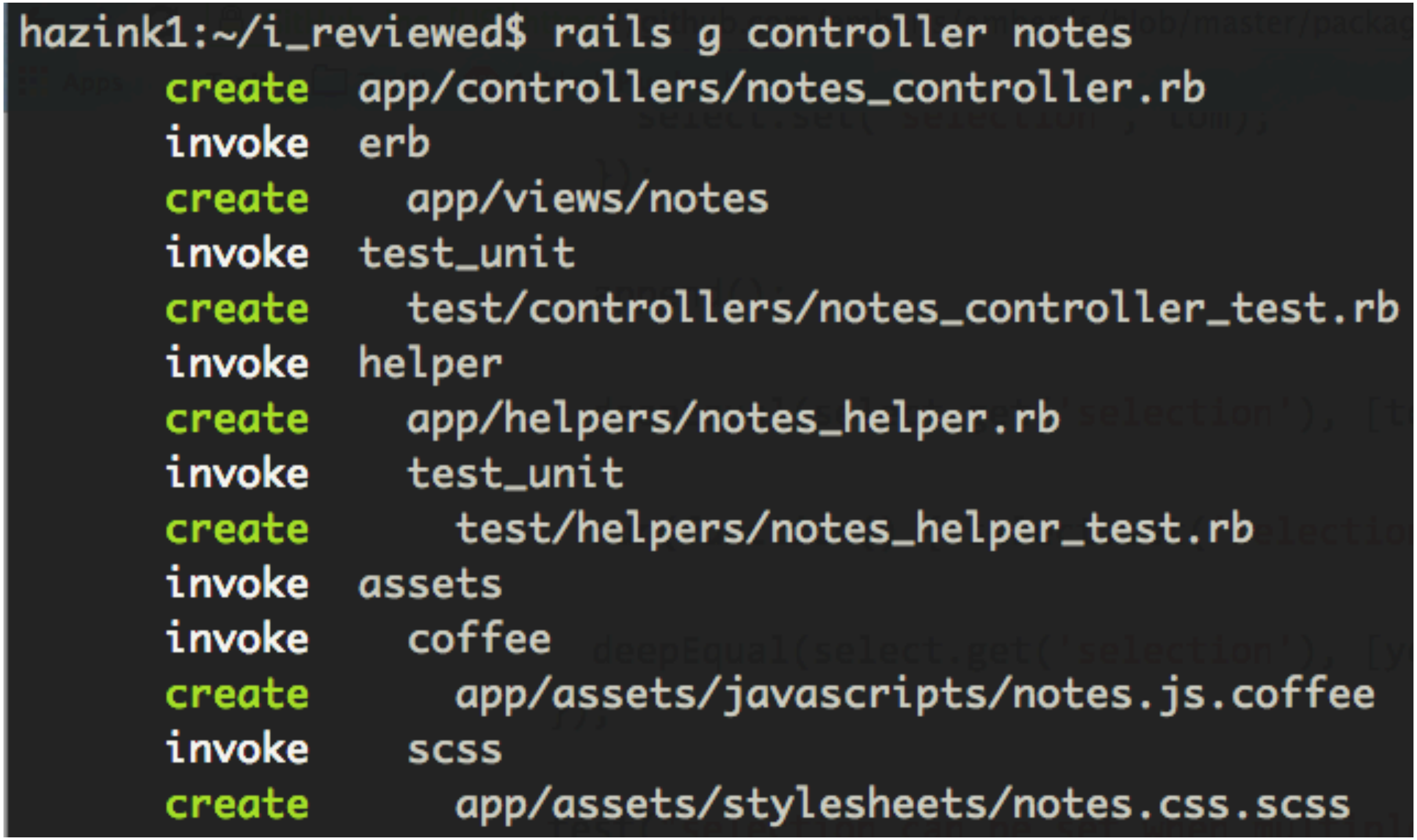
Nested Resources I

Notes Resources

- We can scaffold a regular RESTful controller for notes as well, but…
- Notes resources depend on the book resources
- 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

