In this lecture, we will discuss…

- ✧ Brief Rails history
- ✧ Benefits of using Rails
- ✧ Model View Controller

History - Ruby on Rails (RoR)

- Framework for making **dynamic web applications**
- Created in 2004 - 2005 by David Heinemeier Hansson

Who is Using Rails?

Why Use Rails?

- **C**onvention **O**ver **C**onfiguration (COC)
  - Less code to write
    - Some code Rails automatically generates for you
    - Oftentimes, there is no need to write code at all
  - Learn it once - know what to expect the next time

Why Use Rails?

- ✧ Database Abstraction Layer
  - No need to deal with low-level DB details
  - No more SQL (*Almost*)
  - Important to understand the SQL generated!

Why Use Rails?

- ✧ Agile-friendly
- ✧ **D**on’t **R**epeat **Y**ourself (DRY) principle
- ✧ Cross-platform
- ✧ Open Source
- ✧ Modular

SQLite

- Rails uses SQLite for database by default
  - Self-contained, serverless, zero-configuration, transactional, relational SQL database engine.

CLAIM: Most widely deployed SQL database engine in the world

MVC: Model View Controller

- Invented in 1979 by Trygve Reenskaug
- Well-established software pattern used by many web and desktop frameworks
- ***Model*** - represents the data the application is working with (and possibly business logic)
- ***View*** – (visual) representation of that data
- ***Controller*** - orchestrates interaction between the model and the view

MVC Cycle

1. Request sent
2. Controller ↔ Model
3. Controller invokes View
4. View renders data

Controller

View

Model

DB

Summary

- ✧ Rails is very good for **Rapid Prototyping**
- ✧ MVC and Convention over Configuration enable you to **“think less and do more”**

What’s Next

- ✧ Creating your first Rails app

In this lecture, we will discuss…

- ✧ How to **create and run** your app
- ✧ Directory structure (CoC)
- ✧ Adding **static pages** to your application

Creating First App

```
~$ rails new my_first_app
      create
      create  README.rdoc
      create  Rakefile
      create  config.ru
      create  .gitignore
      create  Gemfile
      create  app
      create  app/assets/javascripts/application.js
      create  app/assets/stylesheets/application.css
      create  app/controllers/application_controller.rb
      create  app/helpers/application_helper.rb
      create  app/views/layouts/application.html.erb
      create  app/assets/images/.keep
      create  app/mailers/.keep
      create  app/models/.keep
      create  app/controllers/concerns/.keep
      create  app/models/concerns/.keep
      create  bin
      create  bin/bundle
      create  bin/rails
      create  bin/rake
      create  bin/setup
      create  config
      create  config/routes.rb
```

rails new appname

(rails new -h for more options)

Bundler (gems manager)

```
      create  test/integration/.keep
      create  test/test_helper.rb
      create  tmp/cache
      create  tmp/cache/assets
      create  vendor/assets/javascripts
      create  vendor/assets/javascripts/.keep
      create  vendor/assets/stylesheets
      create  vendor/assets/stylesheets/.keep
         run  bundle install
Fetching gem metadata from https://rubygems.org/.............
Fetching version metadata from https://rubygems.org/...
Fetching dependency metadata from https://rubygems.org/..
Resolving dependencies...
Using rake 10.4.2
Using i18n 0.7.0
Using json 1.8.3
Using minitest 5.8.0
Using thread_safe 0.3.5
Using tzinfo 1.2.2
Using activesupport 4.2.3
```

Version Control Your Rails App!

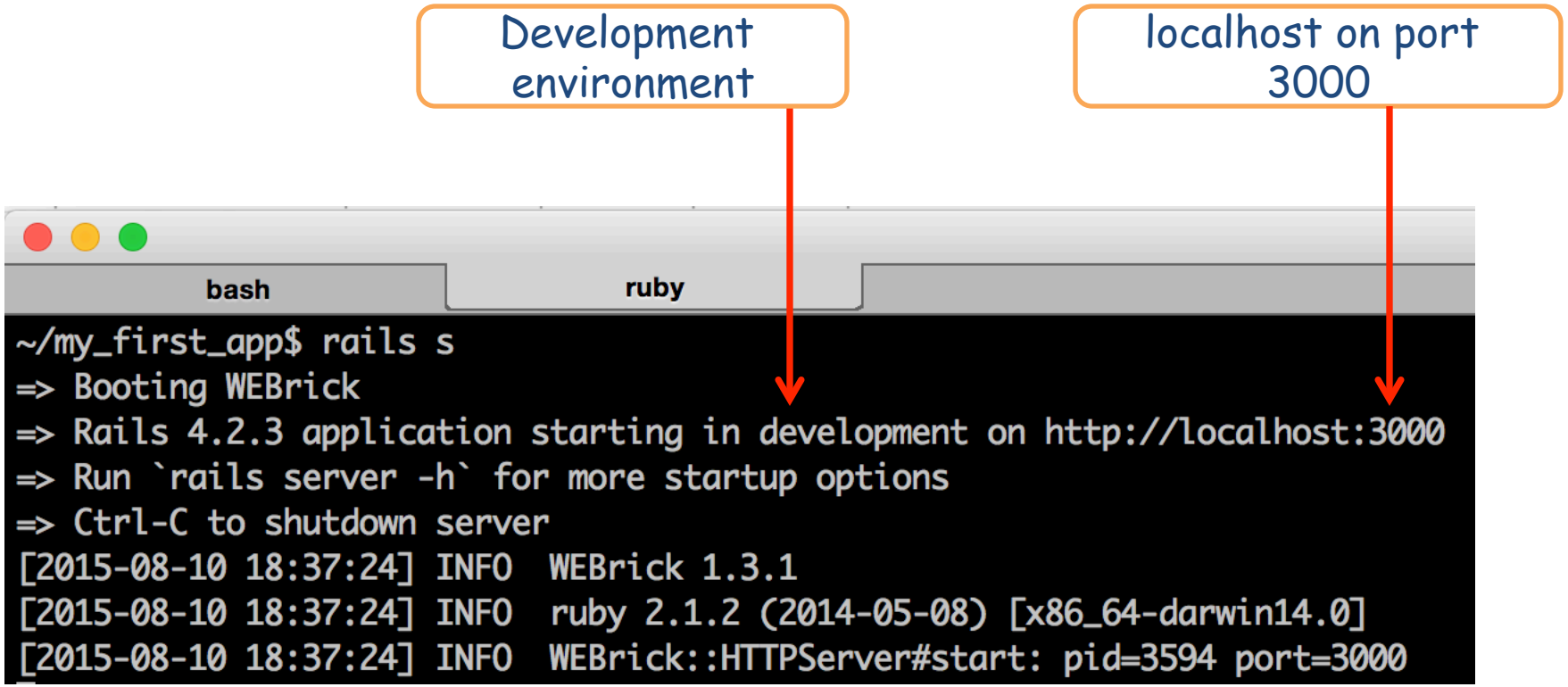
- Rails automatically generates `.gitignore`
- `cd my_first_app`
- `git init`
- `git add .`
- `git commit -m "Initial commit"`

