Ruby on Rails

Recursos Anidados

Overview

Recursos Anidados

Qué debería ir dentro de un controlador anidado o dependiente (nested/dependent controller)

Recursos Anidados (Nested Resources)

- Podemos hacer **scaffold** de un controlador RESTful regular para **notes**, **pero Note** es un recurso que depende del recurso **Book**.
- En otras palabras, cuando hacemos **cualquier cosa** con el recurso **notes**, **tenemos que** estar refiriéndonos a un **book** en **específico** para que esto sea **significativo**.
- Rails le llama a estos recursos secundarios "**Nested Resources**"

Creación de un controller vacío para notes

`rails g controller notes`

```
→  i_reviewed git:(master) ✗ rails g controller notes
Warning: You're using Rubygems 2.0.14 with Spring. Upgrade
-all` for better startup performance.
Running via Spring preloader in process 5440
      create  app/controllers/notes_controller.rb
              erb
      create    app/views/notes
              test_unit
      create    test/controllers/notes_controller_test.rb
              helper
      create    app/helpers/notes_helper.rb
                test_unit
              assets
                coffee
      create      app/assets/javascripts/notes.coffee
                scss
      create      app/assets/stylesheets/notes.scss
```

No se especifican actions

config/routes.rb

Se anida notes dentro de books!

Los paths a notes tendrán un book_id

routes.rb

```
1  Rails.application.routes.draw do
2    resources :books do
3      resources :notes
4    end
5
6    root to: "books#index"
7  end
8
```

Routes disponibles

```
→  i_reviewed git:(master) ✗ rake routes
       Prefix Verb   URI Pattern                                Controller#Action
   book_notes GET    /books/:book_id/notes(.:format)            notes#index
              POST   /books/:book_id/notes(.:format)            notes#create
new_book_note GET    /books/:book_id/notes/new(.:format)        notes#new
edit_book_note GET   /books/:book_id/notes/:id/edit(.:format)  notes#edit
    book_note GET    /books/:book_id/notes/:id(.:format)        notes#show
              PATCH  /books/:book_id/notes/:id(.:format)        notes#update
              PUT    /books/:book_id/notes/:id(.:format)        notes#update
              DELETE /books/:book_id/notes/:id(.:format)        notes#destroy
        books GET    /books(.:format)                           books#index
              POST   /books(.:format)                           books#create
     new_book GET    /books/new(.:format)                       books#new
    edit_book GET    /books/:id/edit(.:format)                  books#edit
         book GET    /books/:id(.:format)                       books#show
              PATCH  /books/:id(.:format)                       books#update
              PUT    /books/:id(.:format)                       books#update
              DELETE /books/:id(.:format)                       books#destroy
         root GET    /                                          books#index
```

Que actions necesitamos para Notes?

- Notes será mostrado en l**a página show de book** (books/show.html.erb), entonces **probablemente no necesitemos** los siete actions del notes controller.
- El formulario para crear un note (usualmente provisto por el action new) será provisto en la página show de book.
- Entonces, **create** y **destroy** son los dos actions que necesitamos.

Restringiendo routes

routes.rb

```
Rails.application.routes.draw do
  resources :books do
    resources :notes, only: [:create, :destroy]
  end
end
```

```
→  i_reviewed git:(master) ✗ rake routes
    Prefix Verb   URI Pattern                                Controller#Action
book_notes POST   /books/:book_id/notes(.:format)          notes#create
 book_note DELETE /books/:book_id/notes/:id(.:format)      notes#destroy
     books GET    /books(.:format)                         books#index
           POST   /books(.:format)                         books#create
  new_book GET    /books/new(.:format)                     books#new
 edit_book GET    /books/:id/edit(.:format)                books#edit
      book GET    /books/:id(.:format)                     books#show
           PATCH  /books/:id(.:format)                     books#update
           PUT    /books/:id(.:format)                     books#update
           DELETE /books/:id(.:format)                     books#destroy
      root GET    /                                        books#index
```

Notes Controller - before_action

Se busca book_id en set_book

routes.rb | notes_controller.rb

```ruby
class NotesController < ApplicationController
  before_action :set_book, only: [:create, :destroy]

  def create
    @note = @book.notes.new(note_params)
    if @note.save
      redirect_to @book, notice: "Note successfully added!"
    else
      redirect_to @book, alert: "Unable to add note!"
    end
  end

  def destroy
    @note = @book.notes.find(params[:id])
    @note.destroy
    redirect_to @book, notice: "Note deleted!"
  end

  private
  def set_book
    @book = Book.find(params[:book_id])
  end

  def note_params
    params.require(:note).permit(:title, :note)
  end
end
```

Entonces...

- Usualmente solo se necesitará agregar un subconjunto de los siete métodos RESTful de un controller.
- Los strong-parameters siguen vigentes para los subrecursos.