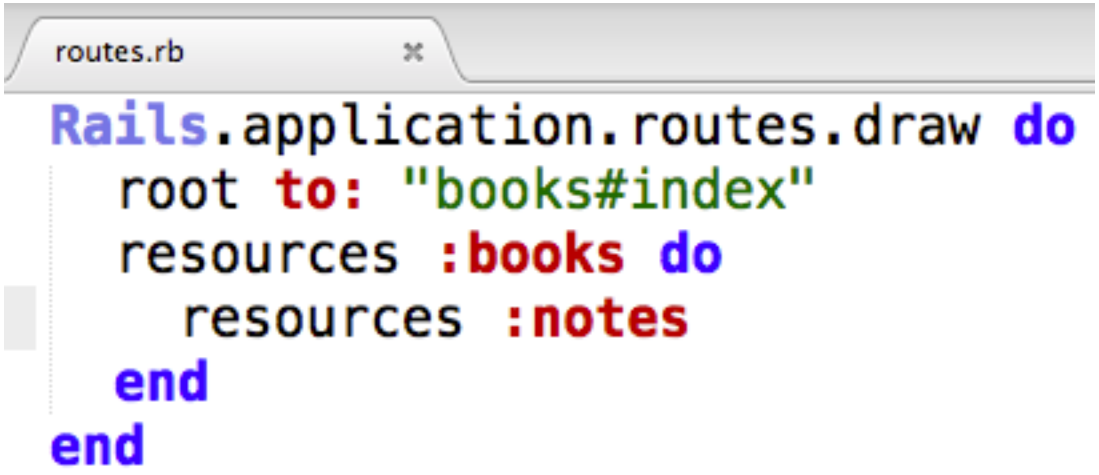


```
hazink1:~/i_reviewed$ rails g controller notes
      create  app/controllers/notes_controller.rb
      invoke  erb
      create    app/views/notes
      invoke  test_unit
      create    test/controllers/notes_controller_test.rb
      invoke  helper
      create    app/helpers/notes_helper.rb
      invoke    test_unit
      create      test/helpers/notes_helper_test.rb
      invoke  assets
      invoke    coffee
      create      app/assets/javascripts/notes.js.coffee
      invoke    scss
      create      app/assets/stylesheets/notes.css.scss
```

No actions specified during creation

config/routes.rb

Paths to notes resources will also have book id in them…



```ruby
Rails.application.routes.draw do
  root to: "books#index"
  resources :books do
    resources :notes
  end
end
```

