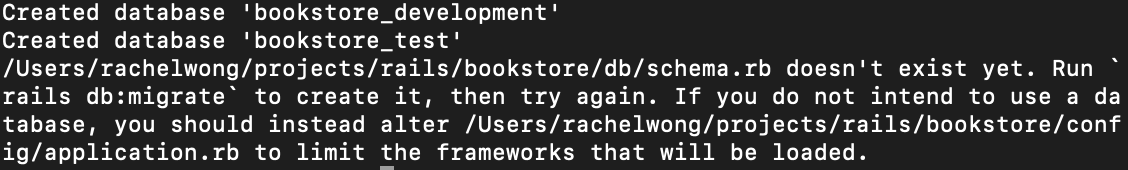
**Matt Stubbs Bookstore Rails App Tutorial**

1. Create the app 🡪 rails new bookstore -database=postgresql
2. Start the rails server 🡪 cd bookstore 🡪 rails s
3. You will get below error in localhost:3000 on the browser.

It is telling you to create a database for the app to work.



1. Create database 🡪 Rails db:setup



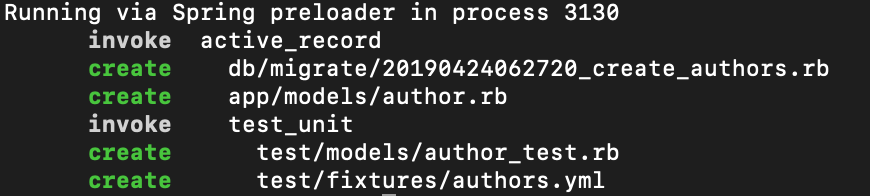
1. Checking to see whether a database is created 🡪 Rails db:create



1. Launch on browser 🡪 Refresh localhost:3000 (rails s if needed)



1. Create author model 🡪 rails g model author



1. Add name and url\_image as attributes to the author model in the db > migrate folder > [timestamp]\_create\_authors.rb

class CreateAuthors < ActiveRecord::Migration[5.2]