Running the App

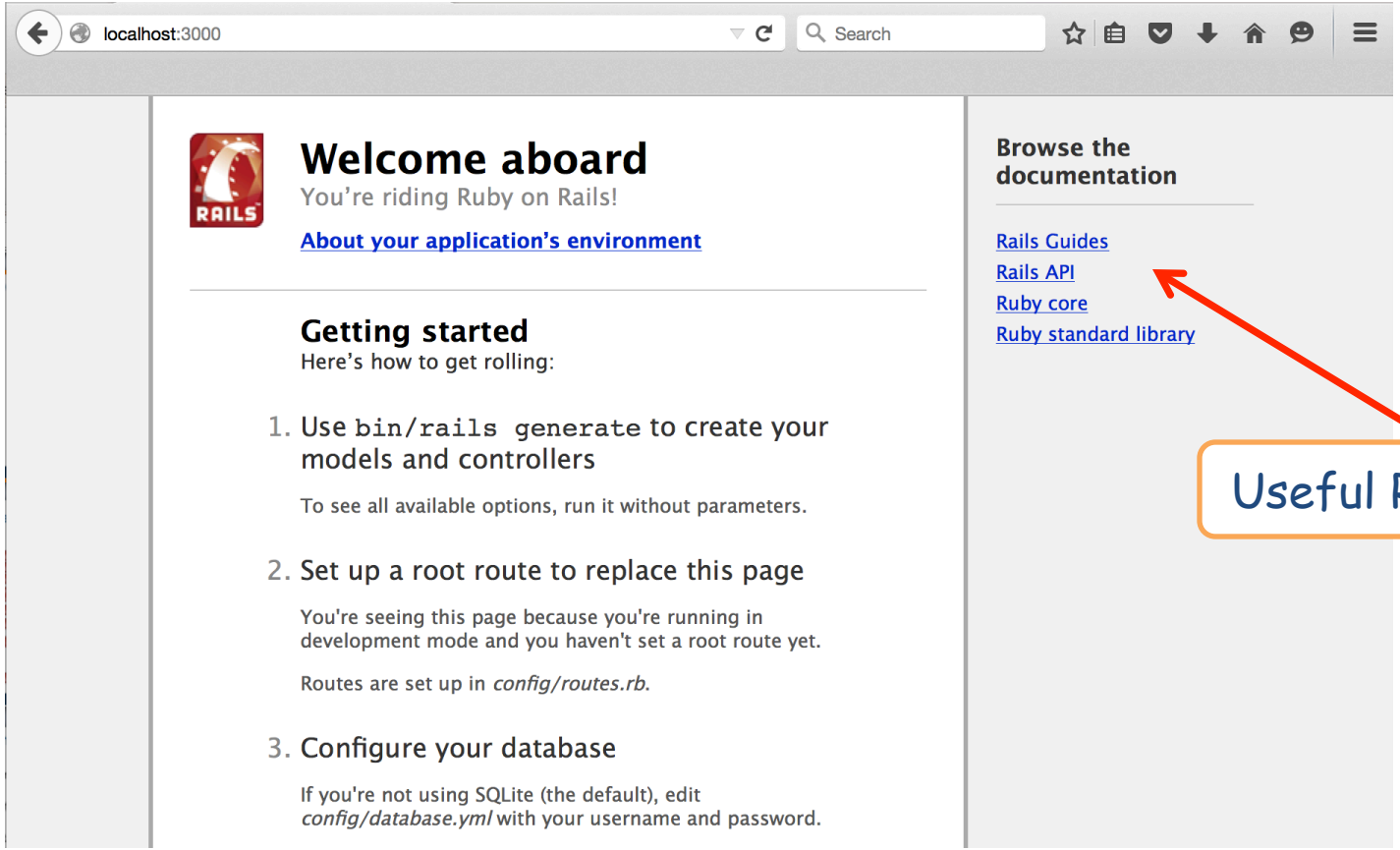
- Now you have `my_first_app` directory with auto-generated structure / (some) code
- Rails also provides a **built-in web server**

- Time to run our app!
  - ***(Optional)*** Open up another terminal window / tab
  - Go into `my_first_app` directory)
  - Run `rails server` (or `rails s`)

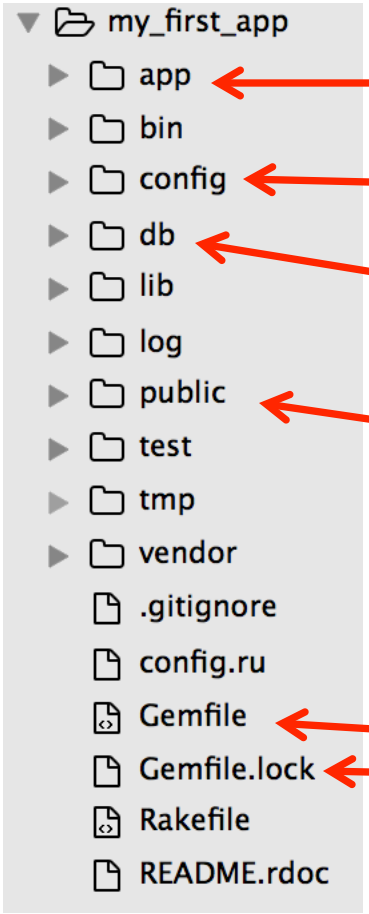
Running the App (Continued)



Running the App (Continued)



Directory structure convention



Controllers, Views, Models (and helpers)

Configuration files (which DB?)

Files related to your db and migration "scripts"

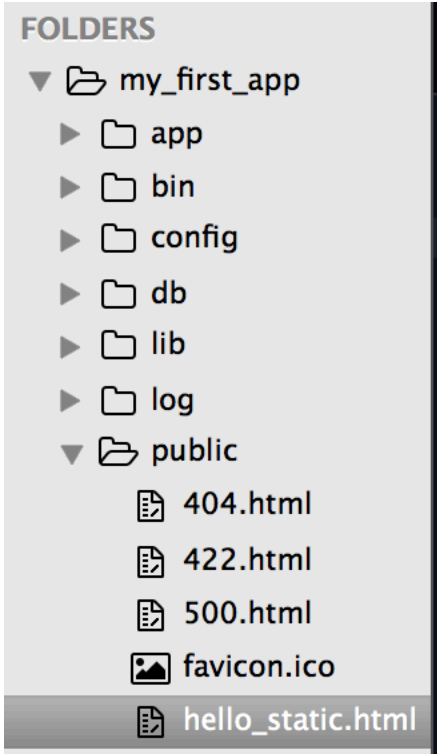
Static files

Dependencies managed by Bundler

public / hello_static.html

- Server looks into `public` directory **before** looking anywhere else
- So… if we want to add a **completely static web page** to our application - we can add it under `public` directory