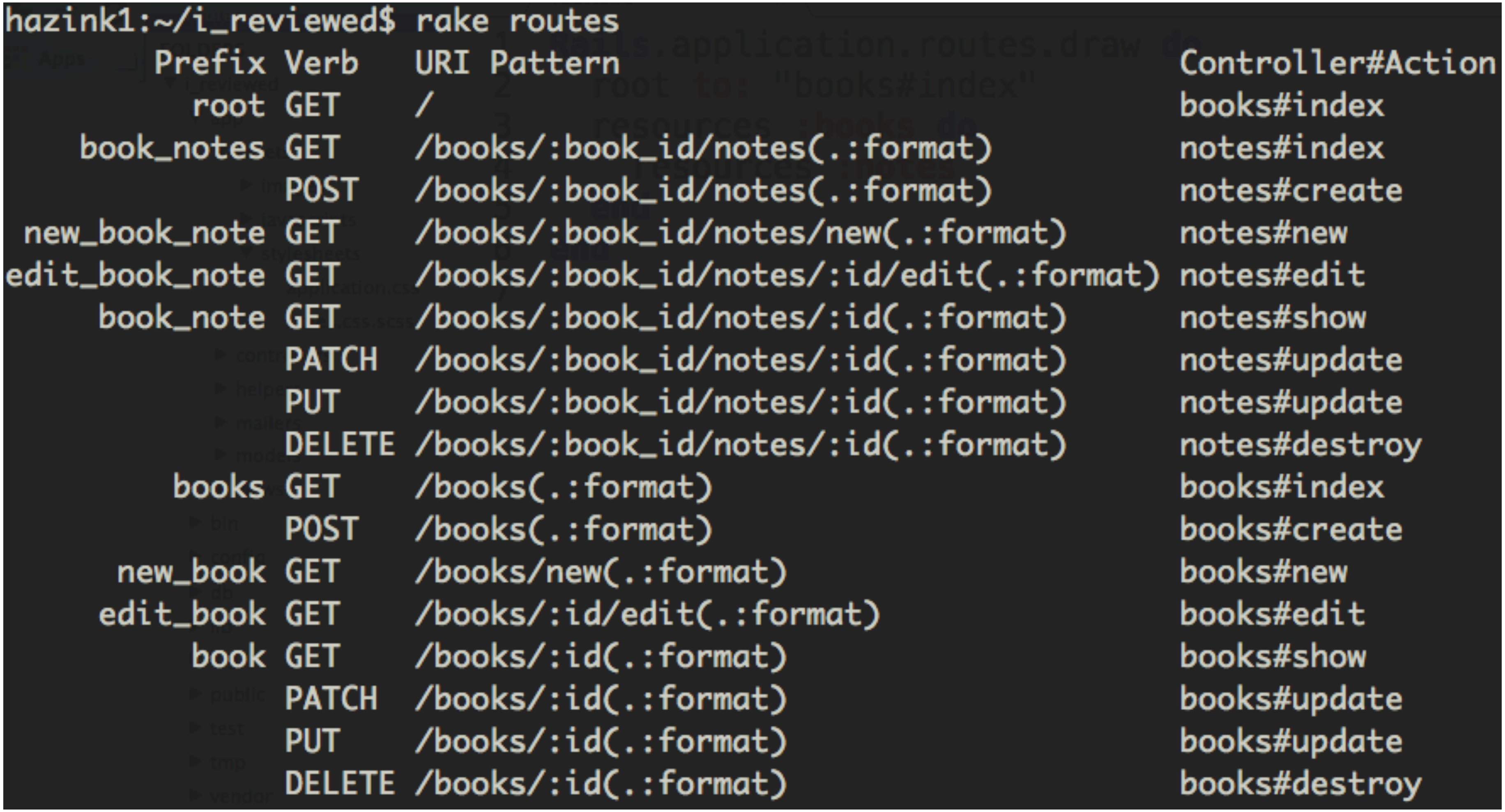
Routes available so far - Books

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

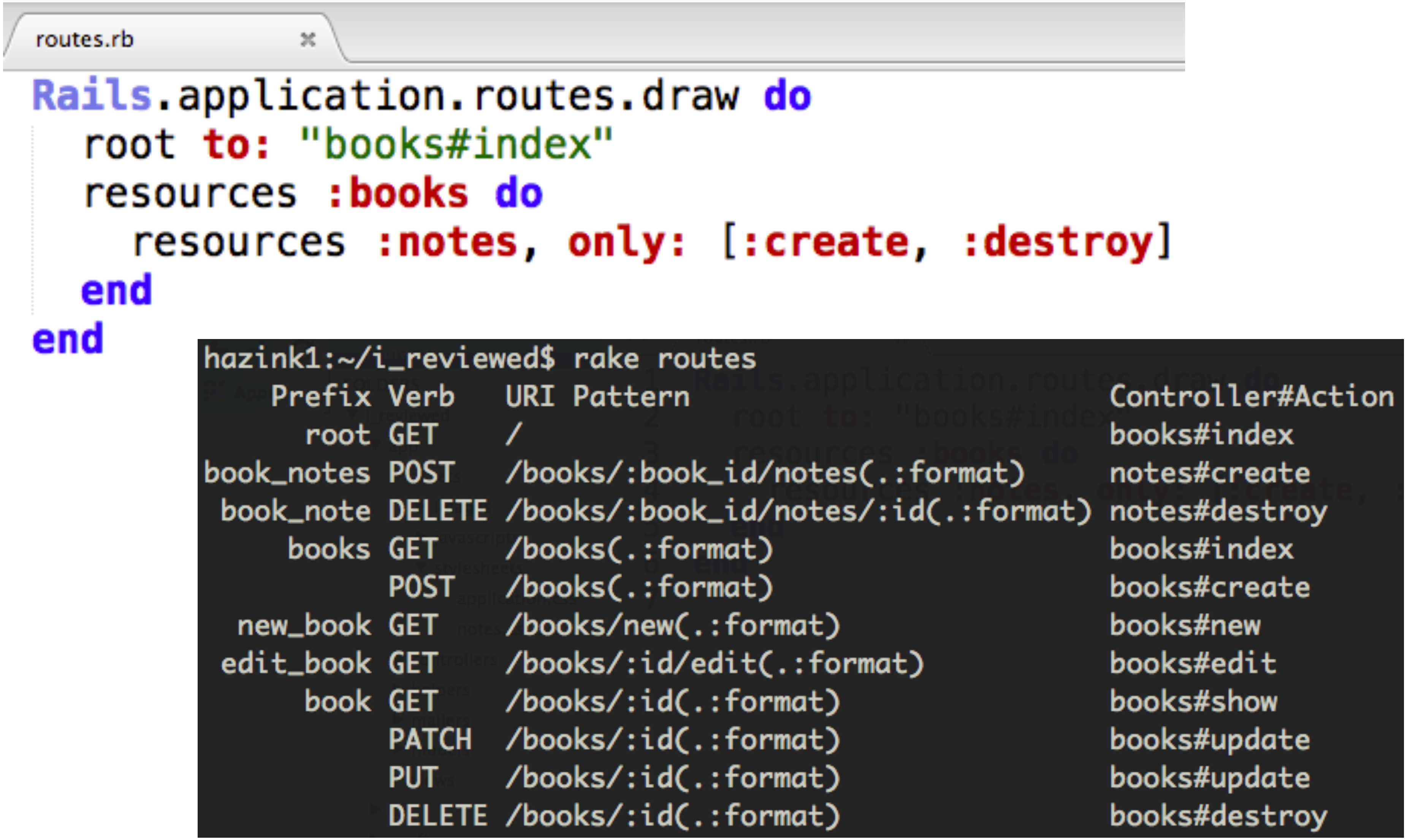


```
hazink1:~/i_reviewed$ rake routes
        Prefix Verb   URI Pattern                               Controller#Action
          root GET    /                                         books#index
    book_notes GET    /books/:book_id/notes(.:format)           notes#index
               POST   /books/:book_id/notes(.:format)           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
```

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 7 actions in the notes controller
- Probably don’t need the ability to edit/update a note (we can always add that later)
- The form for creating a note (usually provided by the `new` action) will be provided on the “book show” page as well
- `create` and `destroy` actions will do

