

# In this lecture, we will discuss...

- ❖ Building an app with multiple resources
- ❖ scaffold\_controller
- ❖ root\_to



# “iReviewed” App

- ✧ Let's write an app that lets a user **make notes** about the books he/she read
- ✧ We'll need three tables:
  - 1. Reviewers
    - name, password\_digest
  - 2. Books
    - name, author, reviewer\_id
  - 3. Notes
    - title, note, book\_id

```
$rails new i_reviewed  
$cd i_reviewed
```



# Creating The App From Scratch

```
~/sandbox/i_reviewed$ rails g model reviewer name password_digest -q
~/sandbox/i_reviewed$ rails g model book name author reviewer:references -q
~/sandbox/i_reviewed$ rails g model note title note:text book:references -q
~/sandbox/i_reviewed$ rake db:migrate
== 20151015232312 CreateReviewers: migrating ==
-- create_table(:reviewers)
  -> 0.0010s
== 20151015232312 CreateReviewers: migrated (0.0010s) ==

== 20151015232349 CreateBooks: migrating ==
-- create_table(:books)
  -> 0.0008s
== 20151015232349 CreateBooks: migrated (0.0008s) ==

== 20151015232430 CreateNotes: migrating ==
-- create_table(:notes)
  -> 0.0007s
== 20151015232430 CreateNotes: migrated (0.0008s) ==
```



# Specify Associations in the Models

The image shows a file structure on the left and three code files on the right.

**FOLDERS:**

- i\_reviewed
  - app
    - assets
    - controllers
    - helpers
    - mailers
  - models
    - concerns
    - .keep
    - book.rb
    - note.rb
    - reviewer.rb
  - views

**reviewer.rb**

```
1 class Reviewer < ActiveRecord::Base
2   has_many :books
3 end
```

**book.rb**

```
1 class Book < ActiveRecord::Base
2   belongs_to :reviewer
3   has_many :notes, dependent: :destroy
4 end
```

**note.rb**

```
1 class Note < ActiveRecord::Base
2   belongs_to :book
3 end
```



# Seeding With Some Data

The screenshot shows a code editor interface with a sidebar labeled "FOLDERS" on the left. The "seeds.rb" file is open in the main editor area. The code within the file is as follows:

```
1 Book.destroy_all
2
3 Book.create! [
4   { name: "Eloquent Ruby", author: "Russ Olsen" },
5   { name: "Beginning Ruby", author: "Peter Cooper" },
6   { name: "Metaprogramming Ruby 2", author: "Paolo Perrotta" },
7   { name: "Design Patterns in Ruby", author: "Russ Olsen" },
8   { name: "The Ruby Programming Language", author: "David Flanagan" }
9 ]
```



# Seeding With Some Data

```
~/sandbox/i_reviewed$ rake db:seed
~/sandbox/i_reviewed$ rails db
SQLite version 3.8.5 2014-08-15 22:37:57
Enter ".help" for usage hints.
sqlite> .headers on
sqlite> .mode columns
sqlite> select * from books;
id      name        author      reviewer_id      created_at          updated_at
-----  -----      -----      -----      -----      -----
1       Eloquent Ruby Russ Olsen      2015-10-15 23:47:13.953707 2015-10-15 23:47:13.953707
2       Beginning Rub Peter Coop      2015-10-15 23:47:13.959031 2015-10-15 23:47:13.959031
3       Metaprogrammi Paolo Perr      2015-10-15 23:47:13.960898 2015-10-15 23:47:13.960898
4       Design Patter Russ Olsen      2015-10-15 23:47:13.962642 2015-10-15 23:47:13.962642
5       The Ruby Prog David Flan      2015-10-15 23:47:13.964320 2015-10-15 23:47:13.964320
sqlite>
```



# Scaffold A Controller With Views

```
~/sandbox/i_reviewed$ rails g scaffold_controller book name author
  create  app/controllers/books_controller.rb
  invoke  erb
  create    app/views/books
  create    app/views/books/index.html.erb
  create    app/views/books/edit.html.erb
  create    app/views/books/show.html.erb
  create    app/views/books/new.html.erb
  create    app/views/books/_form.html.erb
  invoke  test_unit
  create    test/controllers/books_controller_test.rb
  invoke  helper
  create    app/helpers/books_helper.rb
  invoke  test_unit
  invoke  jbuilder
  create    app/views/books/index.json.jbuilder
  create    app/views/books/show.json.jbuilder
```



# config/routes.rb

**FOLDERS**

- ▼ **i\_reviewed**
  - ▶ **app**
  - ▶ **bin**
- ▼ **config**
  - ▶ **environment**
  - ▶ **initializers**
  - ▶ **locales**
  - ▶ **application**
  - ▶ **boot.rb**
  - ▶ **database.yml**
  - ▶ **environment**
  - routes.rb**

routes.rb

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

localhost:3000

## Listing Books

Name	Author	
Eloquent Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Beginning Ruby	Peter Cooper	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Metaprogramming Ruby 2	Paolo Perrotta	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Design Patterns in Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
The Ruby Programming Language	David Flanagan	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[New Book](#)



# Books Index Page (Root Of Our App)

```
index.html.erb      *

<p id="notice"><%= notice %></p>

<h1>Listing Books</h1>

<table>
  <thead> ...
  </thead>

  <tbody>
    <% @books.each do |book| %>
      <tr>
        <td><%= book.name %></td>
        <td><%= book.author %></td>
        <td><%= link_to 'Show', book %></td>
        <td><%= link_to 'Edit', edit_book_path(book) %></td>
        <td><%= link_to 'Destroy', book, method: :delete, data: { confirm: 'Are you sure?' } %></td>
      </tr>
    <% end %>
  </tbody>
</table>

<br>

<%= link_to 'New Book', new_book_path %>
```



# Pull out flash messages into layout

## FOLDERS

```
▼ i_reviewed
  ▶ app
    ▶ assets
    ▶ controllers
    ▶ helpers
    ▶ mailers
    ▶ models
  ▶ views
    ▶ books
  ▶ layouts
    application.html.erb
```

```
application.html.erb * index.html.erb * show.html.erb
1  <!DOCTYPE html>
2  <html>
3  <head> ...
8  </head>
9  <body>
10
11 <% flash.each do |key, value| %>
12   <p id='<%= key %>'><%= value %></p>
13 <% end %>
14
15 <%= yield %>
16
17 </body>
18 </html>
```

flash keys - :notice, :alert



# Summary

- ✧ scaffold\_controller is great for when you already have an existing model
- ✧ root\_to defines your root path
- ✧ Layout is great for common behavior

## What's Next?

- ✧ Nested Resources



# 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



# In this lecture, we will discuss...

- ❖ How to embed nested resources into the parent view



# books/show.html.erb – scaffolded

## FOLDERS

▼ i\_reviewed

▼ app

► assets

► controllers

► helpers

► mailers

► models

▼ views

▼ books

\_form.html.erb

edit.html.erb

index.html.erb

index.json.jbuilder

new.html.erb

show.html.erb

```
<p>
  <strong>Name:</strong>
  <%= @book.name %>
</p>
<p>
  <strong>Author:</strong>
  <%= @book.author %>
</p>
<%= link_to 'Edit', edit_book_path(@book) %> |
<%= link_to 'Back', books_path %>
```



# content\_tag helper

## ❖ content\_tag

- Nice Rails helper to generate HTML content

[http://api.rubyonrails.org/classes/ActionView/Helpers/TagHelper.html#method-i-content\\_tag](http://api.rubyonrails.org/classes/ActionView/Helpers/TagHelper.html#method-i-content_tag)

```
~/sandbox/i_reviewed$ rails c
Loading development environment (Rails 4.2.3)
irb(main):001:0> helper.content_tag :p, "Hello there"
=> "<p>Hello there</p>"
irb(main):002:0> helper.content_tag(:div, helper.content_tag(:p, "Cool"), class: "world")
=> "<div class=\"world\"><p>Cool</p></div>"
```



# views/books/show.html.erb

## FOLDERS

▼ i\_reviewed

▼ app

► assets

► controllers

► helpers

► mailers

► models

▼ views

▼ books

\_form.html.erb

edit.html.erb

index.html.erb

index.json.jbuilder

new.html.erb

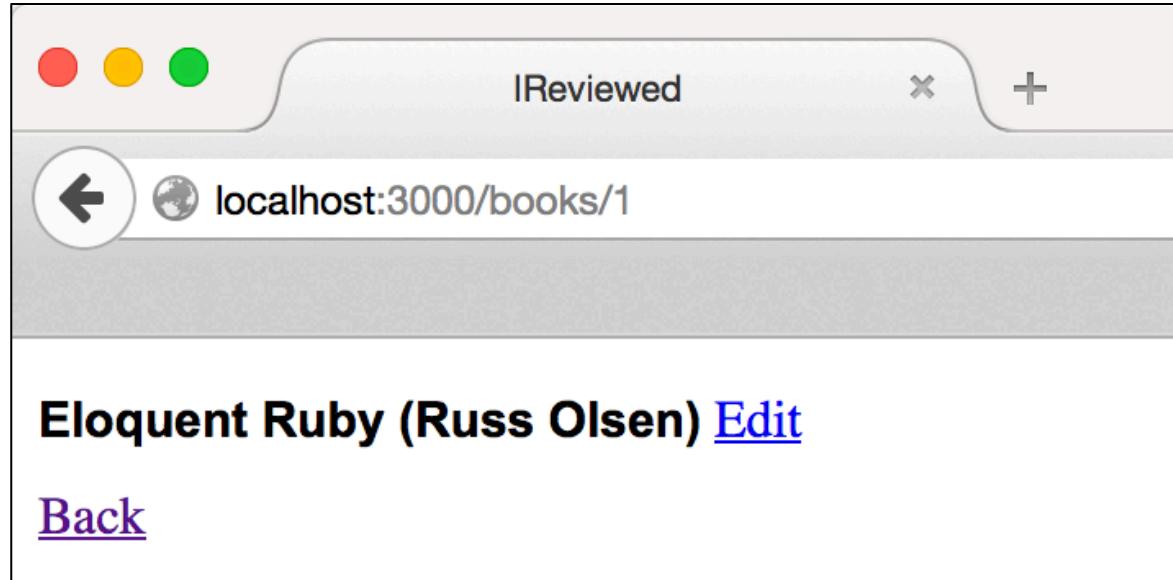
show.html.erb

show.html.erb \*

```
1 <div class="book">
2   <%= content_tag :span, "#{@book.name} (#{@book.author})", class: "book-title" %>
3   <%= link_to 'Edit', edit_book_path(@book) %>
4 </div>
5 |
6 <%= link_to 'Back', books_path %>
7
8
9
10
11
12
13
14
15
16
17
```



# views/books/show.html.erb



Looks lonely – let's add support for notes...