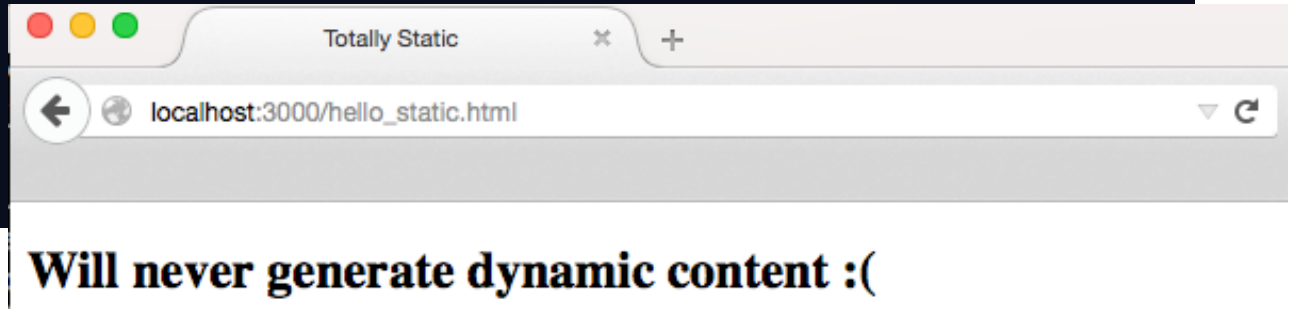
Adding hello_static.html to public dir



hello_static.html

```
1  <!DOCTYPE html>
2  <html>
3      <head><title>Totally Static</title></head>
4      <body><h2>Will never generate dynamic content :(</h2></body>
5  </html>
```

No need to restart the server - just refresh your browser



Summary

- ✧ `rails new` creates a new app
- ✧ `rails s` (or `rails server`) runs the app (need to be inside the generated directory)
- ✧ Static pages live inside `public` directory

What's next?

- ✧ Generating Controller and View

In this lecture, we will discuss…

- How to generate a controller
- Actions
- Embedded Ruby (ERB)

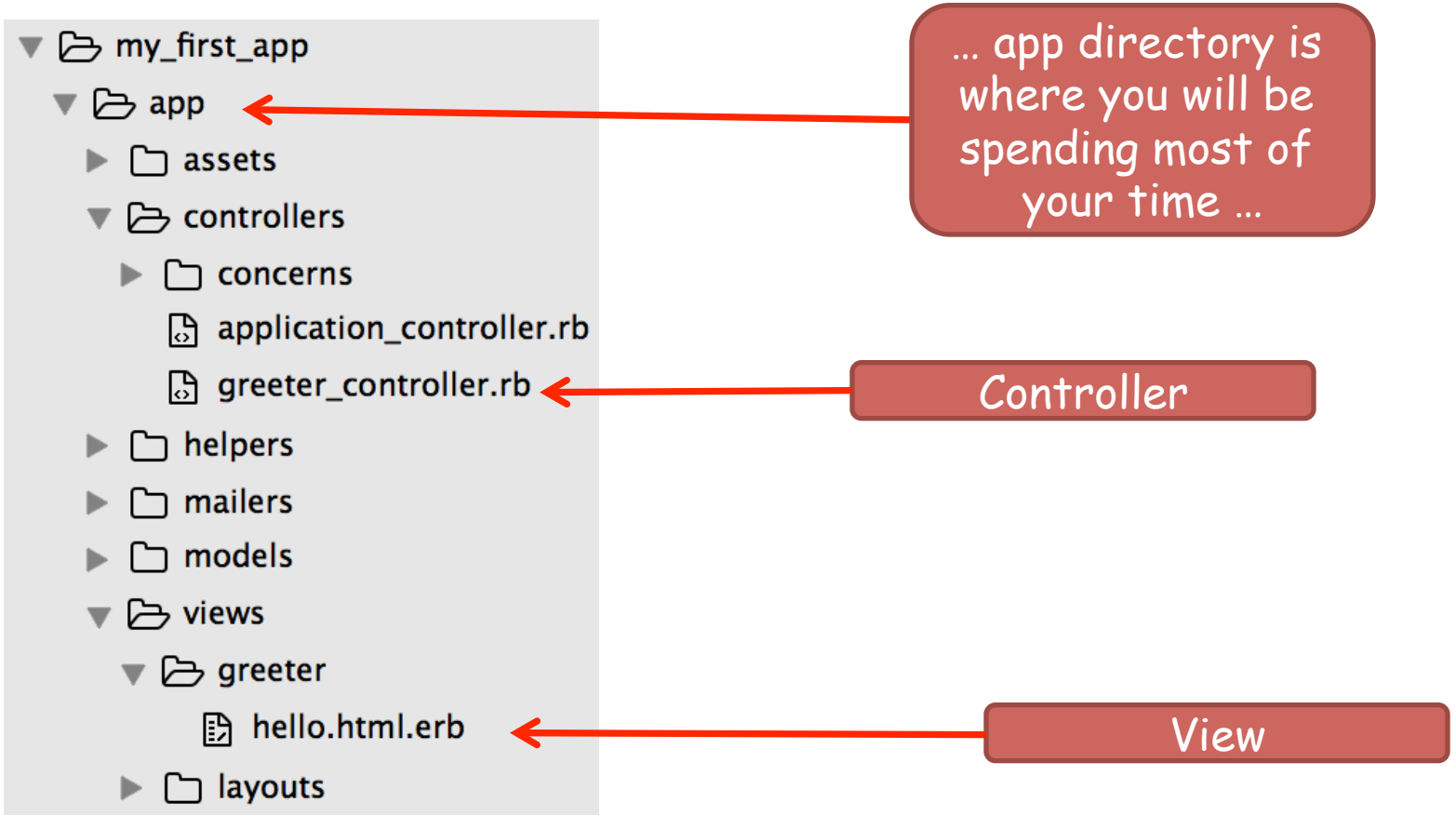
Generating a Controller

- Controllers contain **actions** (Ruby methods) and **orchestrate web requests**
- Rails can **quickly** generate a controller and **0 or more actions** with their **associated views**
- `rails generate controller controller_name [action1 action2]`
  - May substitute `g` for `generate`

Generating a Controller

```
~/my_first_app$ rails g controller greeter hello
      create  app/controllers/greeter_controller.rb
       route  get 'greeter/hello'
      invoke  erb
      create    app/views/greeter
      create    app/views/greeter/hello.html.erb
      invoke  test_unit
      create    test/controllers/greeter_controller_test.rb
      invoke  helper
      create    app/helpers/greeter_helper.rb
      invoke    test_unit
      invoke  assets
      invoke    coffee
      create      app/assets/javascripts/greeter.coffee
      invoke    scss
      create      app/assets/stylesheets/greeter.scss
```

Generating a Controller

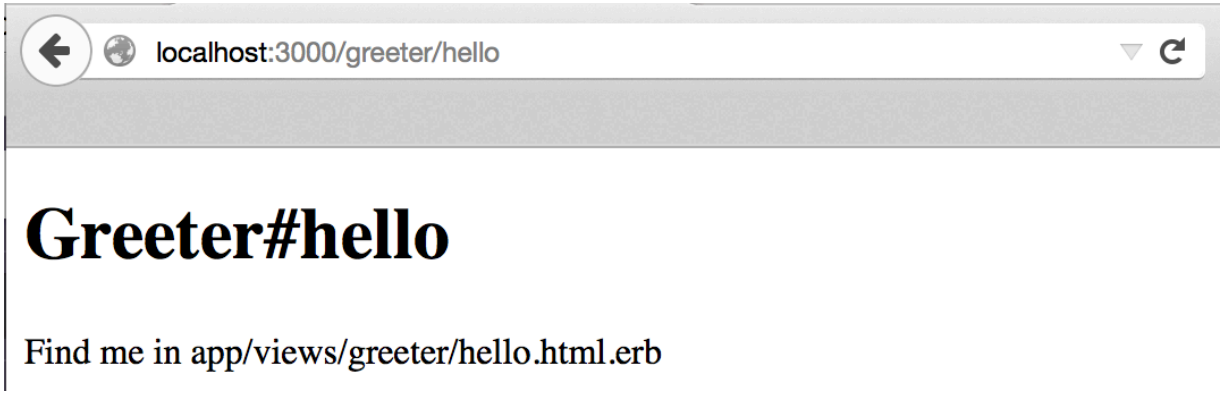


What Does It Look Like? (View)

