
bundle
```

Generate Uploader

Now we will generate an uploader.

```
rails g uploader RecipeImage
```

This will create a file for us at: `app/uploaders/recipe_image_uploader.rb`

Create migration for image

When you run your app, you will likely serve your assets from a cdn. A content delivery network or content distribution network (CDN) is a large distributed system of servers deployed in multiple data centers across the Internet.

We will need some way of linking the image we upload to the space in our database. We will generate a migration for this.

```
rails g migration AddRecipeImageToRecipes recipe_image:string
```

This will create a migration which will add a RecipeImage column to our Recipes table. Now migrate your database.

```
rake db:migrate
```

Then add recipe*image to recipe*params, in recipes controller so it looks like this:

```
def recipe_params
  params.require(:recipe).permit(:name, :recipe_image)
end
```

Mount uploader

Now we will mount the uploader to our recipe. In the recipe model add the uploaded definition and the accessible attribute:

```
class Recipe < ActiveRecord::Base
  mount_uploader :recipe_image, RecipeImageUploader
end
```

Here we are specifying the table column followed by the uploader class.

Add file input to view

Now if we go into view, and recipes form, we can add:

```
<div class="field">
    <%= f.label :recipe_image %><br />
    <%= f.file_field :recipe_image %>
</div>
```

Add image to view

In the recipe/show page and add the new list item.

```
<p>
  <b>Image:</b>
  <%= image_tag @recipe.recipe_image.url %>
</p>
```

In the recipe/index, add: <%= image*tag recipe.recipe*image.url %>

We will now be able to upload your images through the new recipe form. These images will be stored at: public/uploads/recipe/recipe_image/ In the db it will be the filename.

investigate *Recipe.last.recipe_image* in rails console

Remote Image

If you want to add a remote image (from a website that uses HTTP) then all you need to do is to add `remote_recipe_image_url` to recipe_params, in recipes controller so it looks like this:

```
def recipe_params
  params.require(:recipe).permit(:name, :recipe_image, :remote_recipe_image_url)
end
```

Then in the form file, add:

```
<div class="field">
    <%= f.label :remote_recipe_image_url %><br />
    <%= f.text_field :remote_recipe_image_url %>
</div>
```

try uploading with this image: http://99livebirds.com/wp-content/uploads/2014/01/01651.jpg

And carrierwave will do the rest for you!

Test

Style it with CSS in assets/stylesheets/recipes.css

```
img {
  width: 100px;
  height: 100px;
  border: 1px solid #6B1229;
}
```

But it's squished??

Install rmagick

In order to manipulate our images, we will need to use the gem 'rmagick'

`gem 'rmagick'`

We will first need to install imagemagick:

`brew install imagemagick`

And then bundle:

`bundle`

(if this breaks with an error finding 'MagickWand.h' then `brew install pkg-config` and bundle again)

Edit Uploader

Now we can go into our recipe*image*uploader.rb and define some versions.

We can first uncomment the line:

`include CarrierWave::RMagick`

Now we can manipulate images. Now uncomment the lines:

```
version :thumb do
  process :scale => [50, 50]
end
```

And change it to:

```
version :thumb do
  process :resize_to_fill => [100, 100]
end
```

Now add these lines below your thumb version:

```
version :detail do
  process :resize_to_fit => [400, 400]
end
```

*_**inspect element here to show the difference in the two methods**_

Remove CSS

We will now need to remove the css on our images, to allow rmagick to do the work. Remove the styling from your css.

Now we can go back into our recipe/show view and update this line:

`<%= link_to image_tag(@recipe.recipe_image.thumb.url), @recipe.recipe_image.detail.url %>`

(upload a new image now, the old one will not be displayed)

This will render the thumbnail, and link to the 'detail' version we specified in recipe*image*uploader.rb.