# Seed Some Notes

```
$ rake db:seed
```

The screenshot shows a code editor interface with a sidebar containing a file tree and a main pane displaying the contents of a file named `seeds.rb`.

**FOLDERS**

- ✓ i\_reviewed
  - ▶ app
  - ▶ bin
  - ▶ config
  - ▶ db
    - ▶ migrate
    - ▶ development
    - ▶ schema.rb
  - seeds.rb
  - ▶ lib
  - ▶ log
  - ▶ public

**seeds.rb**

```
1 Book.destroy_all
2
3 Book.create! [
4   { name: "Eloquent Ruby", author: "Russ Olsen" },
5   { name: "Beginning Ruby", author: "Peter Cooper" },
6   { name: "Metaprogramming Ruby 2", author: "Paolo Perrotta" },
7   { name: "Design Patterns in Ruby", author: "Russ Olsen" },
8   { name: "The Ruby Programming Language", author: "David Flanagan" }
9 ]
10 eloquent = Book.find_by name: "Eloquent Ruby"
11 eloquent.notes.create! [
12   { title: "Wow", note: "Great book to learn Ruby" },
13   { title: "Funny", note: "Doesn't put you to sleep" }
14 ]
15 ]
```



# views/notes/\_note.html.erb partial

FOLDERS

- ▼ i\_reviewed
- ▼ app
  - assets
  - controllers
  - helpers
  - mailers
  - models
- ▼ views
  - books
  - layouts
  - ▼ notes
    - \_note.html.erb

\_note.html.erb

```
1 <div class="note">
2   <div>
3     <%= content_tag :span, "#{note.title} (#{$time_ago_in_words note.created_at} ago)", 
4       | class: "note-title" %>
5     <%= link_to "Delete Note", [@book, note], method: :delete %>
6   </div>
7   <div><%= simple_format note.note %></div>
8 </div>
```

Formats new  
lines as <br/>



# views/books/show.html.erb

## FOLDERS

```
▼ i_reviewed
  ▶ app
    ▶ assets
    ▶ controllers
    ▶ helpers
    ▶ mailers
    ▶ models
  ▶ views
    ▶ books
      □ _form.html.erb
      □ edit.html.erb
      □ index.html.erb
      □ index.json.jbuilder
      □ new.html.erb
      □ show.html.erb
      □ show.json.jbuilder
    ▶ layouts
    ▶ notes
    ▶ sessions
  ▶ bin
  ▶ config
```

```
show.html.erb *
```

```
1  <div class="book">
2    <%= content_tag :span, "#{@book.name} (#{@book.author})", class: "book-title" %>
3    <%= link_to 'Edit', edit_book_path(@book) %>
4  </div>
5
6  <div id="notes">
7    <%= render @book.notes %>
8  </div>
9
10 <div>
11   <%= form_for [@book, @book.notes.new] do |f| %>
12     <div class="field">
13       <%= f.label :title %>
14       <%= f.text_field :title %>
15     </div>
16     <div class="field">
17       <%= f.label :note %>
18       <%= f.text_area :note, size: "25x5" %>
19     </div>
20     <div class="actions">
21       <%= f.submit "Add New Note" %>
22     </div>
23   <% end %>
24 </div>
25
26 <%= link_to 'Back', books_path %>
```

Note partials

Form for a new note



# books/show.html.erb

Note successfully added!

**Eloquent Ruby (Russ Olsen)** [Edit](#)

**Wow** (37 minutes ago) [Delete Note](#)

Great book to learn Ruby

**Funny** (less than a minute ago) [Delete Note](#)

Doesn't put you to sleep

Title

Note

[Add New Note](#)

[Back](#)



# CSS Used

FOLDERS

- ▼ i\_reviewed
- ▼ app
  - ▼ assets
    - ▶ images
    - ▶ javascripts
  - ▼ stylesheets
    - application.css
    - notes.scss
    - sessions.scss
- ▶ controllers
- ▶ helpers
- ▶ mailers
- ▶ models
- ▶ views
- ▶ bin

application.css

```
1  /* ... */
16
17  .book-title { font: bold 15px arial, sans-serif; }
18  .note-title { font: bold 12px arial, sans-serif; }
19  .book { padding-bottom: 10px; }
20
21  .note {
22      background: lightgray;
23      margin: 15px;
24      padding: 5px;
25      border: 1px solid;
26      border-radius: 5px;
27      width: 300px;
28  }
29
30  #notice { color: green; }
31  #alert { color: red; }
32  .field { margin: 5px; }
33  .actions { margin: 10px 0px; }
```



# Summary

- ✧ Remember to always pass the parent and the child into the `form_for` and `link_to` helpers
- ✧ Use `content_tag`, `simple_format` and various date-related helpers to simplify development

## What's Next?

- ✧ Authentication



# In this lecture, we will discuss...

- ✧ Authentication



# What Happened To The Users?

- ✧ The books/notes stuff is working pretty nicely, but everybody can **see** and **modify** everybody else's books and notes?
- ✧ We need the two security 'A's:
  - Authentication
    - Do we know **who** the user is and are his **credentials valid**?
  - Authorization
    - Provide **access** only to the books/notes a particular logged-in user is **authorized** to modify/see



# Authentication Summary

has secure password to the rescue!

1. Enable (uncomment) `bcrypt-ruby` (Gemfile)
  - Run \$bundle (install)
2. Make sure `password_digest` is table column
3. Account for `password` (not `password_digest`) inside **strong parameters** list in the controller if you plan to use mass assignment when creating users
4. No need for `password` column in the table (virtual attribute)



# bcrypt gem

FOLDERS

- ▼ i\_reviewed
  - app
  - bin
  - config
  - db
  - lib
  - log
  - public
  - test
  - tmp
  - vendor
  - .gitignore
  - config.ru
- Gemfile
- Gemfile.lock
- Rakefile

```
Gemfile
```

```
1 source 'https://rubygems.org'  
2  
3 gem 'rails', '4.2.3'  
4 gem 'sqlite3'  
5 gem 'sass-rails', '~> 5.0'  
6 gem 'uglifier', '>= 1.3.0'  
7 gem 'coffee-rails', '~> 4.1.0'  
8 gem 'jquery-rails'  
9 gem 'turbolinks'  
10 gem 'jbuilder', '~> 2.0'  
11 gem 'sdoc', '~> 0.4.0', group: :doc  
12  
13 # Use ActiveModel has_secure_password  
14 gem 'bcrypt', '~> 3.1.7'  
15  
16 group :development, :test do  
17   gem 'byebug'  
18   gem 'web-console', '~> 2.0'  
19   gem 'spring'  
20 end
```

\$bundle

Restart the server



# has\_secure\_password

**FOLDERS**

- ↳ i\_reviewed
  - ↳ app
    - ↳ assets
      - ▶ images
      - ▶ javascripts
      - ▶ stylesheets
    - ▶ controllers
    - ▶ helpers
    - ▶ mailers
  - ↳ models
    - ▶ concerns
      - .keep
      - book.rb
      - note.rb
      - reviewer.rb
  - ▶ views
  - ▶ bin
  - ▶ config
  - ↳ db
    - ▶ migrate
      - development.sqlite3
    - schema.rb
    - seeds.rb
  - ▶ lib

**reviewer.rb**

```
1 class Reviewer < ActiveRecord::Base
2   has_secure_password
3   has_many :books
4 end
```

**schema.rb**

```
1 # encoding: UTF-8
2 # This file is auto-generated from the current state of the database. Instead
3 # of editing this file, please use the migrations feature of Active Record to
4 # incrementally modify your database, and then regenerate this schema definition.
5 #
6 # Note that this schema.rb definition is the authoritative source for your
7 # database schema. If you need to create the application database on another
8 # system, you should be using db:schema:load, not running all the migrations
9 # from scratch. The latter is a flawed and unsustainable approach (the more migrations
10 # you'll amass, the slower it'll run and the greater likelihood for issues).
11 #
12 # It's strongly recommended that you check this file into your version control system.
13
14 ActiveRecord::Schema.define(version: 20151015232430) do
15   create_table "books", force: :cascade do |t|
16     t.string "title"
17   end
18
19   add_index "books", ["reviewer_id"], name: "index_books_on_reviewer_id"
20
21   create_table "notes", force: :cascade do |t|
22     t.string "text"
23   end
24
25   add_index "notes", ["book_id"], name: "index_notes_on_book_id"
26
27   create_table "reviewers", force: :cascade do |t|
28     t.string "name"
29     t.string "password_digest"
30     t.datetime "created_at", null: false
31     t.datetime "updated_at", null: false
32   end
33 end
```



# How It All Works

```
~/sandbox/i_reviewed$ rails c
Loading development environment (Rails 4.2.3)
irb(main):001:0> ActiveRecord::Base.logger = nil
=> nil
irb(main):002:0> Reviewer.column_names
=> ["id", "name", "password_digest", "created_at", "updated_at"]
irb(main):003:0> Reviewer.create! name: "Joe", password: "abc123"
=> #<Reviewer id: 1, name: "Joe", password_digest: "$2a$10$k7JtAb119soo3.ASpI0sqrERRFkzL3vxx1mPMhV423...", created_at: "2015-10-19 03:34:54", updated_at: "2015-10-19 03:34:54">
irb(main):004:0> joe = Reviewer.find_by name: "Joe"
=> #<Reviewer id: 1, name: "Joe", password_digest: "$2a$10$k7JtAb119soo3.ASpI0sqrERRFkzL3vxx1mPMhV423...", created_at: "2015-10-19 03:34:54", updated_at: "2015-10-19 03:34:54">
irb(main):005:0> joe.authenticate("somepassword")
=> false
irb(main):006:0> joe.authenticate("abc123")
=> #<Reviewer id: 1, name: "Joe", password_digest: "$2a$10$k7JtAb119soo3.ASpI0sqrERRFkzL3vxx1mPMhV423...", created_at: "2015-10-19 03:34:54", updated_at: "2015-10-19 03:34:54">
```