def change

create\_table :authors do |t|

t.string :name

t.string :url\_image

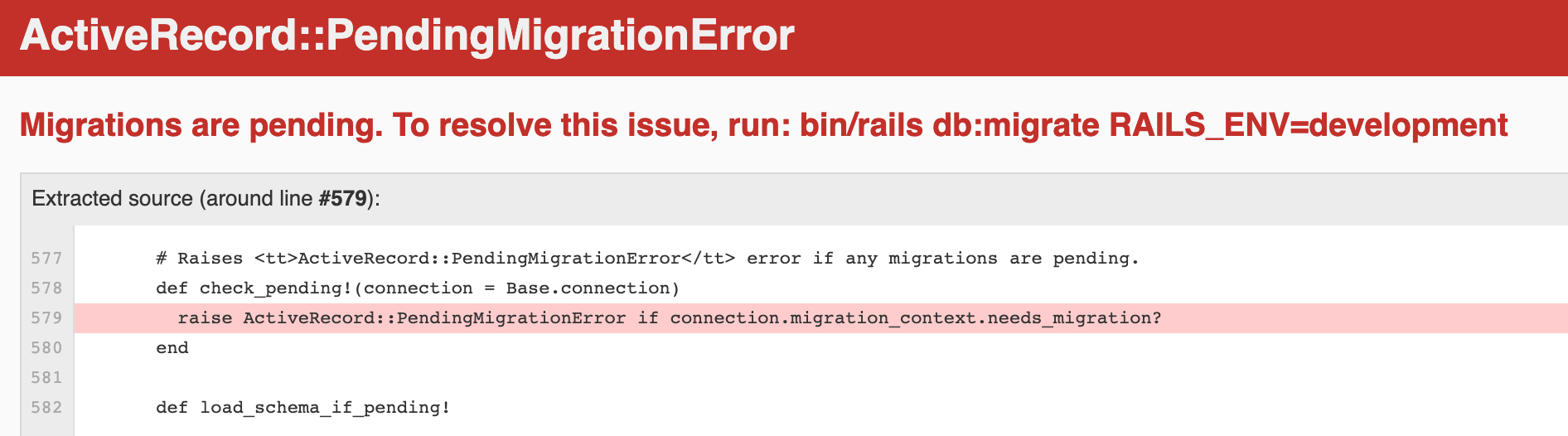
t.timestamps

end

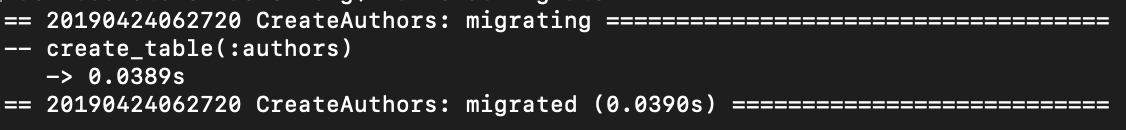
end

end

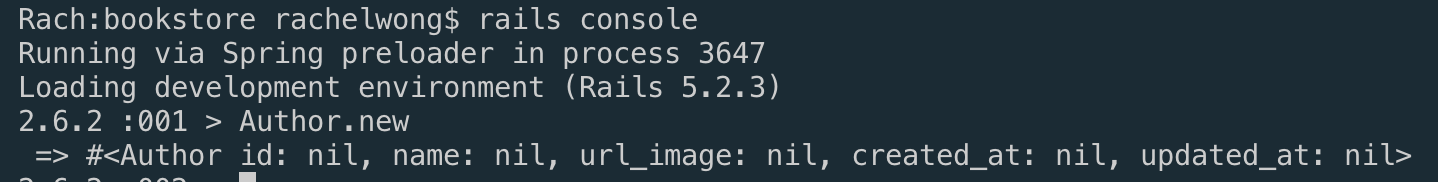
1. You get the below error in localhost: 3000. It is telling you to apply the migration.



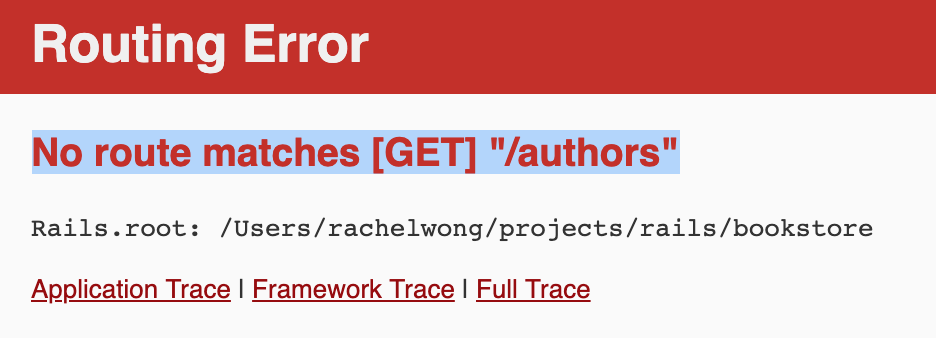
1. Apply migration 🡪 rails db:migrate



1. Confirm you have created an author model with the right attributes 🡪 rails c 🡪 Author.new

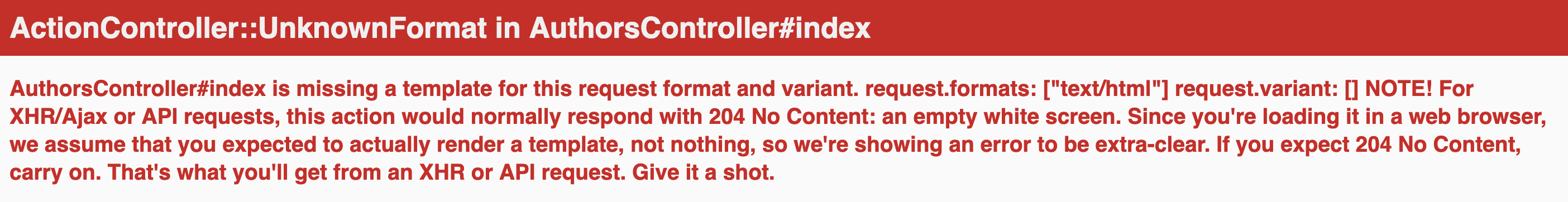


1. Create Controller 🡪 Rails g controller authors
2. We list all the authors by convention so we call it index method 🡪 Create empty index method in apps > controller > authors\_controller.rb
3. On localhost:3000/authors, you get below error. It is telling you that you are missing a routes configuration.

