

# In this lecture, we will discuss...

- ✧ Nested Resources
- ✧ What should go into the nested/dependent controller

# Notes Resources

- ✧ We can **scaffold** a regular RESTful controller for notes as well, **but...**
- ✧ Note resource **depends** on the book resource
- ✧ In other words, when doing **anything** with the notes resources, we **have** to be talking about a **specific** book for it to be **meaningful**
- ✧ Rails calls such secondary resources “*Nested Resources*”



# Empty (for now) Notes controller

```
~/sandbox/i_reviewed$ rails g controller notes
  create  app/controllers/notes_controller.rb
  invoke  erb
  create  app/views/notes
  invoke  test_unit
  create  test/controllers/notes_controller_test.rb
  invoke  helper
  create  app/helpers/notes_helper.rb
  invoke  test_unit
  invoke  assets
  invoke  coffee
  create  app/assets/javascripts/notes.coffee
  invoke  scss
  create  app/assets/stylesheets/notes.scss
```

No actions specified during creation



# config/routes.rb

Nest notes inside books!

Paths to notes resources will have book id in them...

```
routes.rb
Rails.application.routes.draw do
  resources :books do
    resources :notes
  end

  root to: "books#index"
end
```



# Routes Available So Far - Books

HTTP Method	Named Routes	Parameters	Controller Action	Purpose
GET	books_path		index	List all
GET	book_path	ID	show	Show one
GET	new_book_path		new	Provide form to input new book
POST	books_path	Record hash	create	Create new record (in DB)
GET	edit_book_path	ID	edit	Provide form to edit book
PUT/PATCH	book_path	ID and Record hash	update	Update record (in DB)
DELETE	book_path	ID	destroy	Remove record



# Routes Available So Far - Notes

HTTP Method	Named Routes	Parameters	Controller Action	Purpose
GET	book_notes_path	Book ID	index	List all
GET	book_note_path	Book ID, Note ID	show	Show one
GET	new_book_note_path	Book ID	new	Provide form to input new note
POST	book_notes_path	Book ID, Record hash	create	Create new record (in DB)
GET	edit_book_note_path	Book ID, Note ID	edit	Provide form to edit note
PUT/ PATCH	book_note_path	Book ID, Note ID and Record hash	update	Update record (in DB)
DELETE	book_note_path	Book ID, Note ID	destroy	Remove record



# Rake Routes Is Your Friend

```
~/sandbox/i_reviewed$ 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



# What Actions Do We Need For Notes?

- ✧ Notes will be shown inline on the **book show page** (`books/show.html.erb`), so we probably **don't** need all seven actions in the notes controller
- ✧ The form for creating a note (usually provided by the `new` action) will be provided on the “book show” page
- ✧ `create` and `destroy` actions is **all** we need





# Restrict Note Routes

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

  root to: "books#index"
end
```

```
~/sandbox/i_reviewed$ 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 Using before\_action

```
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
```

Strong Parameters



# Summary

- ✧ You will usually need just a subset of the seven actions for the nested controller
- ✧ Remember “Strong Parameters”

## What's Next?

- ✧ Completing Nested Resources