No DOCTYPE, head or body elements…

hello.html.erb

```
1  <h1>Greeter#hello</h1>
2  <p>Find me in app/views/greeter/hello.html.erb</p>
```



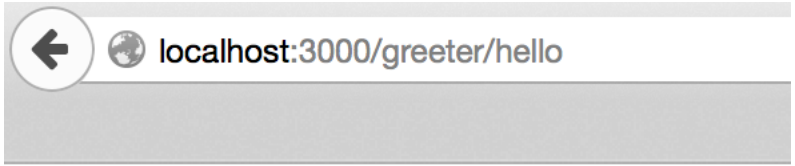
ERB (Embedded Ruby)

- The view file was generated and **it looks like it's an HTML file**, but it has an **.erb extension**
- ERB is a **templating library** (similar to JSP, for example) that lets you **embed Ruby into your HTML**
- Two **tag patterns** to learn:
  - `<% …ruby code… %>` - evaluate Ruby code
  - `<%= …ruby code… %>` - output evaluated Ruby code

New hello.html.erb

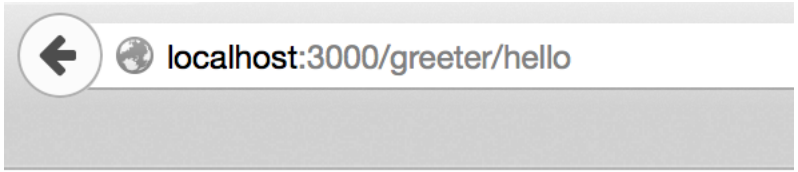
hello.html.erb

```
<% random_names = ["Alex", "Joe", "Michael"] %>
<h1>Greetings, <%= random_names.sample %></h1>
<p>The time now is <%= Time.now %> </p>
```



Greetings, Joe

The time now is 2015-08-12 16:48:57 -0400



Greetings, Alex

The time now is 2015-08-12 16:49:34 -0400

What Does It Look Like? (Controller)

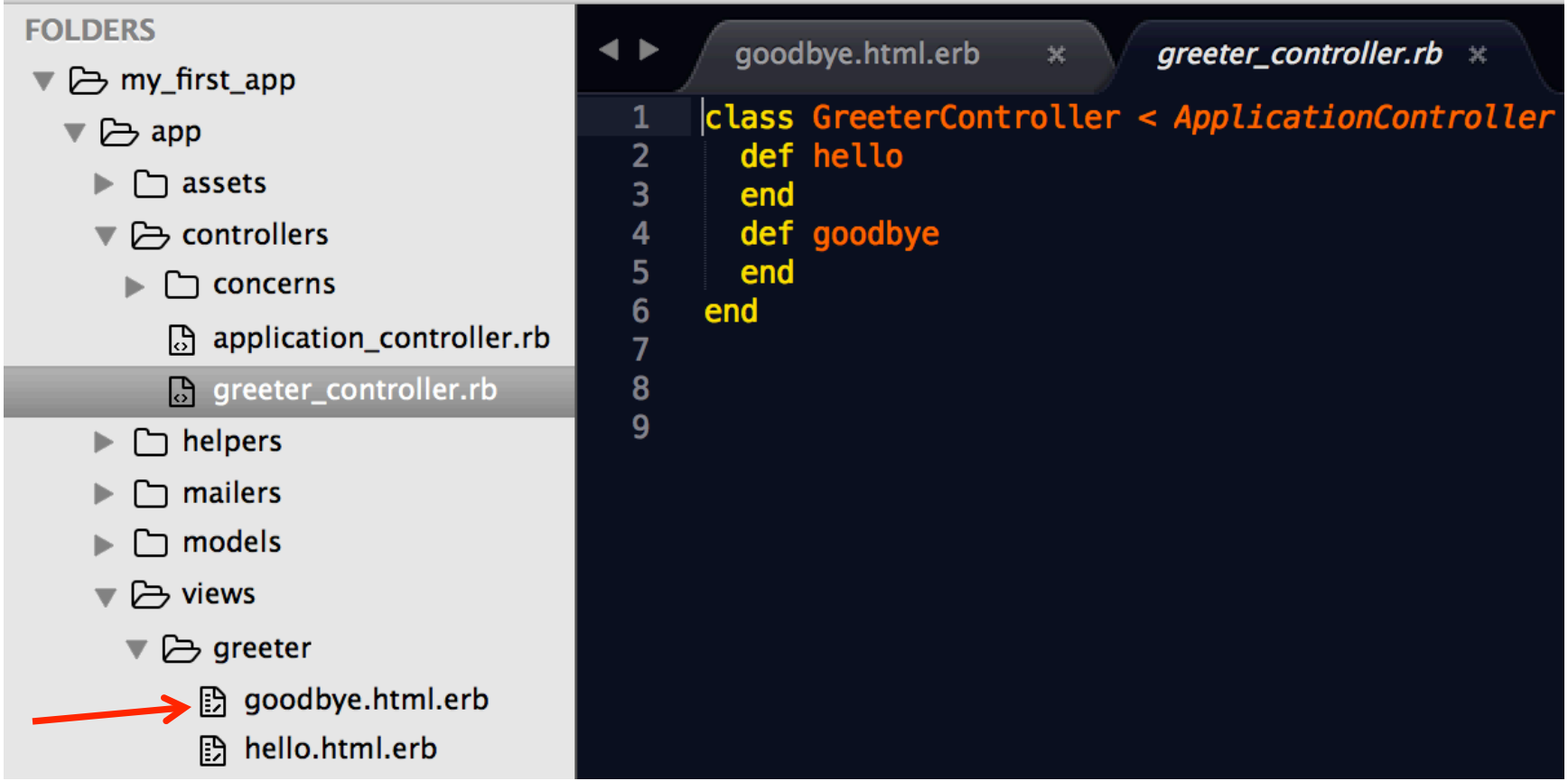
greeter_controller.rb

```
class GreeterController < ApplicationController
  def hello
  end
end
```