Restrict note routes



routes.rb

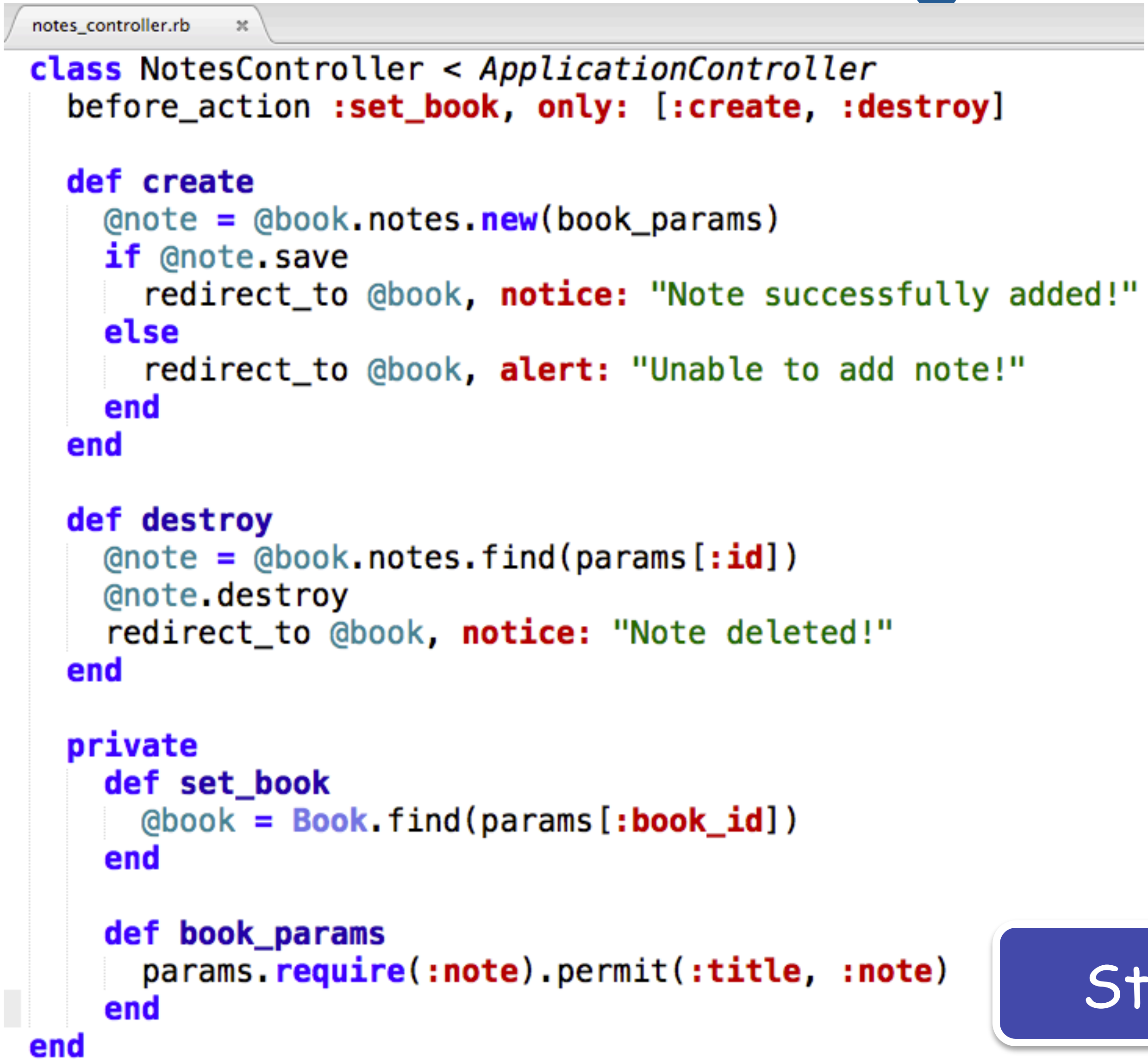
```ruby
Rails.application.routes.draw do
  root to: "books#index"
  resources :books do
    resources :notes, only: [:create, :destroy]
  end
end
```

```
hazink1:~/i_reviewed$ rake routes
    Prefix Verb   URI Pattern                          Controller#Action
      root GET    /                                    books#index
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
```

before_action (*review*)

- Runs the method specified before the actions in the controller
- Can specify `only:` or `except:`hash keys with an array as the value to restrict `before_action` to run (or not) before certain actions

Notes Controller using before_action



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

  def create
    @note = @book.notes.new(book_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 book_params
      params.require(:note).permit(:title, :note)
    end
end
```

Strong parameters