

Getting Started from Scratch

PRO TIP

Throughout the tutorial, we're going to run a lot of Terminal commands. For example, we might ask you to do something like

Run `pwd` to print your current working directory.

To run the command, you'll need to type it in Terminal and hit 'enter' to execute it.

- 1 Open Terminal.
- 2 You're now in your home directory. Run `ls` to see what's in your home directory. You might see some familiar directories like `Desktop`, `Documents`, and `Music`.
- 3 All code deserves a good home. Add a new `Projects` directory to your home directory by running `mkdir Projects`.
- 4 Navigate to your new `Projects` directory by running `cd Projects`.
- 5 Now you're ready to create the bookstore!

SHELL OUTPUT

```
> ls
Desktop Document Music

> mkdir Projects

> cd Projects
```

Before you can complete the following steps, your computer needs to be configured with Rails. If you haven't already, please refer to the [install instructions](#) for steps on configuring your computer.

- 1 In your `Projects` directory, run `rails new bookstore`.
- 2 `rails new` does a lot of stuff! First, it creates the files and directories that make up the basic structure of your bookstore application. Then, it runs `bundle install` to install the dependencies needed to run the application.
- 3 After `rails new` finishes, run `cd bookstore` to navigate to the newly created bookstore application.

SHELL OUTPUT

```
> rails new bookstore
create
create  README.md
create  Rakefile
create  config.ru
create  .gitignore
create  Gemfile
create  app
...
run bundle install
Fetching gem metadata from http://rubygems.org/.....
Fetching version metadata from http://rubygems.org/...
Fetching dependency metadata from http://rubygems.org/..
Resolving dependencies.....
Installing rake 11.3.0
...
```

Let's take a look the files that got created.

- 1 Run `ls -l` to list out all the top level files and directories `rails new` generated. There is lot going on, but a few things stand out.
- 2 Any code that is specific to the bookstore application will live in the `app` directory. Run `ls -l app` to take a look at the some of the things you'll be working with throughout the tutorial.

SHELL OUTPUT

```
> ls -l
total 48
-rw-r--r-- 1 awesomesauce staff 1728 Nov  6 14:31 Gemfile
-rw-r--r-- 1 awesomesauce staff 4329 Nov  6 14:33 Gemfile.lock
-rw-r--r-- 1 awesomesauce staff  374 Nov  6 14:13 README.md
-rw-r--r-- 1 awesomesauce staff  227 Nov  6 14:13 Rakefile
drwxr-xr-x 10 awesomesauce staff   340 Nov  6 14:13 app
drwxr-xr-x  7 awesomesauce staff   238 Nov  6 14:13 bin
drwxr-xr-x 14 awesomesauce staff   476 Nov 12 18:56 config
-rw-r--r--  1 awesomesauce staff  130 Nov  6 14:13 config.ru
drwxr-xr-x  4 awesomesauce staff   136 Nov 12 18:54 db
drwxr-xr-x  4 awesomesauce staff   136 Nov  6 14:13 lib
drwxr-xr-x  4 awesomesauce staff   136 Nov  6 14:34 log
drwxr-xr-x  9 awesomesauce staff   306 Nov  6 14:13 public
drwxr-xr-x  9 awesomesauce staff   306 Nov  6 14:13 test
drwxr-xr-x  7 awesomesauce staff   238 Nov  6 14:34 tmp
drwxr-xr-x  3 awesomesauce staff   102 Nov  6 14:13 vendor

> ls -l app
total 0
drwxr-xr-x  6 awesomesauce staff  204 Nov  6 14:13 assets
drwxr-xr-x  3 awesomesauce staff  102 Nov  6 14:13 channels
drwxr-xr-x  4 awesomesauce staff  136 Nov  6 14:13 controllers
drwxr-xr-x  3 awesomesauce staff  102 Nov  6 14:13 helpers
drwxr-xr-x  3 awesomesauce staff  102 Nov  6 14:13 jobs
drwxr-xr-x  3 awesomesauce staff  102 Nov  6 14:13 mailers
drwxr-xr-x  4 awesomesauce staff  136 Nov  6 14:13 models
drwxr-xr-x  3 awesomesauce staff  102 Nov  6 14:13 views
```