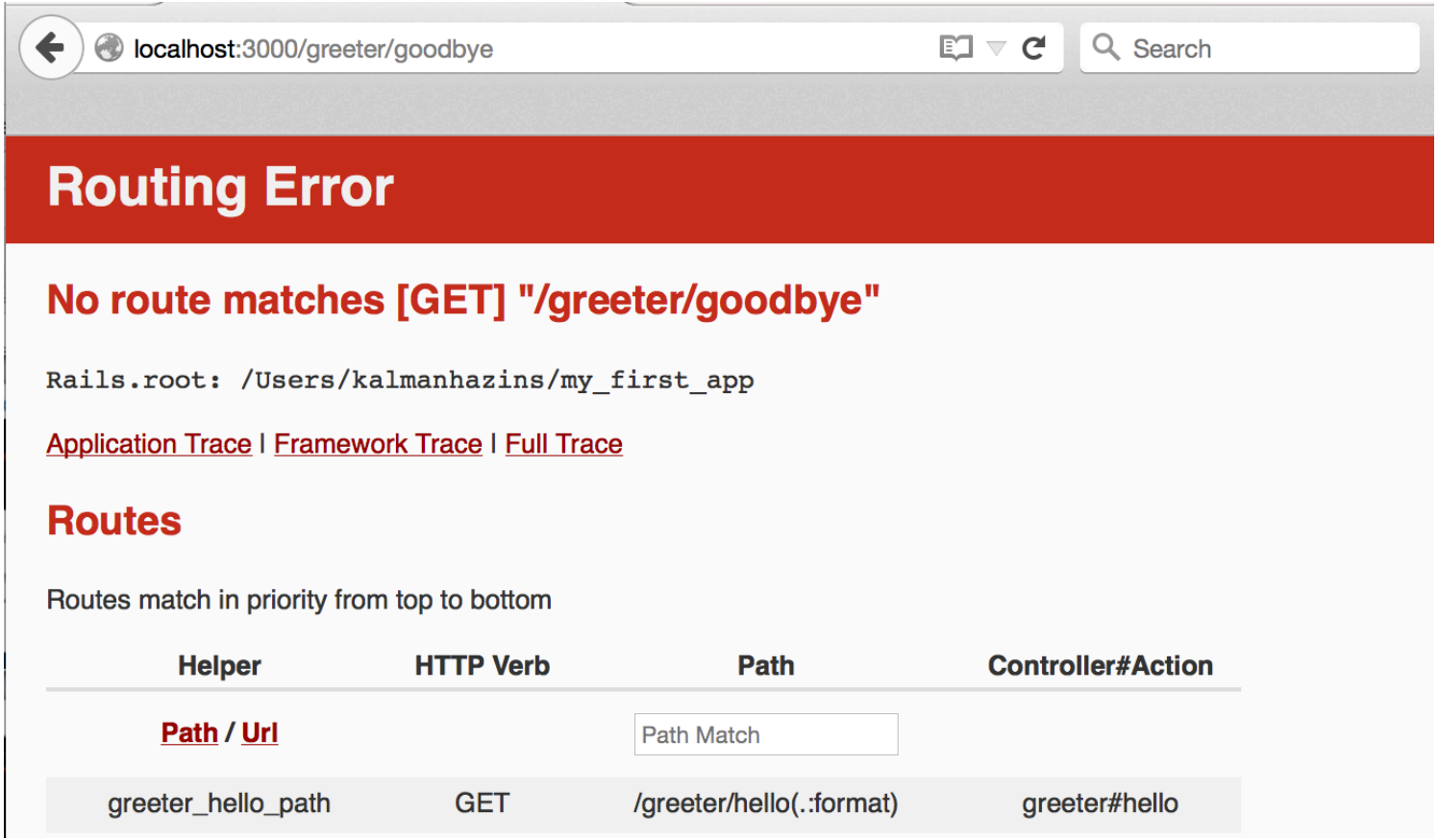
- ✧ `hello` action is just a **regular** (empty in this case) Ruby method
- ✧ What if we want to add a `goodbye` action to the `greeter` controller and also add a `goodbye.html.erb` to `app/views/greeter` directory?

Should work… ?

Adding Goodbye Action



Adding Goodbye Action



Summary

- Controllers **contain actions** (methods)
- ERB allows you to either **evaluate** expression with `<% expression %>` or **output** an expression `<%= expression %>`

What's Next?

- Routes

In this lecture, we will discuss…

- How routing works
- Rake
- How to analyze your current routes

Routes

- It turns out - **we are missing an important piece**

Routing

- Before the Controller can **orchestrate where the web request goes** - the web request **needs to get routed** to the Controller
- So, how did the `hello` action work?
- The route for `hello` action was **automatically generated** (during `rails g controller`)

MVC(R) Cycle

1. Request sent
2. Router routes request to Controller
3. Controller ←→ Model
4. Controller invokes View
5. View renders data

Router

Controller

View

Model

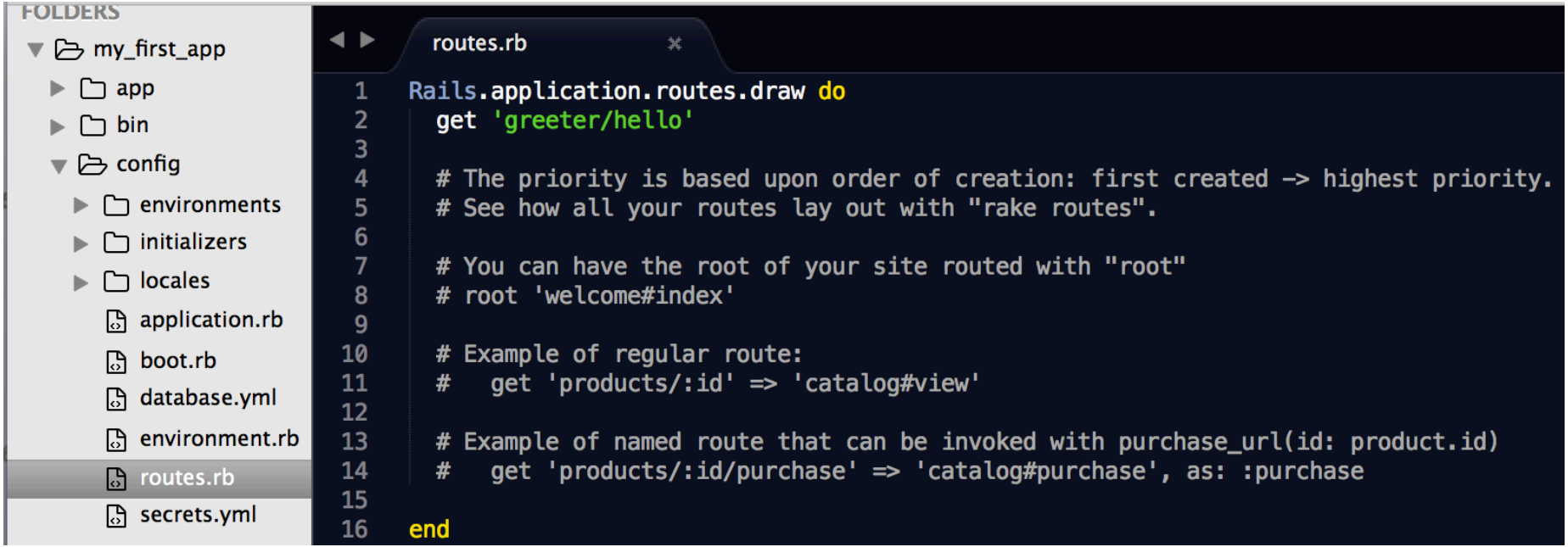
DB

routes.rb

- All the routes **need to be specified** (either generated by rails generators or manually) in the `config/routes.rb` file