# Seed Users

```
◀ ▶ seeds.rb ×
1 Reviewer.destroy_all
2 Book.destroy_all
3
4 Book.create! [
5   { name: "Eloquent Ruby", author: "Russ Olsen" },
6   { name: "Beginning Ruby", author: "Peter Cooper" },
7   { name: "Metaprogramming Ruby 2", author: "Paolo Perrotta" },
8   { name: "Design Patterns in Ruby", author: "Russ Olsen" },
9   { name: "The Ruby Programming Language", author: "David Flanagan" }
10 ]
11
12 eloquent = Book.find_by name: "Eloquent Ruby"
13 eloquent.notes.create! [
14   { title: "Wow", note: "Great book to learn Ruby" },
15   { title: "Funny", note: "Doesn't put you to sleep" }
16 ]
17
18 reviewers = Reviewer.create! [
19   { name: "Joe", password: "abc123" },
20   { name: "Jim", password: "123abc" }
21 ]
22
23 Book.all.each do |book|
24   bookreviewer = reviewers.sample
25   book.save!
26 end
```



# Seed Users

```
~/sandbox/i_reviewed$ rake db:seed
~/sandbox/i_reviewed$ rails c
Loading development environment (Rails 4.2.3)
irb(main):001:0> Reviewer.first.books.count
  Reviewer Load (0.1ms)  SELECT "reviewers".* FROM "reviewers" ORDER BY "reviewers"."id" ASC LIMIT 1
  (0.1ms)  SELECT COUNT(*) FROM "books" WHERE "books"."reviewer_id" = ?  [["reviewer_id", 4]]
=> 2
irb(main):002:0> Reviewer.last.books.count
  Reviewer Load (0.2ms)  SELECT "reviewers".* FROM "reviewers" ORDER BY "reviewers"."id" DESC LIMIT 1
  (0.1ms)  SELECT COUNT(*) FROM "books" WHERE "books"."reviewer_id" = ?  [["reviewer_id", 5]]
=> 3
irb(main):003:0> Reviewer.first.books.pluck :name
  Reviewer Load (0.2ms)  SELECT "reviewers".* FROM "reviewers" ORDER BY "reviewers"."id" ASC LIMIT 1
  (0.1ms)  SELECT "books"."name" FROM "books" WHERE "books"."reviewer_id" = ?  [["reviewer_id", 4]]
=> ["Eloquent Ruby", "Metaprogramming Ruby 2"]
irb(main):004:0> Reviewer.last.books.pluck :name
  Reviewer Load (0.2ms)  SELECT "reviewers".* FROM "reviewers" ORDER BY "reviewers"."id" DESC LIMIT 1
  (0.1ms)  SELECT "books"."name" FROM "books" WHERE "books"."reviewer_id" = ?  [["reviewer_id", 5]]
=> ["Beginning Ruby", "Design Patterns in Ruby", "The Ruby Programming Language"]
irb(main):005:0> █
```



# Summary

- ✧ `has_secure_password`, `password_digest` column in a “users” table and `bcrypt` gem is all you need to get going

## What's Next?

- ✧ HTTP Sessions and cookies



# In this lecture, we will discuss...

- ❖ Sessions and cookies



# HTTP Is A Stateless Protocol

- ✧ HTTP is a **stateless** protocol
  - Each new request even from the same browser **knows nothing** about a previous request that was made
  - This means that even if a user makes a request – he will be treated as **unknown** on all the subsequent requests
- ✧ Cookies and Sessions to the rescue (keep state)

<http://guides.rubyonrails.org/security.html#what-are-sessions-questionmark>



# Sessions in Rails

- ❖ Session is created and made available through a `session` hash
- ❖ The server sends the browser a `cookie` with the session information, which the browser `stores` and `sends back` to the server on all subsequent requests (until the session ends)



# Restful Sessions Controller

- ✧ Session can be thought of as a **resource** – let's go ahead and create a RESTful sessions controller

```
~/sandbox/i_reviewed$ rails g controller sessions new create destroy -q  
~/sandbox/i_reviewed$ █
```



# config/routes.rb

FOLDERS

- ▼ i\_reviewed
  - app
  - bin
  - ▼ config
    - environment
    - initializers
    - locales
    - application
    - boot.rb
    - database.y
    - environment
    - routes.rb

routes.rb \*

```
1 Rails.application.routes.draw do
2   root to: "books#index"
3
4   resources :books do
5     resources :notes, only: [:create, :destroy]
6   end
7   resources :sessions, only: [:new, :create, :destroy]
8
9 end
10
11
12
13
14
15
```



# Sessions Controller Actions

- ✧ We can think of `new` action as **login** page and `destroy` as a **logout** page
- ✧ Thus, we'll need `new` (and `create`) actions to create a session and `destroy` action to destroy a session
- ✧ Let's **map login/logout routes** to make this more clear



# config/routes.rb

```
routes.rb

Rails.application.routes.draw do
  root to: "books#index"

  resources :books do
    resources :notes, only: [:create, :destroy]
  end
  resources :sessions, only: [:new, :create, :destroy]

  get "/login" => "sessions#new", as: "login"
  delete "/logout" => "sessions#destroy", as: "logout"
end
```

This lets us refer to these routes in our code as  
login\_path/logout\_path or login\_url/logout\_url...



# All Routes So Far

```
~/sandbox/i_reviewed$ rake routes
Prefix Verb URI Pattern Controller#Action
  root GET  /
book_notes POST /books/:book_id/notes(.:format) books#index
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
 sessions POST /sessions(.:format) sessions#create
new_session GET  /sessions/new(.:format) sessions#new
 session DELETE /sessions/:id(.:format) sessions#destroy
  login GET  /login(.:format) sessions#new
logout DELETE /logout(.:format) sessions#destroy
```



# Summary

- ✧ Sessions and cookies make the interaction between browser and server stateful
- ✧ You can think of Sessions as yet another resource
- ✧ Custom routes - <http://guides.rubyonrails.org/routing.html>

## What's Next?

- ✧ Sessions Controller and View



# In this lecture, we will discuss...

- ❖ Sessions controller and view
- ❖ Locking down our app



# views/sessions/new.html.erb

## FOLDERS

```
▼ i_reviewed
  ▼ app
    ▶ assets
    ▶ controllers
    ▶ helpers
    ▶ mailers
    ▶ models
  ▼ views
    ▶ books
    ▶ layouts
    ▶ notes
  ▼ sessions
```

new.html.erb

```
new.html.erb
1 <h1>Login</h1>
2
3 <%= form_for(:reviewer, url: sessions_path) do |f| %>
4
5   <div class="field"><%= f.label :name %> <br/> <%= f.text_field :name %></div>
6
7   <p/>
8
9   <div class="field"><%= f.label :password %> <br/> <%= f.password_field :password %></div>
10
11  <div class="actions"><%= f.submit "Login" %></div>
12 <% end %>
```



# Login Page

The screenshot shows a web browser window with a "Login" page. The browser's address bar displays "localhost:3000/login". The developer tools are open, showing the DOM tree. The current selection is on a "label" element within a "div.field" container.

**Developer Tools Header:** Inspector, Console, Debugger, Style Editor, Performance

**DOM Tree Selection Path:** html > body > form > div.field > label

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <h1>Login</h1>
    <form method="post" accept-charset="UTF-8" action="/sessions">
      <input type="hidden" value="✓" name="utf8">
      <input type="hidden" value="Nn7+7JIjyXpnEwnE56qGct3r4Kbm001eKvoQRqmCeA1c+FCL7yLXTPC1h5hxaLot+4ERUioRo/QGX300Y17G1Q==" name="authenticity_token">
      <div class="field">
        <label for="reviewer_name">Name</label>
        <br><br>
        <input id="reviewer_name" type="text" name="reviewer[name]">
      </div>
      <p></p>
      <div class="field">
        <label for="reviewer_password">Password</label>
        <br><br>
        <input id="reviewer_password" type="password" name="reviewer[password]">
      </div>
      <div class="actions">
        <input type="submit" value="Login" name="commit">
      </div>
    </form>
  </body>
</html>
```



# Sessions Controller

## FOLDERS

▼ i\_reviewed

  ▼ app

    ► assets

  ▼ controllers

    ► concerns

       application\_controller.rb

       books\_controller.rb

       notes\_controller.rb

       sessions\_controller.rb

    ► helpers

    ► mailers

    ► models

  ▼ views

    ► books

    ► layouts

    ► notes

  ▼ sessions

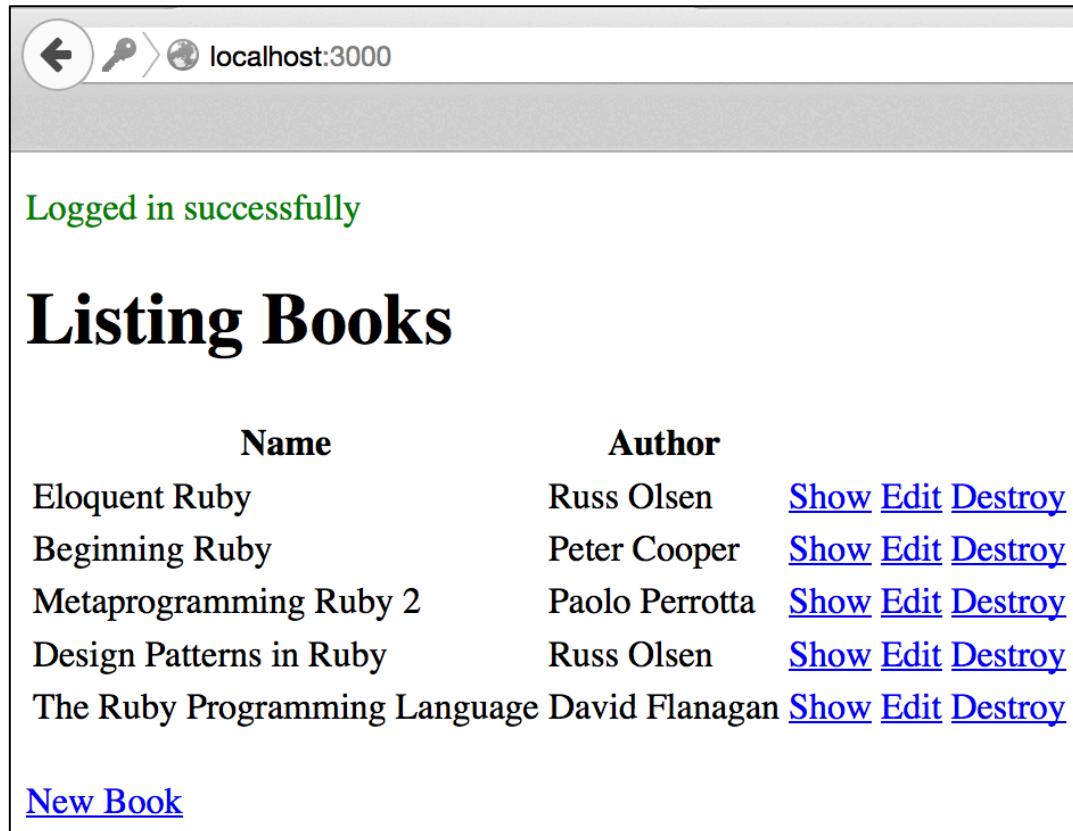
     create.html.erb

sessions\_controller.rb \*

