**CREATE A NEW RAILS PROJECT**

3.1 Installing rails

1. Verify you have a current version of Ruby installed:

$ ruby –v

1. Verify that SQLite3 is correctly installed and in your PATH:

$ sqlite3 --version

The program should report its version.

1. To install Rails, use gem install, the command provided by RubyGems:

$ gem install rails

1. To verify you have everything installed correctly, you should be able to run the following:

$ rails --version

3.2 Creating the Blog Application

1. Use the new application generator to create everything necessary to start working on a new task:

$ rails new blog

🡪 This will create a Rails application called Blog in a blog directory, and install the gem dependencies that are already mentioned in Gemfile using bundle install.

To see all command line options that the Rails application builder accepts, run: rails new –h

1. Switch to the blog application folder:

$ cd blog

**4) Hello, Rails!**

* 1. Starting up the Web Server

1. To see the Rails app, start a web server on your dev machine. Run the following in the blog directory:

$ bin/rails server

🡪 This will fire up Puma, a web server distributed with Rails by default. To see the app in action, go to <http://localhost:3000> in the browser

4.2 Get Rails to say “Hello”

1. To do this, you need to create a *controller* and a *view*
   1. The *controller* receives specific requests for the app – this is where information is collected (then, is displayed by the view).
   2. *Routing* decides which controller receives which requests
   3. There is often more than one route to each controller, and different routes can be served by different *actions*.
   4. Each action’s purpose is to collect info and provide it to a view, which displays this info in human readable format.
      1. By default, view templates are written in a language called eRuby (embedded Ruby) which is processed by the request cycle in Rails before being sent to the user.
2. To create a new controller, run the “controller” generator and tell it you want a controller called “Welcome” with an action called “index”:

$ bin/rails generate controller Welcome index

🡪 Rails will create several files and a route for you.

create app/controllers/welcome\_controller.rb

route get ‘welcome/index’

invoke erb

create app/views/welcome

create app/views/welcome/index.html.erb

invoke test\_unit

create test/controllers/welcome\_controller\_test.rb

invoke helper

create app/helpers/welcome\_helper.rb

invoke test\_unit

invoke assets

invoke coffee

create app/assets/javascripts/welcome.coffee

invoke scss

create app/assets/stylesheets/welcome.scss

1. Open the app/views/welcome/index.html.erb file in your text editor. Delete all of the existing code in the file, and replace it with:

<h1>Hello, Rails!</h1>

4.3 Setting the App Home Page

Now that we’ve made the controller and view, we need to tell Rails when we want “Hello, Rails!” to show up; in our case, we want it to show up when we navigate to the root URL of the site, <http://localhost:3000> - right now, “Welcome Aboard” is occupying that spot (I think this is ‘Yay! You’re on Rails!’ 🡪 is that right?)

1. To tell Rails where the home page is located:

* Open the file config/routes.rb in the text editor (Sublime, in my case) – the config sub-folder is a direct sub-folder of ‘blog’
  + This is the application’s routing file, which holds entries in a special DSL (domain-specific language) that tells Rails how to connect incoming requests to controllers and actions.
* Edit the file by adding the line of code root ‘welcome#index’ after get ‘wecome/index’ so that it looks like:

Rails.application.routes.draw do

get 'welcome/index'

root ‘welcome#index’

# For details on the DSL available within this file, see http://guides.rubyonrails.org/routing.html

end

* root ‘welcome#index’ tells Rails to map requests to the root of the application, to the welcome controller’s index action
* get ‘welcome/index’ tells Rails to map requests to <http://localhost:3000/welcome/index> to the welcome controller’s index action (this was created when I ran the controller action earlier (bin/rails generate controller Welcome index)
* NOW, the web server should be running the “Hello, Rails!” message I put in to app/views/welcome/index.html.erb, indicating that this new route is going to WelcomeController’s index action and is rendering the view correctly.
* **Q: WHAT WAS THE ROOT PREVIOUSLY? WHERE WAS THE ORIGINAL ‘YAY RAILS ETC.’ MESSAGE COMING FROM?**

**5) Getting Up and Running**

Now that the controller, an action and a view have been created, I’m going to create a new **resource** in the Blog application.

* A **resource** is a term used for a collection of similar objects, such as articles, people or animals.
  + You can create, read, update and destroy items for a resource 🡪 these operations are referred to as *CRUD* operations
* Rails provides a *resources* method which can be used to declare a standard *REST resource*. You need to add the *article resource* to the config/routes.rb so the file will look as follows:

Rails.application.routes.draw do

get 'welcome/index'

resources :articles

root 'welcome#index'

# For details on the DSL available within this file, see http://guides.rubyonrails.org/routing.html

end