So, what does `config/routes.rb` look like?

config/routes.rb



Let's add the route for the goodbye action

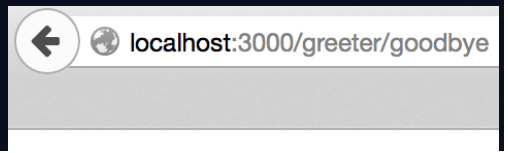
New config/routes.rb

routes.rb

```
Rails.application.routes.draw do
  # get 'greeter/hello'

  # SAME AS ABOVE
  get 'greeter/hello' => "greeter#hello"
  get 'greeter/goodbye'

  # The priority is based upon order of creation: first created -> highest priority.
  # See how all your routes lay out with "rake routes".

  # You can have the root of your site routed with "root"
  # root 'welcome#index'
end
```

Controller

Action



Bye, Joe

Rake

- ✧ Ruby's build language
  - **R**uby's m**ake**
  - No XML - **written entirely** in Ruby
- ✧ Rails uses rake to **automate app-related tasks**
  - Database, running tests, etc.
- ✧ To see a **list of rake tasks** for your app:
  - `rake --tasks`

Rake Tasks

```
~/my_first_app$ rake --tasks
rake about                                  # List versions of all Rails framew...
rake assets:clean[keep]                     # Remove old compiled assets
rake assets:clobber                         # Remove compiled assets
rake assets:environment                     # Load asset compile environment
rake assets:precompile                      # Compile all the assets named in c...
rake cache_digests:dependencies             # Lookup first-level dependencies f...
rake cache_digests:nested_dependencies      # Lookup nested dependencies for TE...
rake db:create                              # Creates the database from DATABAS...
rake db:drop                                # Drops the database from DATABASE_...
rake db:fixtures:load                       # Load fixtures into the current en...
rake db:migrate                             # Migrate the database (options: VE...
rake db:migrate:status                      # Display status of migrations
rake db:rollback                            # Rolls the schema back to the prev...
rake db:schema:cache:clear                  # Clear a db/schema_cache.dump file
rake db:schema:cache:dump                   # Create a db/schema_cache.dump file
rake db:schema:dump                         # Create a db/schema.rb file that i...
```

Individual Rake Task

- You can **zero-in** on an individual rake task and **what it does** with a `--describe` flag
- `rake --describe task_name`

```
~/my_first_app$ rake --describe routes
rake routes
    Print out all defined routes in match order, with names. Target specific controller with CONTROLLER=x.
```

Rake Routes

- `rake routes` explains your **currently defined routes**

```
~/my_first_app$ rake routes
       Prefix Verb URI Pattern                       Controller#Action
 greeter_hello GET  /greeter/hello(.:format)         greeter#hello
greeter_goodbye GET  /greeter/goodbye(.:format)     greeter#goodbye
```

Summary

- Router **directs the request** to the **right controller**
- `rake routes` lets you see **which routes are currently defined** for your application

What's Next?

- Moving business logic out of the View

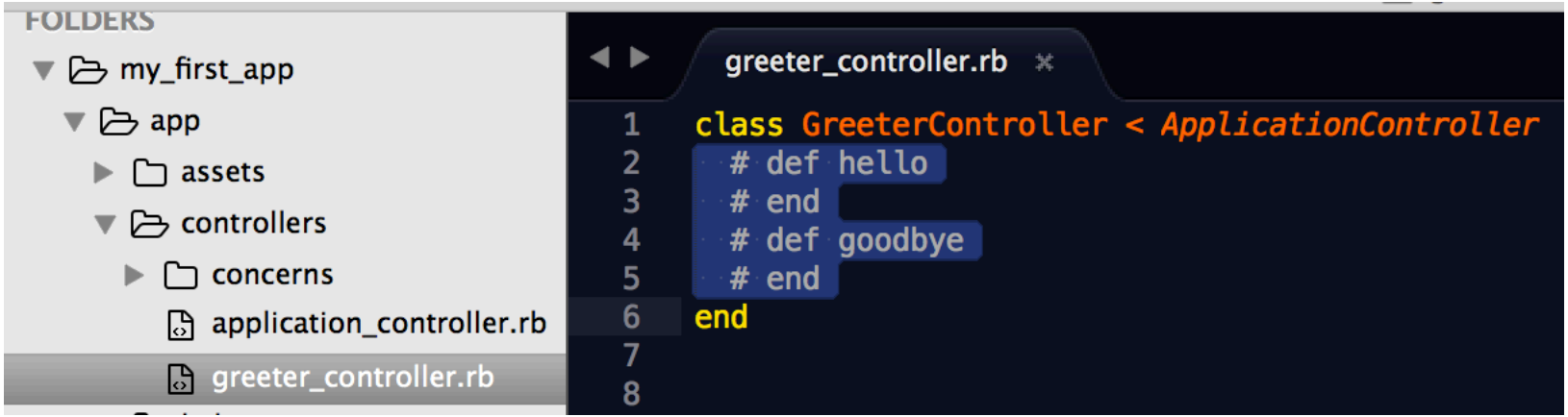
In this lecture, we will discuss…

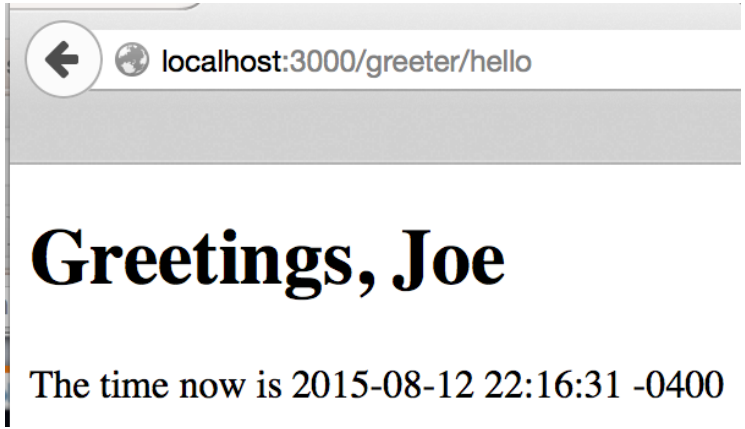
- How to move business logic out of View and into Controller in order to comply with the MVC pattern
- How long the controller's instance variables stay in place

Actions (Methods) Inside Controller

- If the action (method) is not really doing anything (i.e., it's empty) - **we can remove it**
- As long as there is a **proper route defined** and there is a **properly named view file/template,** the action method **does not have to be there** and Rails **will find the correct template** by convention

Controller: New Look





Moving Business Logic Out

- Our app “works”, but **business logic does not belong** in the **View**. The **View** should have **as little Ruby code logic** as possible

Instead, let’s move the logic to the controller!