```
1 class SessionsController < ApplicationController
2   def new
3     # Login Page - new.html.erb
4   end
5
6   def create
7     reviewer = Reviewer.find_by(name: params[:reviewer][:name])
8     password = params[:reviewer][:password]
9
10    if reviewer && reviewer.authenticate(password)
11      session[:reviewer_id] = reviewer.id
12      redirect_to root_path, notice: "Logged in successfully"
13    else
14      redirect_to login_path, alert: "Invalid username/password combination"
15    end
16  end
17
18  def destroy
19    reset_session # wipe out session and everything in it
20    redirect_to login_path, notice: "You have been logged out"
21  end
22 end
```



# Logged In



localhost:3000

Logged in successfully

## List of Books

Name	Author	
Eloquent Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Beginning Ruby	Peter Cooper	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Metaprogramming Ruby 2	Paolo Perrotta	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Design Patterns in Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
The Ruby Programming Language	David Flanagan	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[New Book](#)



# Cookie in the Browser

**Cookies**

Search:  (X)

The following cookies match your search:

Site	Cookie Name
localhost	_i_reviewed_session

Name: \_i\_reviewed\_session  
Content: dnNTTINmdU9BOXITaWhqTkNkck1IdERQQkpMbFZJbm5ScFlxRklzMzZ1MzZLZEJZM0hjZE05VjdUd0pla0RnclEvc3hqVXBESi8v  
Host: localhost  
Path: /  
Send For: Any type of connection  
Expires: At end of session

Remove Selected Remove All



# Locking Down The App

- ✧ We can have a `before_action` in the  `ApplicationController` (from which all the other controllers `inherit`) that will `make you login` if you are not yet logged in
- ✧ But if everything is `blocked off` – how will we get to the login page? Hmm...
- ✧ Controllers can `override before_action` with `skip_before_action`



# application\_controller.rb

## FOLDERS

```
▼ i_reviewed
  ▼ app
    ▶ assets
    ▼ controllers
      ▶ concerns
        ▄ application_controller.rb
        ▄ books_controller.rb
        ▄ notes_controller.rb
        ▄ sessions_controller.rb
    ▶ helpers
    ▶ mailers
    ▶ models
```

```
application_controller.rb *
```

```
1 class ApplicationController < ActionController::Base
2   # Prevent CSRF attacks by raising an exception.
3   # For APIs, you may want to use :null_session instead.
4   protect_from_forgery with: :exception
5
6   before_action :ensure_login
7
8   protected
9     def ensure_login
10       # Always go to login page unless session contains
11       # reviewer_id
12       redirect_to login_path unless session[:reviewer_id]
13     end
14 end
```



# sessions\_controller.rb

FOLDERS

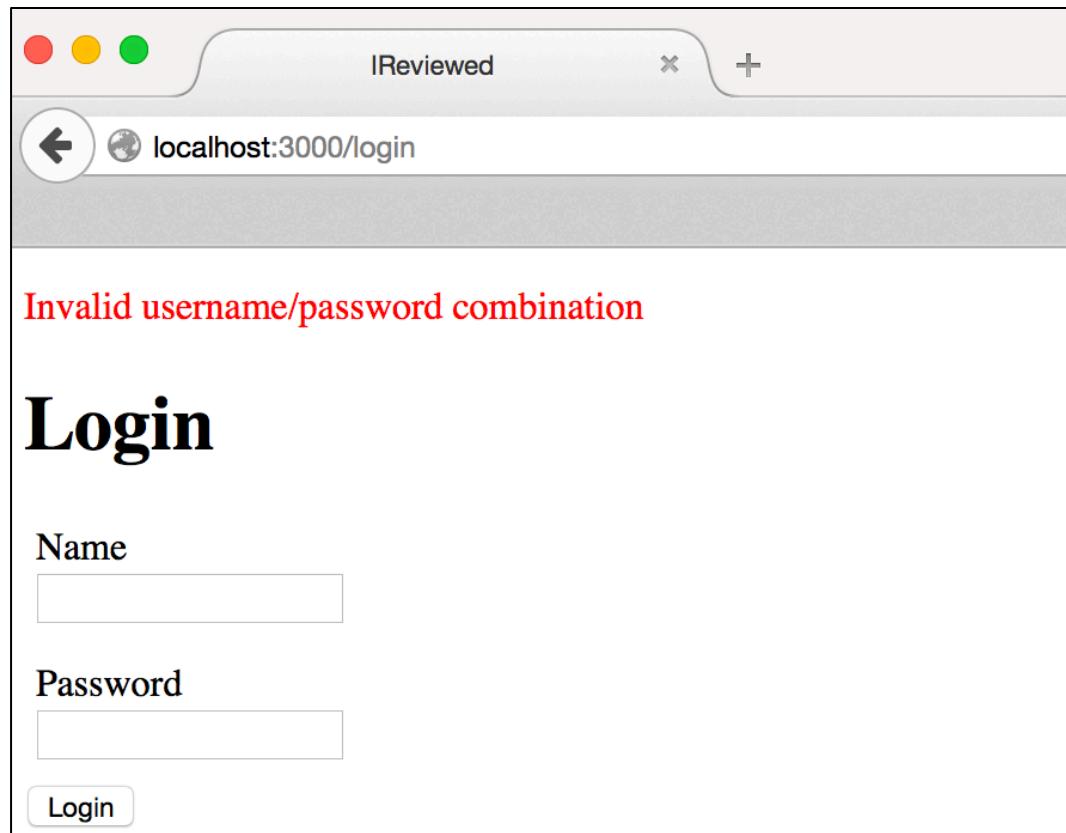
- ▼ i\_reviewed
  - ▼ app
    - assets
    - ▼ controllers
      - concerns
      - application\_controller.rb
      - books\_controller.rb
      - notes\_controller.rb
      - sessions\_controller.rb
  - helpers

◀ ▶ sessions\_controller.rb \*

```
1 class SessionsController < ApplicationController
2   skip_before_action :ensure_login, only: [:new, :create]
3   def new
4     # Login Page - new.html.erb
5   end
6
7   def create ...
8   end
9
10  def destroy ...
11  end
12
13  def index ...
14  end
15
16  def show ...
17
18  end
19
20  def edit ...
21
22  end
23  end
```



# Unsuccessful Login



# Summary

- ✧ Login page corresponds to `new` action `SessionsController`, but uses attributes from `Reviewer`
- ✧ Lock down the app by specifying a `before_action` in `ApplicationController`

## What's Next?

- ✧ Authorization



# In this lecture, we will discuss...

- ✧ Authorization



# Security Helpers

- ✧ Let's add `logged_in?` and `current_user` helpers to `ApplicationController` and make them available as helper methods to all **controllers** and **views** via `helper_method`
- ✧ Then, we can add **logic** to `application.html.erb` for logging out and information about the user who is logged in



# application\_controller.rb

FOLDERS

- ▼ i\_reviewed
  - ▼ app
    - assets
    - ▼ controllers
      - concerns
      - application\_controller.rb
      - books\_controller.rb
      - notes\_controller.rb
      - sessions\_controller.rb
    - helpers
    - mailers
    - models
  - ▼ views
    - books
    - ▼ layouts
      - application.html.erb
    - notes
    - sessions