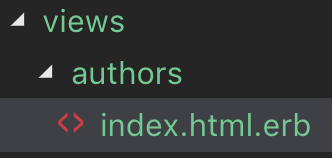


* Check localhost:3000/rails/info/routes for all available routes, methods and actions (when not deployed)

1. Add a route for browser to get to index method. 🡪 go to config > routes.rb 🡪 add resources :authors
2. Rails.application.routes.draw do
3. # For details on the DSL available within this file, see http://guides.rubyonrails.org/routing.html
4. resources :authors
5. end
6. Refresh localhost:3000/authors and you get below error. It is telling you that you are missing a view template file.



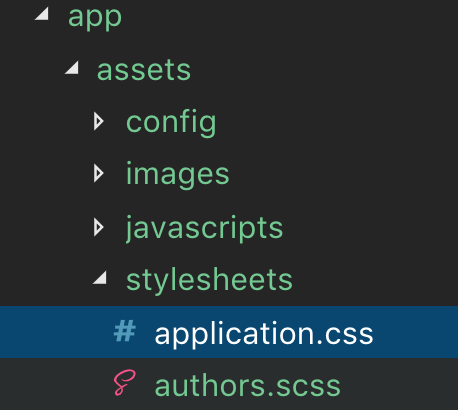
1. Create view file for authors 🡪 views > authors > create index.html.erb



* Add some identifiable text in this file so that when you refresh localhost:3000/index you can see it

1. Add Matt S’s css to assets > stylesheets > application.css

* Refresh your localhost:3000/authors



1. Controller need to request the data and render it in the view.

Demo example

Add new instance variable in authors\_controller.rb 🡪 def index @authors = “hi there”

Add in authors’ view index.html.erb anywhere 🡪 <%= @authors %>

To fetch the authors data from the model, ask the model for everything.

In controllers > Authors\_controller.rb

class AuthorsController < ApplicationController

def index

@authors = Author.all

end

end

In view > authors > index.html.erb

<h1>List of Authors (Authors Index View)</h1>

<!-- display a full list of author's name from the authors database via the Author model -->

<%= @authors.each do |author| %>

<%= author.name %>

<% end %>

1. Add Matt’s database seed data to db > seed.rb

* Run rails db:seed

1. Add html to render the page a bit better in views > authors > index.html.erb
2. <h1>List of Authors (Authors Index View)</h1>
3. <div *class*="authors">
4. <!-- display a full list of author's name from the authors database via the Author model -->
5. <%= @authors.each do |author| %>
6. <div *class*="author">
7. <h3><%= author.name %></h3>
8. </div>
9. <% end %>
10. </div>

On localhost:3000, the names of the author will appear as a column in bold.



27. Add images next to each of the authors

Rails has a native method image\_tag that displays images pulled from the database.

It does the same thing as <img src="<%= author.url\_image %>

<h1>List of Authors (Authors Index View)</h1>

<div *class*="authors">

<!-- display a full list of author's name from the authors database via the Author model -->

