UploadDocumentsTool

This Gem provides a tool which can be integrated easaly in a panel/dashboard of any kind. It is providing essentially three partial views which can be implemented anywhere in an app where this Gem is imported. The implementation is easy due the helper method made especially for render the views. The tool provides this three views:

```
- _documents.html.erb.  // This is the table showing all he documents
- _form.html.erb        // This is the upload form for upload the documents
- _form_error.html.erb  // This is the form error for show erros in the form by a
message
```

The tool autogenerate during the first upload a new folder under uploads called documents. Here all documents are stored.

Installation

Add this line to your application's Gemfile:

```
gem 'upload_documents_tool'
```

Or for local enviroment

```
gem 'uploade_document_tool', path: "<pathToTheGem>/uploade_document_tool"
```

And then execute:

```
$ bundle
```

Or install it yourself as:

```
$ gem install upload_documents_tool
```

Usage

To be able to use the Gem in the correct way we need the next dependencies:

- "railties", "~> 4.2.6"
- "jquery-rails"
- "kaminari", "~> 1.0.1"

- "bootstrap-kaminari-views", "~> 0.0.5"
- "bootstrap-sass"
- "jasny-bootstrap-rails"
- "font-awesome-sass"
- "toastr-rails"

After the Gem was succesfully installed in your system, you need to set up a table in your database where to store the uploaded documents. You can do that under the folder `./db/migrate/` and in your migration file you can add like next(Your migration file should be similar with different name and less fields):

```
class CreateDocuments < ActiveRecord::Migration
  def change
    create_table :documents do |t|
      t.string :filename
      t.string :content_type
      t.binary :file_contents

      t.timestamps null: false
    end
  end
end
```

Then run migration:

```
rake db:migrate
```

To be able to use the functionality of the Gem you need tu use the next implementation in your controller using this `@documents = ...` this calls the Gem:

```
def index
    @documents = Document.order(sort_column + " " + sort_direction).page(params[:page]).per(params[:limit])
  end
```

In this example we are setting up in the Index the table of all documents providing the sort, pagination and item per page to show.

For the validation checking we need to import in to a model the functionality of the tool. We can do like the next model example:

```
class Document < ActiveRecord::Base
  include UploadDocumentsTool
  validate :document_file_format
  before_save :upload_local
  validates_uniqueness_of :filename, message: "File exist"
end
```

The helper method inside the Gem provide an easy way to render the partials inside a prefered view.

Table of all documents:

```
<%= documents_table @documents %>
```

Upload form:

```
<%= document_form @document %>
```

The tool provide a column in the table with the date/time of the added document. However, be aware there is no time conversion the time is in UTC format. If you need to diplay your local time go to the file `config/application.rb` inside the file set your time zone like the example:

```
# Time zone
config.time_zone = 'Copenhagen'  // Change this to your time zone
```

Along this Gem is also provided a test application where to see the functionality in act:

UploadGemTest

Development

After checking out the repo, run `bin/setup` to install dependencies. Then, run `rake spec` to run the tests. You can also run `bin/console` for an interactive prompt that will allow you to experiment.

To install this gem onto your local machine, run `bundle exec rake install`. To release a new version, update the version number in `version.rb`, and then run `bundle exec rake release`, which will create a git tag for the version, push git commits and tags, and push the `.gem` file to rubygems.org.

Contributing

Bug reports and pull requests are welcome on GitHub at https://github.com/[USERNAME]/upload*documents*tool. This project is intended to be a safe, welcoming space for collaboration, and contributors are expected to adhere to the Contributor Covenant code of conduct.

License

The gem is available as open source under the terms of the MIT License.

Code of Conduct

Everyone interacting in the UploadDocumentsTool project’s codebases, issue trackers, chat rooms and mailing lists is expected to follow the code of conduct.