```
application_controller.rb *
```

```
1 class ApplicationController < ActionController::Base
2   # Prevent CSRF attacks by raising an exception.
3   # For APIs, you may want to use :null_session instead.
4   protect_from_forgery with: :exception
5
6   before_action :ensure_login
7   helper_method :logged_in?, :current_user
8
9   protected
10  def ensure_login
11    # Always go to login page unless session contains
12    # reviewer_id
13    redirect_to login_path unless session[:reviewer_id]
14  end
15
16  def logged_in?
17    session[:reviewer_id] # nil is false
18  end
19
20  def current_user
21    @current_user ||= Reviewer.find(session[:reviewer_id])
22  end
23 end
```



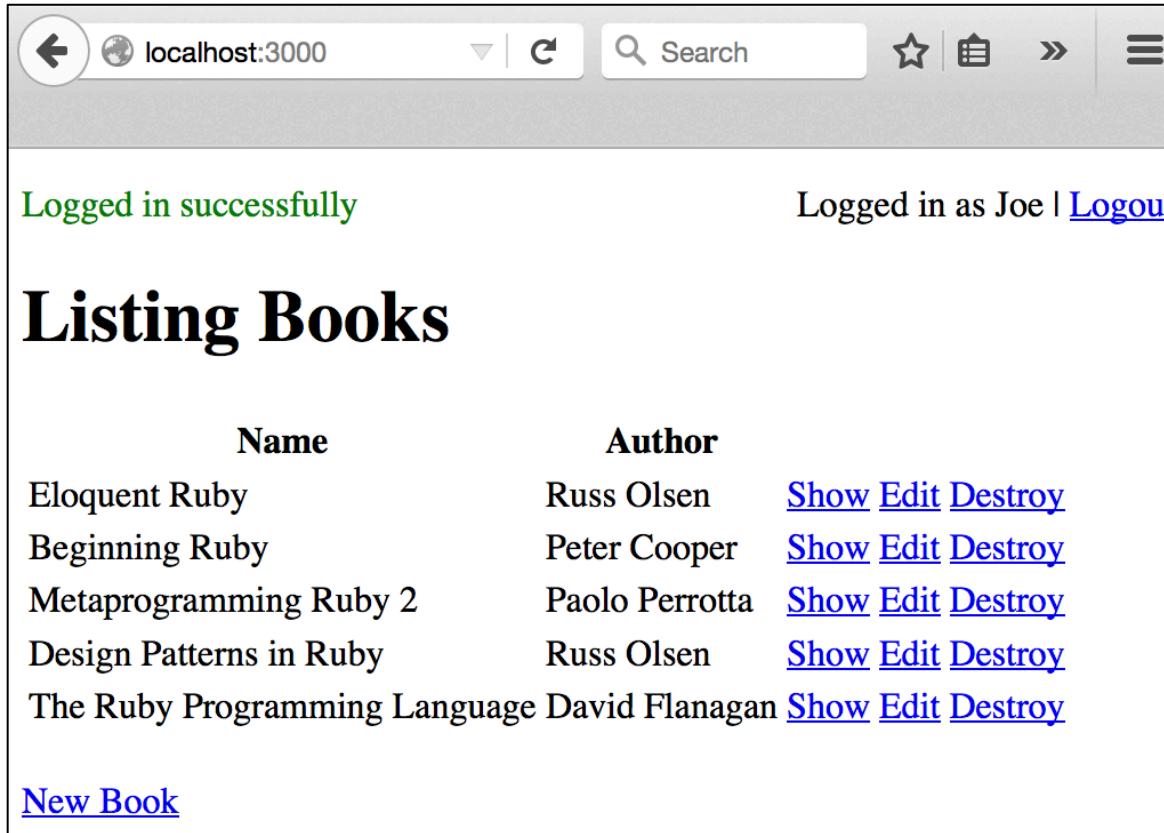
# views/layouts/application.html.erb

The image shows a file explorer on the left and a code editor on the right. The file explorer lists the project structure under 'FOLDERS'. The 'application.html.erb' file is selected in the 'layouts' folder.

```
<!DOCTYPE html>
<html>
<head> ...
</head>
<body>
<% if logged_in? %>
<div style='float: right;'>
  Logged in as <%= current_user.name %> |
  <%= link_to "Logout", logout_path, method: :delete %>
</div>
<% end %>
<% flash.each do |key, value| %>
<p id='<%= key %>'><%= value %></p>
<% end %>
<%= yield %>
</body>
</html>
```



# views/layouts/application.html.erb



The screenshot shows a web browser window with the address bar displaying "localhost:3000". The page content includes a success message "Logged in successfully", a logged-in status message "Logged in as Joe | [Logout](#)", and a title "Listing Books". Below the title is a table listing books with columns for Name and Author, followed by links for Show, Edit, and Destroy. At the bottom is a link to "New Book".

Name	Author	
Eloquent Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Beginning Ruby	Peter Cooper	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Metaprogramming Ruby 2	Paolo Perrotta	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Design Patterns in Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
The Ruby Programming Language	David Flanagan	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[New Book](#)



# Authorization

- ✧ We have implemented basic *Authentication*, but this still does nothing for our *Authorization*
- ✧ Anybody who logs into the system can edit anyone else's books and notes?!
- ✧ **SOLUTION:** We can go back to the `BooksController` and *scope things down* based on the `current_user` (instance of `Reviewer`)



# Authorization – index, new, create

```
books_controller.rb  *

1 class BooksController < ApplicationController
2   before_action :set_book, only: [:show, :edit, :update, :destroy]
3
4   def index
5     @books = current_user.books.all
6   end
7
8   def new
9     @book = current_user.books.new
10  end
11
12
13  def create
14    @book = current_user.books.new(book_params)
15  end
```



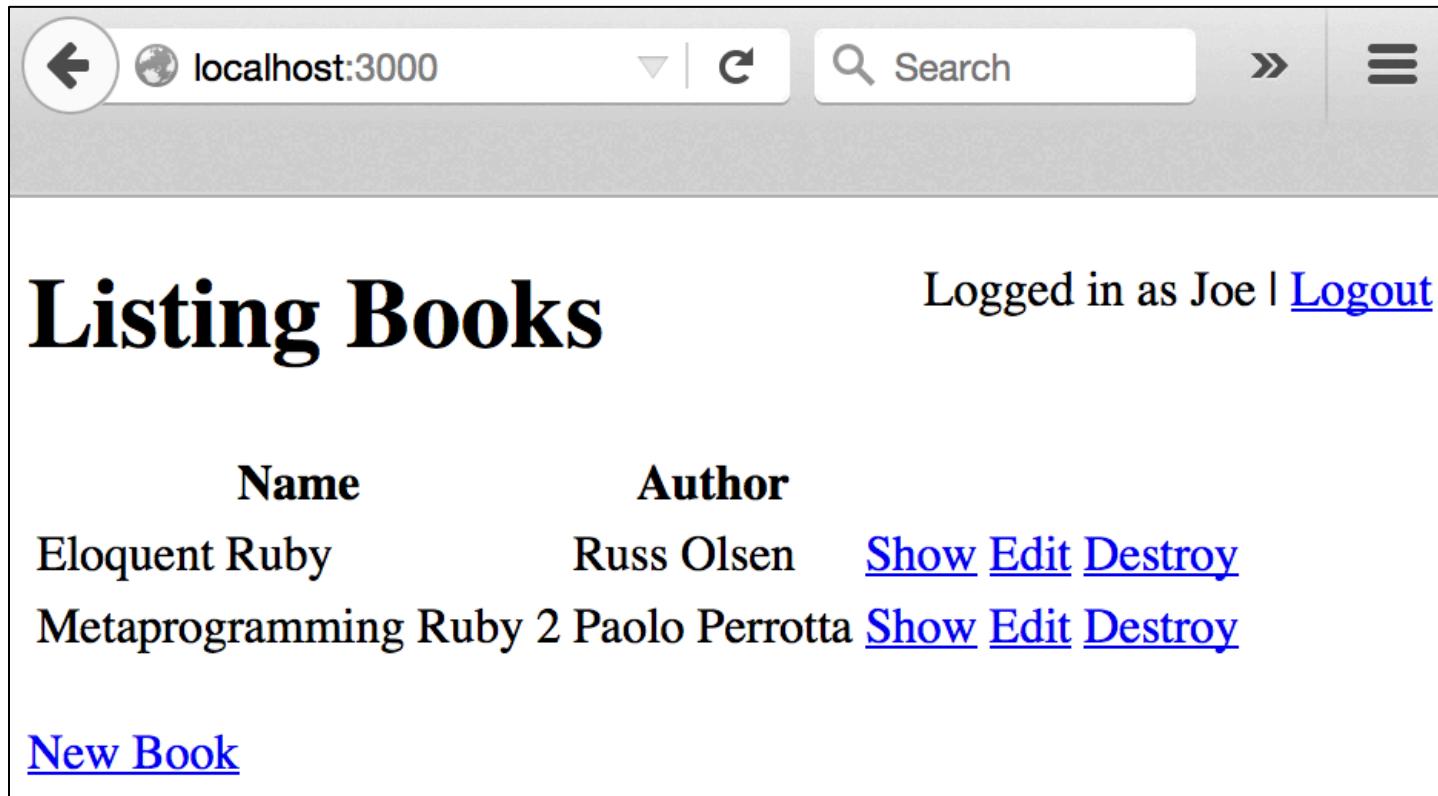
# Authorization – the Other Actions

