[Lesson 5.1: File Uploads With Paperclip](#h.x8nxadm737lb)

[Gem of the Day](#h.5aki30pwi6vr)

[Git Branch](#h.z2s5gt941ibb)

[Review Setting Up Paperclip](#h.fpjzb9pjnffw)

[Optimizing Paperclip](#h.pfy5awbu30vg)

[Center and Crop](#h.dlunw67o2uj6)

[Limiting File Size](#h.o537pg9lc4j8)

[Git Branch and Merge](#h.8w2vjzdauhwx)

[Stepping Away From Default Rails Views](#h.y0k896a67tqr)

[Panels on Dog Index](#h.mpv8budk6faf)

[Default Image](#h.geztsopwsah3)

[Green Panels for Dogs in Daycare](#h.lz4gbkox1fdi)

[Classroom Challenge: More Views](#h.e3sbz86m9hg6)

# Lesson 5.1: File Uploads With Paperclip

## Gem of the Day

PlaceGant! <https://github.com/AngieGreen/placegant>

## Git Branch

(feature branches - create a new branch for paperclip)

## Review Setting Up Paperclip

Paperclip is an awesome gem that handles file uploads and manipulates images. First, you must have ImageMagick installed. See the Lesson 4.3 homework for more info.

Take a look at the Paperclip gem documentation. The readme file is actually quite detailed. It can be overwhelming, but when you take it step-by-step it’s not too bad.

1. First, we need to let Paperclip have access to ImageMagick. See [here](https://github.com/thoughtbot/paperclip#image-processor). First, run which convert. Grab that path and put it in config/environments/development.rb:  
    # Telling Paperclip gem where to find ImageMagick  
    Paperclip.options[:command\_path] = "/usr/local/bin/"
2. Add gem ‘paperclip’, bundle, commit.  
    gem "paperclip", "~> 4.2"
3. Now we can generate our Paperclip stuff. Paperclip conveniently comes with an automatic migration generator to add an avatar to a table. Read more [here](https://github.com/thoughtbot/paperclip#migrations). Run:  
    rails g paperclip dog avatar
4. Check the migration then run rake db:migrate. Add :avatar to the Dog controller whitelist.
5. Now, let’s add some validations to Dog for avatar for safety.  
    has\_attached\_file :avatar, :styles => { :medium => "300x300>", :thumb => "100x100>" }, :default\_url => "/images/:style/missing.png"  
    validates\_attachment\_content\_type :avatar, :content\_type => /\Aimage\/.\*\Z/
6. Now we can add the file upload to our dog form:  
    <div class="field">  
    <%= f.label :avatar %><br>  
    <%= f.file\_field :avatar %>  
    </div>
7. And to show:

<p>

<%= image\_tag @dog.avatar.url %>

<%= image\_tag @dog.avatar.url(:medium) %>

<%= image\_tag @dog.avatar.url(:thumb) %>

</p>

Run your rails server, upload a file for a dog, and test everything out. So cool! If everything worked, **commit your changes**. Work through any errors during office hours and over the weekend.

## Optimizing Paperclip

### Center and Crop

So far, Paperclip is really impressive. But, it would be even better if we could center and crop our photos down to a square. Go to your dogs model and edit it to look like this:

has\_attached\_file :avatar,

:styles => {

:medium => "300x300#",

:thumb => "100x100#"

},

:default\_url => "/images/:style/missing.png"

What did we do here? Default behavior is to resize the image and maintain aspect ratio . Some commonly used options are:

* **trailing #:** thumbnail will be centrally cropped, ensuring the requested dimensions.
* **trailing >:** thumbnail will only be modified if it is currently larger requested dimensions. (e.g the :small thumb for a 120×80 original image will be unchanged)

### Limiting File Size

We can also limit the file size by adding another validation:

validates\_attachment\_size :avatar, :less\_than => 3.megabytes,

:unless => Proc.new {|m| m[:avatar\_file\_name].blank?}

Once you’re satisfied with your changes, commit them.

## Git Branch and Merge

Now that we know paperclip is working, let’s merge our changes back into master.

As a review, we created a **branch** by typing “git branch branchname”. To see a full list of branches, we typed “git branch” alone. To start working on a different branch, first make sure you don’t have changes that need to be committed for the current branch, then type “git checkout branchname”.

Once you are finished developing on your branch and are ready to **merge** the changes back into master, first make sure you have committed your final changes, then:

git checkout master

git merge branchname --no-ff

The --no-ff flag stands for “no fast-forward” and will preserve our commit history from the old branch.

Once we are done merging, if we are finished with a branch, we typically delete it. You can delete a branch with:

git branch -d branchname

## Stepping Away From Default Rails Views

### Panels on Dog Index

We now have a really cool tool with the dog photos, but we’re not really using it to its full potential. These index pages are nice but kind of really ugly and not useful. Let’s make them pretty by getting away from the table format, and using Bootstrap divs and [panels](http://getbootstrap.com/components/#panels) instead!

Read about [panels](http://getbootstrap.com/components/#panels), and then let’s update our Dogs index.html.erb:

<h1>Our Doggies</h1>

<div class="row">

<% @dogs.each do |dog| %>

<div class="col-md-3">

<div class="panel panel-default">

<div class="panel-heading">

<h3 class="panel-title"><%= dog.name %> <%= dog.owner.last\_name %></h3>

</div>

<div class="panel-body">

Panel content

</div>

<div class="panel-footer">

In daycare? <%= dog.in\_daycare %><br>

<%= link\_to content\_tag(:i, nil, class: "fa fa-eye"), dog %>

<%= link\_to content\_tag(:i, nil, class: "fa fa-pencil"), edit\_dog\_path(dog) %>

<%= link\_to content\_tag(:i, nil, class: "fa fa-trash-o"), dog, method: :delete, data: { confirm: 'Are you sure?' } %>

</div>

</div>

</div>

<% end %>

</div>

Refresh your page. That’s so cool! Now let’s replace the panel content with our photos:

<%= image\_tag dog.avatar.url(:medium), class: "img-responsive" %>

### Default Image

So cool! But, it doesn’t look pretty if an image hasn’t yet been uploaded for a dog.

1. Find an image to use as a default image, and upload it so that Paperclip will generate all three versions of it for us. You can find it in the public folder (keep digging in all the subfolders).
2. Copy all three versions to your assets/images folder. Rename them in that folder to missing\_original, missing\_medium, and missing\_thumb, with the correct file type extension.
3. Remember in the model when we added validation? Take a look at that code. Edit the default\_url to this, except adjust the file type extension if your images were not jpgs:  
   :default\_url => "missing\_:style.jpg"
4. Commit your changes!

### Green Panels for Dogs in Daycare

We can even change the panel color based on whether a dog is in daycare using erb:

<div class="panel panel-default <%= 'panel-success' if dog.in\_daycare %>">

Comment out the In Daycare (both html and erb). Commit your changes!

### Classroom Challenge: More Views

On the Owner show view, show a list of dogs but as panels instead of a bulleted list. Make sure you have at least one link to go to that dog’s show page.

Make the Dog show view prettier. Pick one picture to show and reformat the view. Make the owner a link back to that owner’s show view.