<%= @authors.each do |author| %>

<div *class*="author">

<!--Rails native helper method image\_tag replaces this -->

<!--<img src="<%= author.url\_image %>"-->

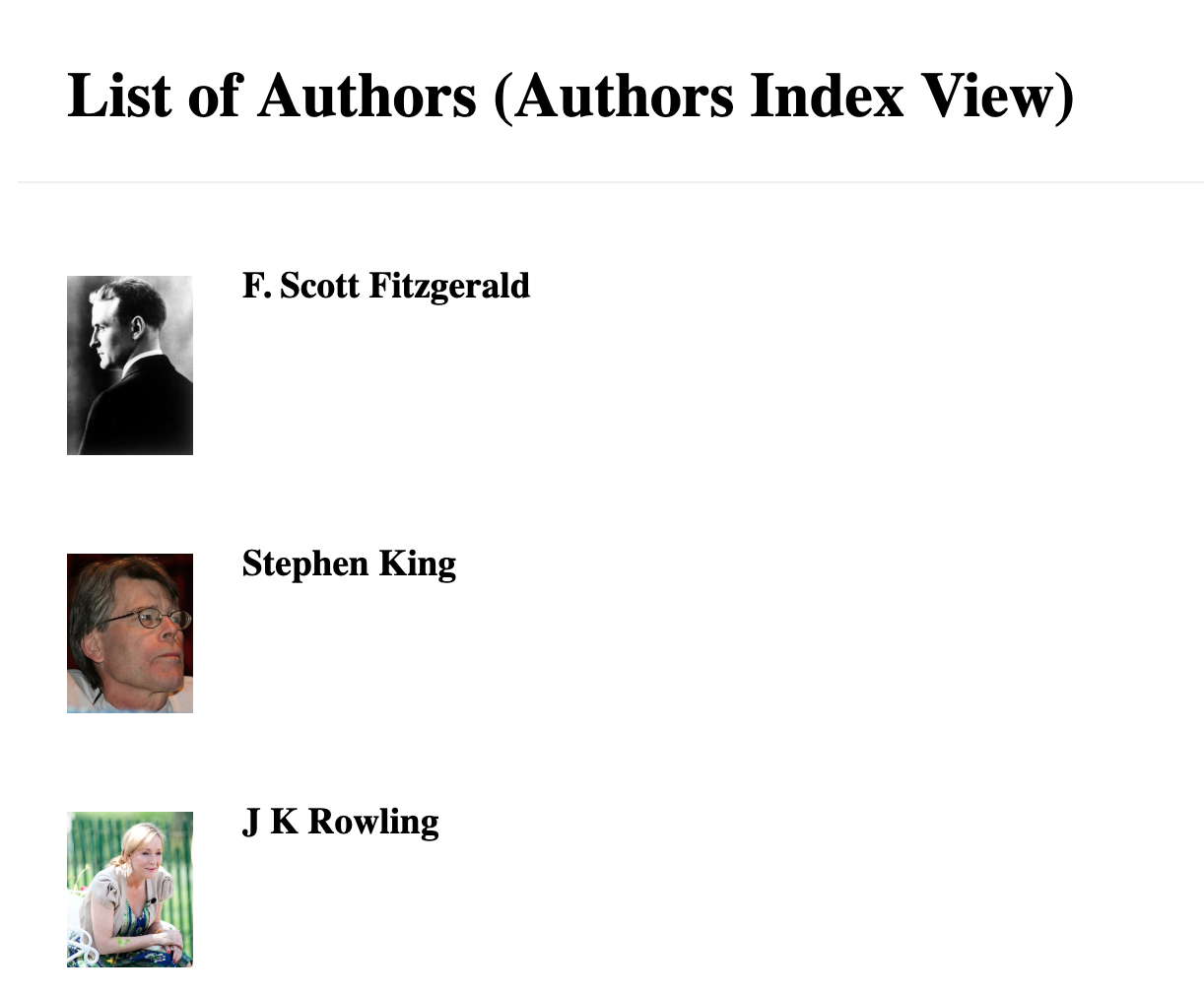
<%= image\_tag author.url\_image, width: 64 %>