```
books_controller.rb

1 class BooksController < ApplicationController
2   before_action :set_book, only: [:show, :edit, :update, :destroy]
3
4   def show
5   end
6
7   def edit
8   end
9
10  def update
11  end
12
13  def destroy
14  end
15
16  private
17    # Use callbacks to share common setup or constraints between actions.
18    def set_book
19      @book = current_user.books.find(params[:id])
20    end
21
22
23
24
```



# Authorization



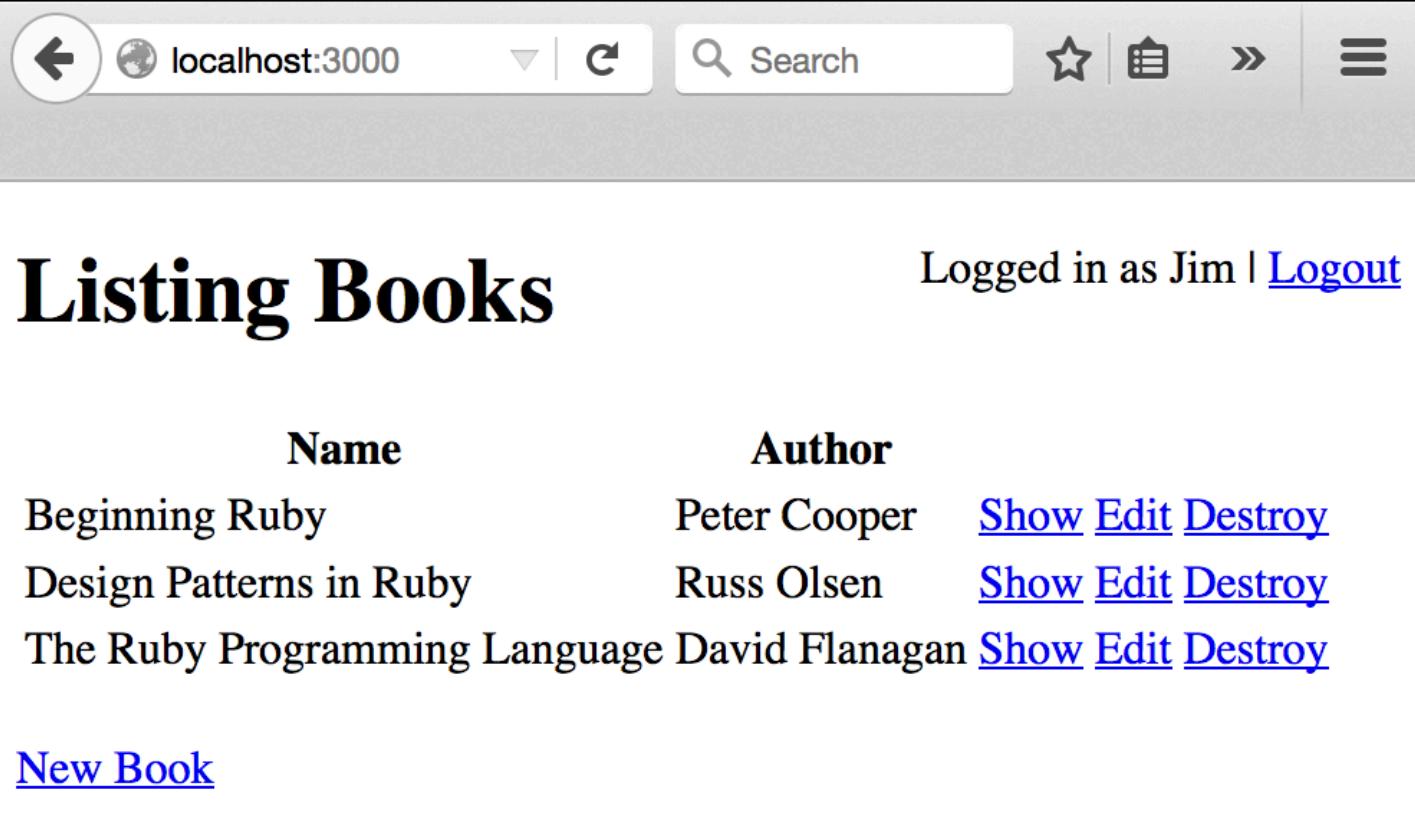
A screenshot of a web browser window titled "Authorization". The address bar shows "localhost:3000". The main content area displays a heading "Listing Books" and a user status message "Logged in as Joe | [Logout](#)". Below this, there is a table-like structure showing two books:

Name	Author	
Eloquent Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Metaprogramming Ruby 2	Paolo Perrotta	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

At the bottom left, there is a link "[New Book](#)". The browser interface includes standard navigation buttons (back, forward, search) and a menu icon.



# Authorization



A screenshot of a web browser window displaying a book listing page. The browser's header includes a back button, a refresh button, a search bar with placeholder text "Search", and a menu icon. The main content area shows a title "Listing Books" and a user status message "Logged in as Jim | [Logout](#)". Below this, there is a table-like structure showing three books, each with columns for Name and Author, and links for Show, Edit, and Destroy. At the bottom left, there is a link to "New Book".

Name	Author	
Beginning Ruby	Peter Cooper	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Design Patterns in Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
The Ruby Programming Language	David Flanagan	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[New Book](#)



# Summary

- ✧ Can use the info stored in the `session` to store current user id
- ✧ Then, can look up resources associated with the current user

## What's Next?

- ✧ Pagination



# In this lecture, we will discuss...

- ✧ Pagination



# Pagination Summary

1. Shut down your server
2. Include `will_paginate` gem
3. Run `$bundle`
4. One line of code in the controller
5. One line of code in your view
6. Restart your server



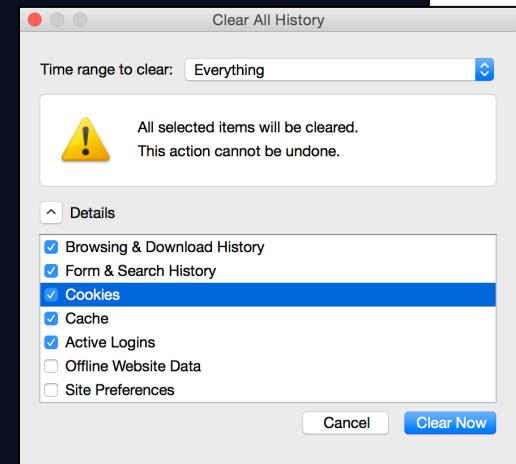
# Seed (Many More) Books

FOLDERS

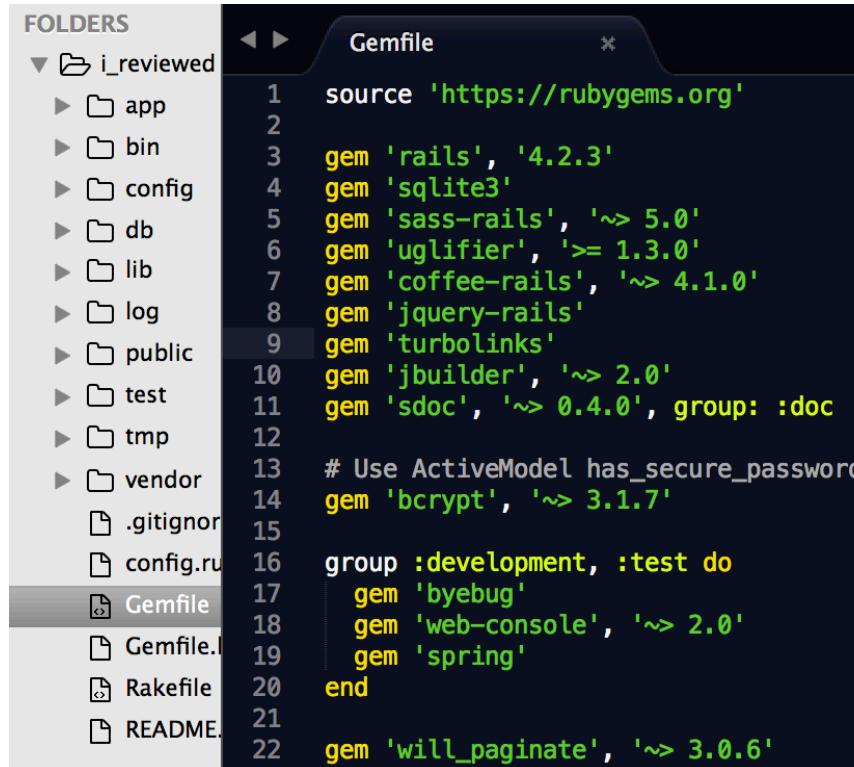
- i\_reviewed
  - app
  - bin
  - config
  - db
    - migrate
    - development
    - schema.rb
  - seeds.rb
- lib
- log
- public
- test
- tmp
- vendor
  - .gitignore
  - config.ru
  - Gemfile

```
seeds.rb
```

```
1 Reviewer.destroy_all
2 Book.destroy_all
3
4 Book.create! [ ... ]
10 ]
11
12 100.times { |index| Book.create! name: "Book#{index}", author: "Author#{index}" }
13
14 eloquent = Book.find_by name: "Eloquent Ruby"
15 eloquent.notes.create!
16   { title: "Wow", note: "Great book to learn Ruby" },
17   { title: "Funny", note: "Doesn't put you to sleep" }
18 ]
19
20 reviewers = Reviewer.create!
21   { name: "Joe", password: "abc123" },
22   { name: "Jim", password: "123abc" }
23 ]
24
25 Book.all.each do |book|
26   bookreviewer = reviewers.sample
27   book.save!
28 end
```



# will\_paginate in the Gemfile



The screenshot shows a code editor with a sidebar titled "FOLDERS" containing a project structure for a Rails application named "i\_reviewed". The "Gemfile" is selected in the sidebar and is displayed in the main editor area. The Gemfile lists various dependencies, including "will\_paginate" at the bottom.

```
source 'https://rubygems.org'  
gem 'rails', '4.2.3'  
gem 'sqlite3'  
gem 'sass-rails', '~> 5.0'  
gem 'uglifier', '>= 1.3.0'  
gem 'coffee-rails', '~> 4.1.0'  
gem 'jquery-rails'  
gem 'turbolinks'  
gem 'jbuilder', '~> 2.0'  
gem 'sdoc', '~> 0.4.0', group: :doc  
# Use ActiveModel has_secure_password  
gem 'bcrypt', '~> 3.1.7'  
group :development, :test do  
  gem 'byebug'  
  gem 'web-console', '~> 2.0'  
  gem 'spring'  
end  
gem 'will_paginate', '~> 3.0.6'
```



# BooksController

```
books_controller.rb  *

class BooksController < ApplicationController
  before_action :set_book, only: [:show, :edit, :update, :destroy]

  def index
    @books = current_user.books.paginate(page: params[:page], per_page: 10)
  end
end
```

```
~/sandbox/i_reviewed$ rails c
Loading development environment (Rails 4.2.3)
irb(main):001:0> Reviewer.first.books.paginate(page: 3, per_page: 10)
Reviewer Load (0.1ms)  SELECT "reviewers".* FROM "reviewers" ORDER BY "reviewers"."id" ASC LIMIT 1
Book Load (0.2ms)  SELECT "books".* FROM "books" WHERE "books"."reviewer_id" = ? LIMIT 10 OFFSET 20 [{"reviewer_id": 6}]
(0.1ms)  SELECT COUNT(*) FROM "books" WHERE "books"."reviewer_id" = ? [{"reviewer_id": 6}]
```



# books/index.html.erb

## FOLDERS

```
▼ └── i_reviewed
    └── app
        ├── assets
        ├── controllers
        ├── helpers
        ├── mailers
        ├── models
        └── views
            └── books
                ├── _form.html.erb
                ├── edit.html.erb
                ├── index.html.erb
                ├── index.json.jbuilder
                ├── new.html.erb
                ├── show.html.erb
                └── show.json.jbuilder
            └── layouts
            └── notes
            └── sessions
        └── bin
```

