

Documents MVC Plan

I'm trying to figure out how to best organize an app I'm working on. I've don't have much have experience using nested resources but thought they might be a good solution in this case.

In my app, I have a Loan resource. Each Loan can have many Documents. The app will have two different types of Documents (though this number will increase later). Right now I'm using the same model for all documents. In the model, I've set the different Document types apart with a `kind` string field. The main place that the two Document types will differ is in their views, and especially their forms. For example, the OneOff documents form will allow the user to set a title, a description, and an attachment. The template form will allow the user to set a title and select from a list of template documents. I thought a solution might be to nest controllers under DocumentsController, and give these subcontrollers their own views.

Right now, I have a Documents controller. I've created a OneOffsController and a TemplatesController and both inherit from DocumentsController. Those two sub-controllers have their own create and update actions.

This is the controller directory structure:

- Controllers
 - DocumentsController
 - documents/
 - OneOffsController
 - TemplatesController
 - ...

This is the view directory structure:

- Views
 - Documents
 - OneOffs
 - _row.erb

- _edit.erb
 - _new.erb
 - _form.erb
- Templates
 - _row.erb
 - _edit.erb
 - _new.erb
 - _form.erb
- index.erb
- ...

At `loans/:id/documents/`, I have a table containing all documents. Something like this:

```
<% @documents.each do |document| %>
  <% if document.kind == 'oneoff' %>
    <%= render 'documents/oneoffs/row', document: document %>
  <% elsif document.kind == 'template' %>
    <%= render 'documents/templates/row', document: document %>
  <% end %>
<% end %>
```

Everything seemed to be going well until I decided to add an Edit modal to each row in the Documents table. Each row has an `Edit` button which should open a modal with its document type's respective `documents/[oneoffs|templates]/_edit.erb` partial rendered inside.

For example, if I click 'Edit' on a OneOff document's row in the `documents/index.erb` table, a modal should open with the `documents/oneoffs/_edit.erb` form partial rendered in the modal body:

```
<h1>Editing One-Off Document</h1>
<%= form_for document, url: {controller: "oneoffs", action: 'update'} do |f| %>
  <%= render 'layouts/errors', model: document %>

  <div class="form-group">
    <%= f.label :title, 'Document Title', class: 'control-label' %>
```

```

    <%= f.text_field :title, class: 'form-control' %>
  </div>

  <div class="form-group">
    <%= f.label :description, 'Description', class: 'control-label' %>
    <%= f.text_area :description, class: 'form-control' %>
  </div>

  <div class="form-group">
    <%= f.label :attachment, 'Attachment', class: 'control-label' %>
    <%= f.file_field :attachment, class: 'form-control' %>
  </div>

  <div class="actions">
    <%= f.submit class: 'btn btn-info' %>
  </div>
<% end %>
<%= link_to 'Back', mortgage_documents_path(document.mortgage_id) %>

```

So I have this modal working, but when trying to submit I get the error `No route matches {:action=>"update", :controller=>"oneoffs", :loan_id=>"1"}`

These are my routes right now for the Loan and Document resources:

```

resources :loans do
  resources :documents
  resources :oneoffs, module: 'documents' do
    collection do
      put :update
    end
  end
end
resources :templates, module: 'documents' do
  collection do
    put :update
  end
end

```

```

    end
  end
  member do
    post :archive
  end
end
end

```

I need to study up on routing and I'm not sure how to fix the update route right now.

So right now my questions are:

- How can I fix my routes so I can submit the /documents/oneoff/_edit.erb form from the /documents/index page? It should route to documents/oneoffs#update.
- In my routes.erb, should :oneoffs and :templates be nested under :documents like this?

```

○ resources :loans do
  resources :documents do
    resources :oneoffs, module: 'documents' do
      collection do
        put :update
      end
    end
  end
  resources :templates, module: 'documents' do
    collection do
      put :update
    end
  end
end
member do
  post :archive
end
end

```

- How can I fix the routes to get the edit form to submit?
- Does the way I'm setting this up make sense? Should I be inheriting from DocumentController? Should I be using nested resources? Or is there a better way to handle this Documents resource and their different but related types?
 - Would it make more sense to use a single Documents controller, but within the new action, have a conditional based on a document type param to render different forms? This sounds messy to me.
- Should the DocumentsController be inside the controllers/documents directory or with all the other top-level controllers?