<h3><%= author.name %></h3>

</div>

<% end %>

</div>



1. Add an extra entry into the Authors table for testing 🡪

Rails c

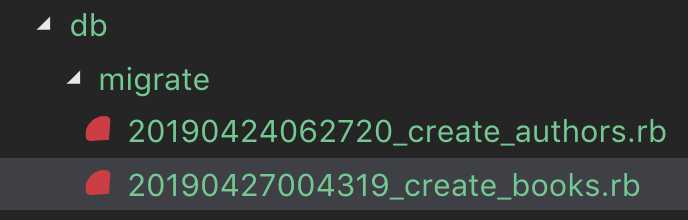
Author.create(name: “Jane Austen”, url\_image: <https://upload.wikimedia.org/wikipedia/commons/thumb/c/cc/CassandraAusten-JaneAusten%28c.1810%29_hires.jpg/400px-CassandraAusten-JaneAusten%28c.1810%29_hires.jpg>)

1. Create a book model and run the seed file again. (No point instantiate books data when there’s no model)

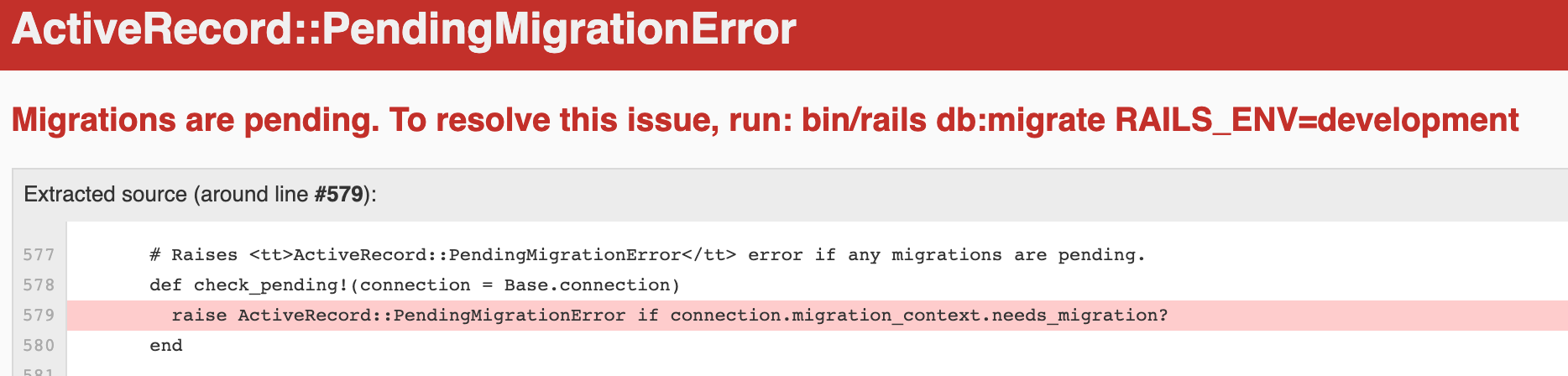
The book belongs to an author so it needs to refer(ence) to the author model.

🡪 Rails g model book title:string url\_image:string author:references

1. A new migrate file is created for the book model in db folder



1. Refreshing localhost:3000 you get below error. It is telling you to migrate the database with the creation of the new model



* Rails db:migrate

1. ON models > book.rb, the book model belongs\_to :author which will provide the book.author method
2. On models > author.rb, the relationship between authors and book is that an author has many books.
3. class Author < ApplicationRecord
4. has\_many :books
5. end
6. Draw out the number of books an author has by adding to views > authors > index.html.erb

<div *class*="authors">

<!-- display a full list of author's name from the authors database via the Author model -->

<%= @authors.each do |author| %>

<div *class*="author">

<!--Rails native helper method image\_tag replaces this -->

<!--<img src="<%= author.url\_image %>"-->

<%= image\_tag author.url\_image, width: 64 %>

<h3><%= author.name %></h3>

<p>

<%= author.books.count %> books

</p>

</div>

<% end %>

</div>



1. Modify the line
2. <%= author.books.count %> books

to

1. <%= pluralize author.books.count, "book" %>

Pluralize is a rails method that will know when the .count returns a number greater than 1 to use the plural form of the word “book”.

1. Make localhost:3000/authors your default home page (appear as localhost:300) by modifying config > routes.rb

Rails.application.routes.draw do

resources :authors

root to: "authors#index"

end