```
index.html.erb
1 <h1>Listing Books</h1>
2
3 <table>
4     <thead> ...
5     </thead>
6
7     <tbody>
8         <% @books.each do |book| %>
9             <tr>
10                 <td><%= book.name %></td>
11                 <td><%= book.author %></td>
12                 <td><%= link_to 'Show', book %></td>
13                 <td><%= link_to 'Edit', edit_book_path(book) %></td>
14                 <td><%= link_to 'Destroy', book, method: :delete, data: { confirm: 'Are you sure?' } %></td>
15             </tr>
16         <% end %>
17     </tbody>
18 </table>
19 <p>
20
21 <%= will_paginate @books %>
22
23 <br>
24
25
26 <%= link_to 'New Book', new_book_path %>
```



# Pagination Results

## **Listing Books**

Logged in as Joe | [Logout](#)

Name	Author
------	--------

Eloquent Ruby	Russ Olsen
Book1	Author1
Book6	Author6
Book8	Author8
Book12	Author12
Book21	Author21
Book22	Author22
Book23	Author23
Book26	Author26
Book27	Author27

[Show](#) [Edit](#) [Destroy](#)

← Previous [1](#) [2](#) [3](#) [4](#) [5](#) Next →

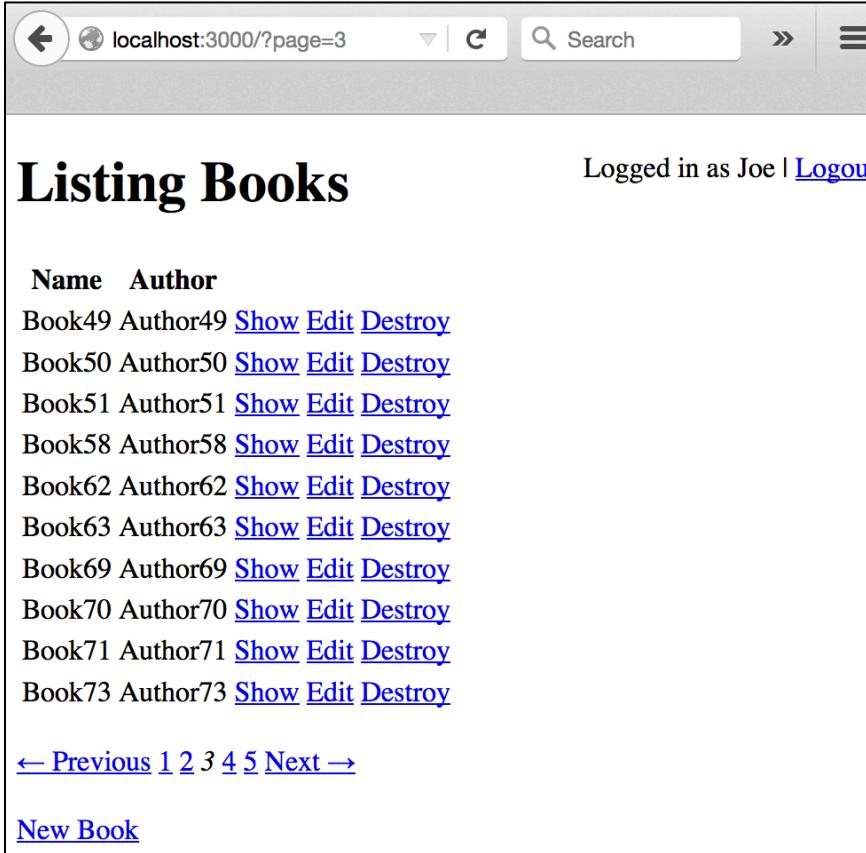
[New Book](#)

localhost:3000/?page=3

**REMEMBER:**  
Restart the server after  
installing new gem



# Pagination Results



A screenshot of a web browser window displaying a list of books. The browser's address bar shows "localhost:3000/?page=3". The main content area has a title "Listing Books" and a user status "Logged in as Joe | [Logout](#)". Below the title is a table header with columns "Name" and "Author". A list of 10 book entries follows, each with three actions: "Show", "Edit", and "Destroy". At the bottom, there are navigation links for "← Previous 1 2 3 4 5 Next →" and a link to "New Book".

Name	Author
Book49	Author49
Book50	Author50
Book51	Author51
Book58	Author58
Book62	Author62
Book63	Author63
Book69	Author69
Book70	Author70
Book71	Author71
Book73	Author73

[← Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Next →](#)

[New Book](#)



# Summary

- ✧ `will_paginate` gem makes it easy to paginate your results
- ✧ Scope your Active Record call from controller and pass in a “which page?” parameter from your view after installing `will_paginate` is all you need

## What's Next?

- ✧ Deploying the app to Heroku



# In this lecture, we will discuss...

- ✧ Deploying to Heroku
- ✧ Enabling SSL



# Heroku Deployment

FOLDERS

- ▼ i\_reviewed
  - ▶ app
    - ▶ assets
    - ▶ controllers
    - ▶ helpers
    - ▶ mailers
    - ▶ models
    - ▶ views
  - ▶ bin
  - ▶ config
  - ▶ db
  - ▶ lib
  - ▶ log
  - ▶ public
  - ▶ test
  - ▶ tmp
  - ▶ vendor
- ▶ .gitignore
- ▶ config.ru
- ▶ Gemfile
- ▶ Gemfile.lock

Gemfile

```
1 source 'https://rubygems.org'  
2  
3 gem 'rails', '4.2.3'  
4 gem 'sass-rails', '~> 5.0'  
5 gem 'uglifier', '>= 1.3.0'  
6 gem 'coffee-rails', '~> 4.1.0'  
7 gem 'jquery-rails'  
8 gem 'turbolinks'  
9 gem 'jbuilder', '~> 2.0'  
10 gem 'sdoc', '~> 0.4.0', group: :doc  
11  
12 # Use ActiveModel has_secure_password  
13 gem 'bcrypt', '~> 3.1.7'  
14  
15 group :development, :test do  
16   gem 'byebug'  
17   gem 'web-console', '~> 2.0'  
18   gem 'spring'  
19   gem 'sqlite3'  
20 end  
21  
22 group :production do  
23   gem 'pg'  
24   gem 'rails_12factor'  
25 end  
26  
27 gem 'will_paginate', '~> 3.0.6'
```

```
~/sandbox/i_reviewed$ bundle --without production
```

config

```
1 ---  
2 BUNDLE_WITHOUT: production  
3 config
```



# Heroku Deployment

```
~/sandbox/i_reviewed$ git st
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
~/sandbox/i_reviewed$ ls
.          .git      Gemfile.lock app      config.ru   log      tmp
..         .gitignore README.rdoc bin      db        public    vendor
.bundle    Gemfile   Rakefile   config    lib       test
~/sandbox/i_reviewed$ heroku login
Setting up node-v4.2.1... done
Installing core plugin heroku-cli-addons... done
Enter your Heroku credentials.
Email: kalmanh@gmail.com
Password (typing will be hidden):
Logged in as kalmanh@gmail.com
~/sandbox/i_reviewed$ heroku create ireview-books
Creating ireview-books... done, stack is cedar-14
https://ireview-books.herokuapp.com/ | https://git.heroku.com/ireview-books.git
Git remote heroku added
~/sandbox/i_reviewed$ █
```



# Heroku Deployment

```
~/sandbox/i_reviewed$ git push heroku master
Counting objects: 177, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (159/159), done.
Writing objects: 100% (177/177), 29.72 KiB | 0 bytes/s, done.
Total 177 (delta 51), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Ruby app detected
remote: -----> Compiling Ruby/Rails
remote: -----> Using Ruby version: ruby-2.0.0
remote: -----> Installing dependencies using bundler 1.9.7
remote:           Running: bundle install --without development:test
```



# Heroku Deployment

```
remote: -----> Discovering process types
remote:           Procfile declares types -> (none)
remote:           Default types for Ruby -> console, rake, web, worker
remote:
remote: -----> Compressing... done, 29.4MB
remote: -----> Launching... done, v5
remote:           https://ireview-books.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy.... done.
To https://git.heroku.com/ireview-books.git
 * [new branch]      master -> master
```



# Heroku Deployment

The image displays two side-by-side browser windows. The left window shows a successful login attempt, while the right window shows an error message.

**Left Browser Window:**

- Address bar: ireview-books.herokuapp.com
- Content: A "Login" form with fields for Name (containing "Joe") and Password (containing "\*\*\*\*\*"). A "Login" button is present.

**Right Browser Window:**

- Address bar: ireview-books.herokuappapp.com/sessions
- Content: An error message box with the text "We're sorry, but something went wrong." and a sub-message: "If you are the application owner check the logs for more information."



# Wait For Me...

- ❖ We deployed the app, but what about the database?
- ❖ \$heroku run rake db:migrate
- ❖ \$heroku run rake db:seed



# Migrations And Seeding On Heroku

```
~/sandbox/i_reviewed$ heroku run rake db:migrate
Running rake db:migrate on ireview-books... up, run.9261
  (15.3ms) CREATE TABLE "schema_migrations" ("version" character varying NOT NULL)
  (10.1ms) CREATE UNIQUE INDEX "unique_schema_migrations" ON "schema_migrations" ("version")
ActiveRecord::SchemaMigration Load (2.5ms) SELECT "schema_migrations".* FROM "schema_migrations"
Migrating to CreateReviewers (20151015232312)
  (1.4ms) BEGIN
== 20151015232312 CreateReviewers: migrating =====
-- create_table(:reviewers)
  (23.2ms) CREATE TABLE "reviewers" ("id" serial primary key, "name" character varying, "password_digest" character varying,
"created_at" timestamp NOT NULL, "updated_at" timestamp NOT NULL)
    -> 0.0288s
== 20151015232312 CreateReviewers: migrated (0.0311s) =====

