

## Website Hosting - Feature #859

### Uploading Raw Files into Database With Ruby on Rails

10/14/2016 10:28 PM - Daniel Curtis

<b>Status:</b>	Closed	<b>Start date:</b>	10/16/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Daniel Curtis	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.50 hour
<b>Target version:</b>	Ruby	<b>Spent time:</b>	1.00 hour

#### Description

This is a simple guide for setting up a simple file upload to database with Ruby on Rails.

- Create the new Ruby on Rails web application:

```
rails new uploadr
```

- Change into the message\_app directory:

```
cd uploadr
```

- Start the rails development web server locally:

```
bundle exec rails s
```

- **NOTE:** The development web server bind the IP address to the localhost by default, so accessing it over the network will not work. To allow network access to the development web server:

```
bundle exec rails s -b 0.0.0.0
```

#### Document Scaffold

- Generate the scaffold model, view, and controller code for the document.

```
bundle exec rails generate scaffold document filename:string content_type:string file_contents:binary
```

- Edit the route config:

```
vi config/routes.rb
```

- And add the default route to the app:

```
Uploadr::Application.routes.draw do  
  resources :documents
```

```
root :to => 'documents#index'  
end
```

- Edit the form view:

```
vi app/view/documents/_form.html.erb
```

- And remove the unneeded **form div** tags, leaving the following:

```
<div class="field">  
  <%= f.file_field :file %>  
</div>  
<div class="actions">  
  <%= f.submit %>  
</div>
```

- Edit the document controller:

```
vi app/controllers/documents_controller.rb
```

- And the following right before the final `end` statement to allow **file** as a parameter so you can use it in the model:

```
def document_params  
  params.require(:document).permit(:file)  
end
```

- Next edit the document model:

```
vi app/models/document.rb
```

- And add the following to update the model to read and save the file

```
def initialize(params = {})  
  file = params.delete(:file)  
  super  
  if file  
    self.filename = sanitize_filename(file.original_filename)  
    self.content_type = file.content_type  
    self.file_contents = file.read  
  end  
end  
  
private  
def sanitize_filename(filename)  
  return File.basename(filename)  
end
```

## Showing the Document

- Edit the document controller again:

```
vi app/controllers/documents_controller.rb
```

- And modify the show method to send the raw file for viewing:

```
def show
  send_data(@document.file_contents,
            type: @document.content_type,
            filename: @document.filename)
end
```

## File Size Validation

- Edit the document model again:

```
vi app/controllers/documents_controller.rb
```

- And modify the file using the following code:

```
validate :max_file_size

def initialize(params = {})
  # File is now an instance variable so it can be validated.
  @file = params.delete(:file)
  super
  if @file
    self.filename = sanitize_filename(@file.original_filename)
    self.content_type = @file.content_type
    self.file_contents = @file.read
  end
end

NUM_BYTES = 1048576 # 1MB
def max_file_size
  if (@file.size.to_f / NUM_BYTES) > 1
    errors.add(:file, 'File size cannot be over #{NUM_BYTES} bytes.')
  end
end
```

## Resources

- <http://ryan.endacott.me/2014/06/10/rails-file-upload.html>
- <https://matt.berther.io/2007/10/19/uploading-files-to-a-database-using-rails/>
- <https://www.tutorialspoint.com/ruby-on-rails/rails-file-uploading.htm>
- <https://gist.github.com/macek/610596>

## History

#1 - 10/14/2016 10:31 PM - Daniel Curtis

- % Done changed from 0 to 100

- Status changed from New to Resolved

#2 - 10/14/2016 10:32 PM - Daniel Curtis

- Description updated

**#3 - 10/14/2016 10:33 PM - Daniel Curtis**

- *Description updated*

**#4 - 12/03/2016 12:49 PM - Daniel Curtis**

- *Status changed from Resolved to Closed*