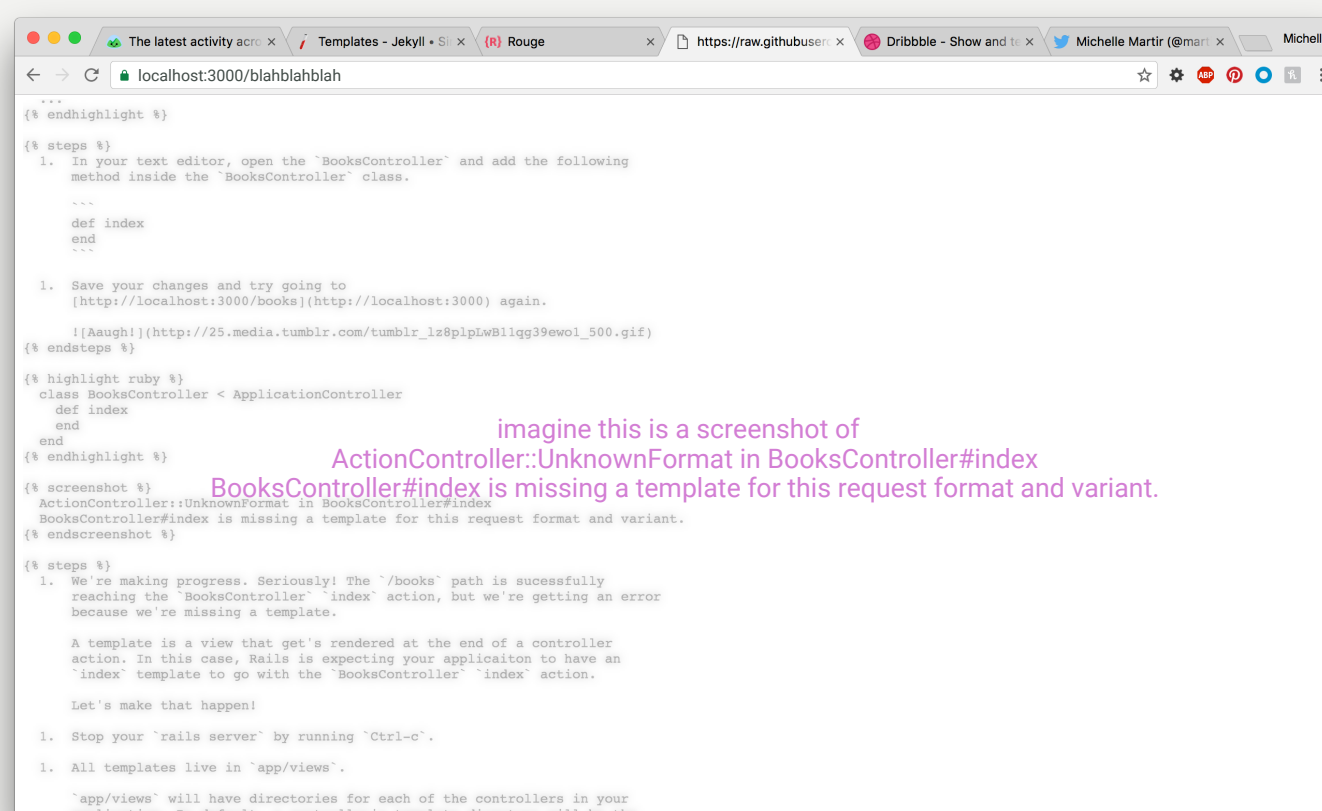
- 1 In your text editor, open the `BooksController` and add the following method inside the `BooksController` class.

```
def index
end
```

- 2 Save your changes and try going to <http://localhost:3000/books> again.

RUBY OUTPUT

```
1 class BooksController < ApplicationController
2   def index
3   end
4 end
```



We're making progress. Seriously! The `/books` path is successfully reaching the `BooksController` `index` action, but we're getting an error because we're missing a template.

A template is a view that gets rendered at the end of a controller action. In this case, Rails is expecting your application to have an `index` template to go with the `BooksController` `index` action.

Let's make that happen!

- 1 Stop your `rails server` by hitting `Ctrl + C`.

All templates live in `app/views`.

`app/views` will have directories for each of the controllers in your application. By default, a controller's template directory will be the name of the controller...minus the controller.

For example, the `BooksController`'s template directory will be `app/views/books`.

- 2 Create the `BooksController` template directory by running `mkdir app/views/books`.

- 3 Now, create the `BooksController`'s `index` template by running `touch app/views/books/index.html.erb`.

- 4 Restart your web server. (Remember? Start your web server by running `rails server`.)

- 5 Now, try going to <http://localhost:3000/books>. Yay, no errors!

SHELL OUTPUT

```
> rails server
=> Booting Puma
...
^CExiting

> mkdir app/views/books

> touch app/views/books/index.html.erb

> rails server
...
Started GET "/books" for :::1 at 2016-11-29 22:15:07 -0500
ActiveRecord::SchemaMigration Load (0.2ms) SELECT
"schema_migrations".* FROM "schema_migrations"
Processing by BooksController#index as HTML
Rendering books/index.html.erb within layouts/application
Rendered books/index.html.erb within layouts/application (1.7ms)
Completed 200 OK in 645ms (Views: 631.6ms | ActiveRecord: 0.0ms)
```

Ok, let's make this page a little more interesting.

HTML OUTPUT

- 1 In your text editor, open `app/views/books/index.html.erb` and add the following line:

```
<h1>Welcome to My Super Rad Bookstore!</h1>
```

- 2 Save your changes and go to <http://localhost:3000/books> again. Yay! 🎉

```
1 <h1>Welcome to My Super Rad Bookstore!</h1>
```