SQL (1.4ms) INSERT INTO "schema_migrations" ("version") VALUES ($1) [[{"version": "20151015232312"}]]
  (2.5ms) COMMIT
Migrating to CreateBooks (20151015232349)
  (1.3ms) BEGIN
== 20151015232349 CreateBooks: migrating =====
-- create_table(:books)
  (18.9ms) CREATE TABLE "books" ("id" serial primary key, "name" character varying, "author" character varying, "reviewer_id"
" integer, "created_at" timestamp NOT NULL, "updated_at" timestamp NOT NULL)
  (11.1ms) CREATE INDEX "index_books_on_reviewer_id" ON "books" ("reviewer_id")
  (2.8ms) ALTER TABLE "books" ADD CONSTRAINT "fk_rails_edc0310d15"
FOREIGN KEY ("reviewer_id")
```



# Migrations And Seeding On Heroku

```
~/sandbox/i_reviewed$ heroku run rake db:seed
Running rake db:seed on ireview-books... up, run.2793
  ActiveRecord::SchemaMigration Load (1.0ms)  SELECT "schema_migrations".* FROM "schema_migrations"
  Reviewer Load (4.9ms)  SELECT "reviewers".* FROM "reviewers"
  Book Load (13.0ms)  SELECT "books".* FROM "books"
  (14.6ms)  BEGIN
  SQL (82.7ms)  INSERT INTO "books" ("name", "author", "created_at", "updated_at") VALUES ($1, $2, $3, $4) RETURNING "id" [[{"name": "Eloquent Ruby"}, {"author": "Russ Olsen"}, {"created_at": "2015-10-22 16:11:36.972204"}, {"updated_at": "2015-10-22 16:11:36.972204"}]]
  (13.9ms)  COMMIT
  (13.2ms)  BEGIN
  SQL (13.8ms)  INSERT INTO "books" ("name", "author", "created_at", "updated_at") VALUES ($1, $2, $3, $4) RETURNING "id" [[{"name": "Beginning Ruby"}, {"author": "Peter Cooper"}, {"created_at": "2015-10-22 16:11:37.196688"}, {"updated_at": "2015-10-22 16:11:37.196688"}]]
  (48.7ms)  COMMIT
  (9.0ms)  BEGIN
  SQL (29.9ms)  INSERT INTO "books" ("name", "author", "created_at", "updated_at") VALUES ($1, $2, $3, $4) RETURNING "id" [[{"name": "Metaprogramming Ruby 2"}, {"author": "Paolo Perrotta"}, {"created_at": "2015-10-22 16:11:37.325074"}, {"updated_at": "2015-10-22 16:11:37.325074"}]]
  (19.3ms)  COMMIT
  (2.8ms)  BEGIN
  SQL (49.8ms)  INSERT INTO "books" ("name", "author", "created_at", "updated_at") VALUES ($1, $2, $3, $4) RETURNING "id" [[{"name": "Design Patterns in Ruby"}, {"author": "Russ Olsen"}, {"created_at": "2015-10-22 16:11:37.462418"}, {"updated_at": "2015-10-22 16:11:37.462418"}]]
  (11.1ms)  COMMIT
```



# Heroku Logs

```
2015-10-22T16:07:51.833892+00:00 app[web.1]: 
2015-10-22T16:07:51.801124+00:00 app[web.1]: Processing by SessionsController#create as HTML
2015-10-22T16:07:51.832898+00:00 app[web.1]: PG::UndefinedTable: ERROR: relation "reviewers" does not exist
2015-10-22T16:07:51.832900+00:00 app[web.1]: LINE 5:           WHERE a.attrelid = '"reviewers"'::regclass
2015-10-22T16:07:51.832901+00:00 app[web.1]:                                ^
2015-10-22T16:07:51.832902+00:00 app[web.1]: :                     SELECT a.attname, format_type(a.atttypid, a.atttypmod),
2015-10-22T16:07:51.832903+00:00 app[web.1]:                               pg_get_expr(d.adbin, d.adrelid), a.attnotnull, a.atttypid, a.atttypmod
2015-10-22T16:07:51.832904+00:00 app[web.1]: FROM pg_attribute a LEFT JOIN pg_attrdef d
2015-10-22T16:07:51.832904+00:00 app[web.1]:   ON a.attrelid = d.adrelid AND a.attnum = d.adnum
2015-10-22T16:07:51.832905+00:00 app[web.1]: WHERE a.attrelid = '"reviewers"'::regclass
2015-10-22T16:07:51.832906+00:00 app[web.1]:   AND a.attnum > 0 AND NOT a.attisdropped
2015-10-22T16:07:51.832906+00:00 app[web.1]: ORDER BY a.attnum
2015-10-22T16:07:51.832907+00:00 app[web.1]: 
2015-10-22T16:07:51.833120+00:00 app[web.1]: Completed 500 Internal Server Error in 3ms (ActiveRecord: 1.2ms)
2015-10-22T16:07:51.928494+00:00 heroku[router]: at=info method=POST path="/sessions" host=ireview-books.herokuapp.com request_id=d19918fc-1c0c-43a7-9a6b-02669d4344c5 fwd="192.12.13.14" dyno=web.1 connect=2ms service=99ms status=500 bytes=1754
2015-10-22T16:09:39.443387+00:00 heroku[api]: Starting process with command `bundle exec rake db:migrate` by kalmanh@gmail.com
2015-10-22T16:09:42.454159+00:00 heroku[run.9261]: Awaiting client
2015-10-22T16:09:42.486496+00:00 heroku[run.9261]: Starting process with command `bundle exec rake db:migrate`
2015-10-22T16:09:42.854498+00:00 heroku[run.9261]: State changed from starting to up
2015-10-22T16:09:47.880649+00:00 heroku[run.9261]: Process exited with status 0
2015-10-22T16:09:47.904400+00:00 heroku[run.9261]: State changed from up to complete
2015-10-22T16:11:26.998612+00:00 heroku[api]: Starting process with command `bundle exec rake db:seed` by kalmanh@gmail.com
2015-10-22T16:11:29.859107+00:00 heroku[run.2793]: Awaiting client
2015-10-22T16:11:29.897069+00:00 heroku[run.2793]: Starting process with command `bundle exec rake db:seed`
```



# Happy Ending

The screenshot shows a web browser window with the URL `ireview-books.herokuapp.com`. The page title is "Happy Ending". The top navigation bar includes a back button, a globe icon, a search bar with placeholder "Search", and various icons for bookmarking, download, and user profile.

On the left, a green message says "Logged in successfully". On the right, it says "Logged in as Joe | [Logout](#)".

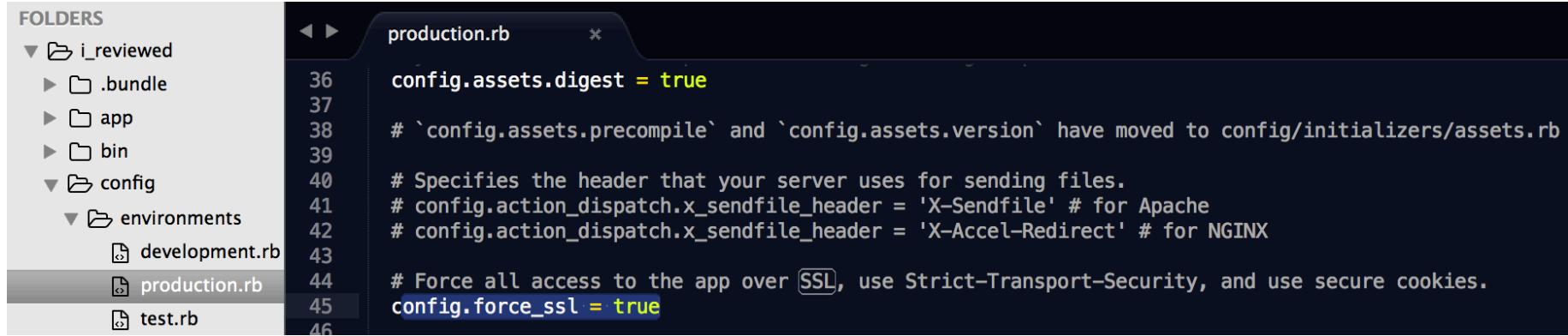
## Listing Books

Name	Author	
Eloquent Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Beginning Ruby	Peter Cooper	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Metaprogramming Ruby 2	Paolo Perrotta	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
The Ruby Programming Language	David Flanagan	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book3	Author3	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book4	Author4	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book10	Author10	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book12	Author12	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book13	Author13	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book14	Author14	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[← Previous](#) [1](#) [2](#) [3](#) [4](#) [Next →](#)



# Enabling SSL



The screenshot shows a code editor with a sidebar titled "FOLDERS". The "production.rb" file is open in the main pane. The code contains several configuration settings, including one that forces all access to the app over SSL:

```
config.assets.digest = true
# `config.assets.compile` and `config.assets.version` have moved to config/initializers/assets.rb
# Specifies the header that your server uses for sending files.
# config.action_dispatch.x_sendfile_header = 'X-Sendfile' # for Apache
# config.action_dispatch.x_sendfile_header = 'X-Accel-Redirect' # for NGINX
# Force all access to the app over SSL, use Strict-Transport-Security, and use secure cookies.
config.force_ssl = true
```

Commit the change to git and  
push to Heroku



# Happy Ending with https!

The screenshot shows a web browser window with the following details:

- Address Bar:** https://ireview-books.herokuapp.com
- Header:** Logged in successfully (green text) and Logged in as Joe | [Logout](#)
- Title:** Listing Books
- Table:** A list of books with their authors and actions.
- Pagination:** ← Previous 1 [2](#) [3](#) [4](#) Next →

Name	Author	
Eloquent Ruby	Russ Olsen	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Beginning Ruby	Peter Cooper	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Metaprogramming Ruby 2	Paolo Perrotta	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
The Ruby Programming Language	David Flanagan	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book3	Author3	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book4	Author4	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book10	Author10	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book12	Author12	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book13	Author13	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book14	Author14	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>



# Summary

- ✧ Use “`bundle --without production`” if you don’t need or can’t install production gems on your local system
- ✧ Don’t forget to run DB commands on the remote server after deployment
- ✧ Enable https with `config.force_ssl = true`