Moving Business Logic Out

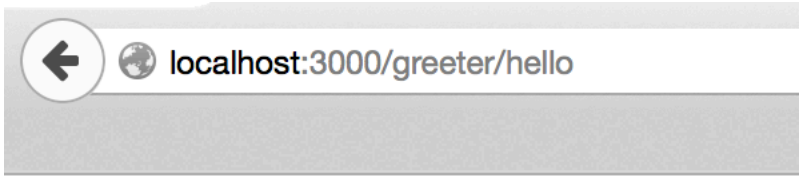
- Instance variables from the controller are available **inside the view**

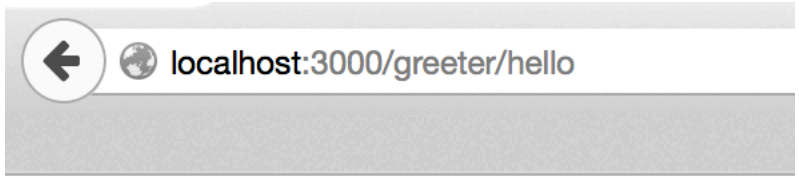
```
class GreeterController < ApplicationController
  def hello
    random_names = ["Alex", "Joe", "Michael"]
    @name = random_names.sample
    @time = Time.now
    @times_displayed ||= 0
    @times_displayed += 1
  end
end
```

New Views/Greeter/hello.html.erb

hello.html.erb

```
<h1>Greetings, <%= @name %></h1>
<p>The time now is <%= @time %> </p>
<p>This page has been viewed <%= @times_displayed %> time(s)</p>
```





Greetings, Michael

The time now is 2015-08-12 22:40:50 -0400

This page has been viewed 1 time(s)

1 both times?

Greetings, Alex

The time now is 2015-08-12 22:42:11 -0400

This page has been viewed 1 time(s)

Instance Variables in Rails

- Unlike some other web frameworks (Servlets), you **cannot "store" values in the controller's instance variables** in between requests
- Alternatives?
  - Session
  - Database

Summary

- ✧ Keep business logic **out of the view**
- ✧ Instance variables **in the controller** are **available to view**
- ✧ Instance variables **do not stick around** between requests

What's Next?

- ✧ Helpers

In this lecture, we will discuss…

- ✧ What are helpers and why are they useful
- ✧ Why use the `link_to` Rails helper

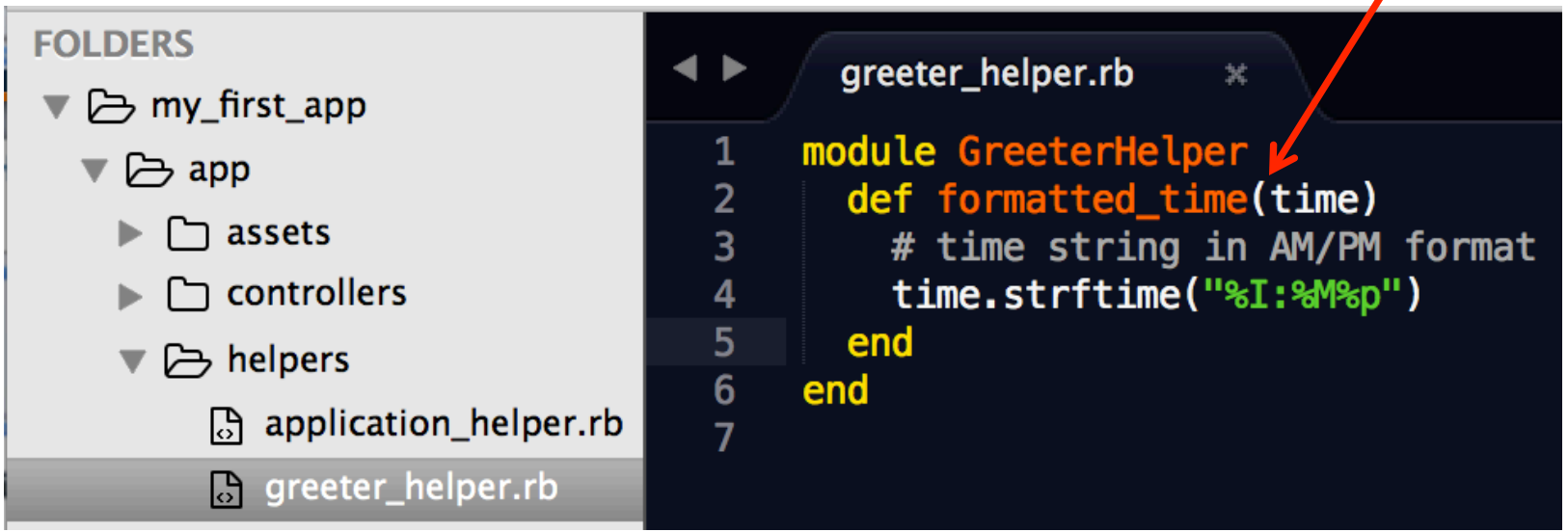
Helpers

- We've made the **current time available** through `@time` instance variable (controller)
- But what if we want to format **how the time looks**?
  - Should that code go in the View? Non-reusable…
  - Controller? Should be "view format" agnostic

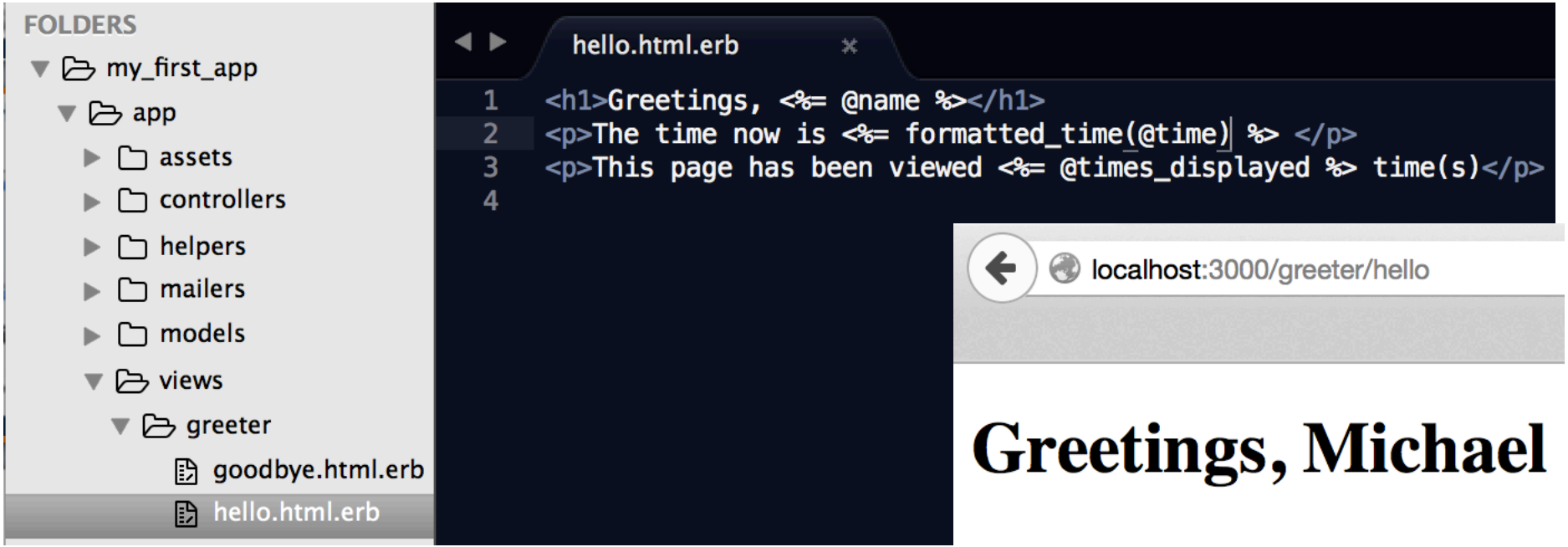
Helpers to the rescue!

Helpers

- (empty) `greeter_helper.rb` module generated
- Let's **add** a **helper method**



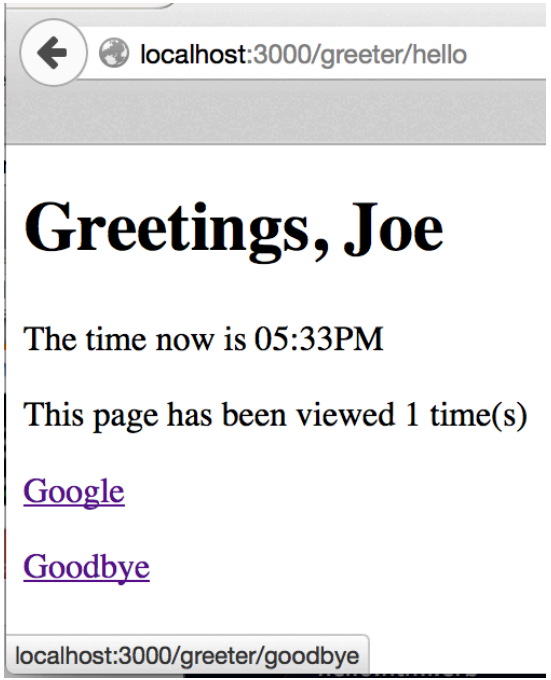
Helpers: New View



Rails Built-In Helpers: `link_to`

- Rails provides **many built-in helpers**
- `link_to name, path`
  - Hyperlink generator **that displays** the `name` and **links** to the `path`
  - Path could either be a **regular string** or a **route defined in the routes.rb file** ending with `_url` (full path) or `_path` (relative path)
  - `_url` and `_path` used **interchangeably**, but according to the spec full path is **required in cases of redirection**

`link_to` in action



hello.html.erb

```
<h1>Greetings, <%= @name %></h1>
<p>The time now is <%= formatted_time(@time) %> </p>
<p>This page has been viewed <%= @times_displayed %> time(s)</p>

<p><%= link_to "Google", "http://www.google.com" %></p>
<p><%= link_to "Goodbye", greeter_goodbye_path %></p>
```

greeter_goodbye derived from routes.rb
(See "Prefix" column in rake routes)

Summary

- ✧ Helpers are **“macros” / “formatters”** for your view
- ✧ When using `link_to`, there is **no need** to change things if a path changes

What’s Next?

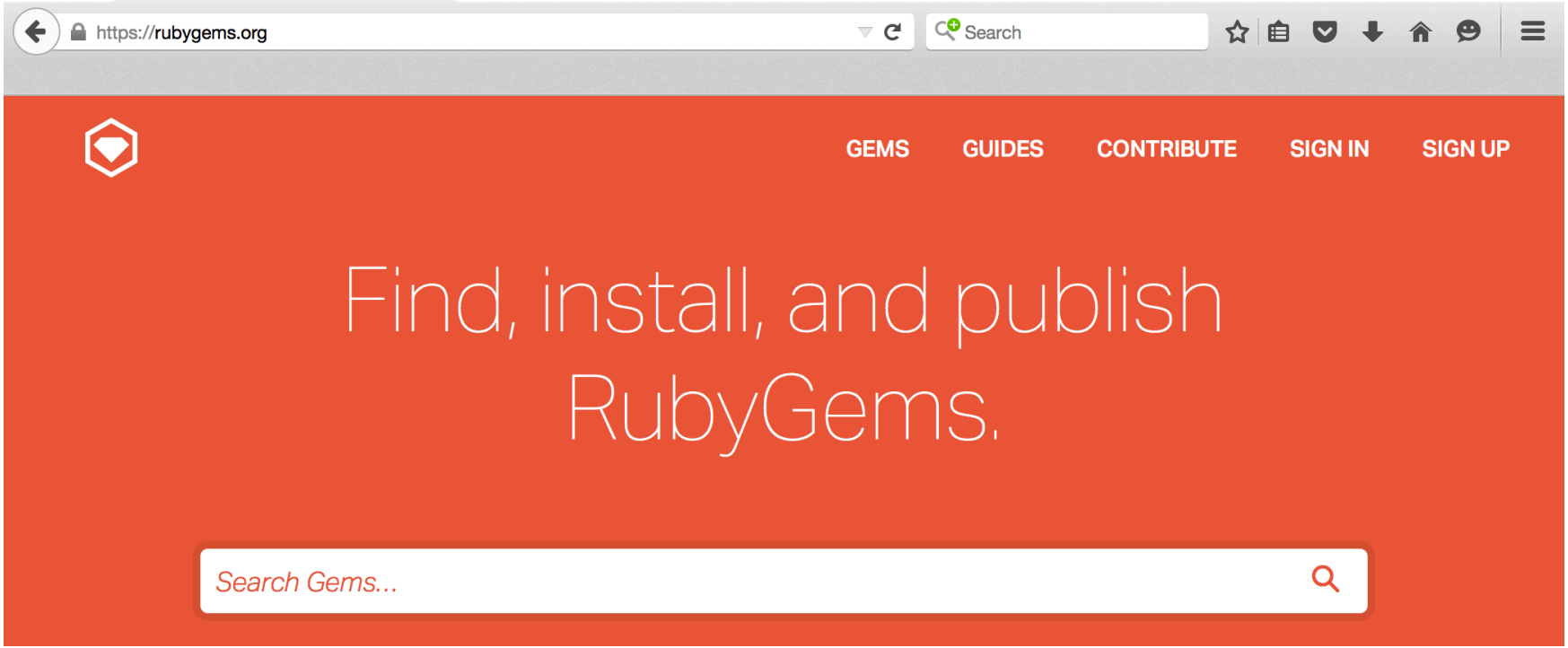
- ✧ Introduction to HTTParty

In this lecture, we will discuss…

- ✧ Ruby gems
- ✧ How to use the HTTParty Ruby gem

RubyGems

- ✧ Many gems (3rd party plugins / additions) for Ruby



RubyGems Package Manager

```
~$ gem
RubyGems is a sophisticated package manager for Ruby.  This is a
basic help message containing pointers to more information.

  Usage:
    gem -h/--help
    gem -v/--version
    gem command [arguments...] [options...]

  Examples:
    gem install rake
    gem list --local
    gem build package.gemspec
    gem help install
```

Installing HTTParty

```
~$ gem list httparty

*** LOCAL GEMS ***


~$ gem install httparty
Fetching: httparty-0.13.5.gem (100%)
When you HTTParty, you must party hard!
Successfully installed httparty-0.13.5
Parsing documentation for httparty-0.13.5
Installing ri documentation for httparty-0.13.5
Done installing documentation for httparty after 0 seconds
1 gem installed
```

More Details on Gem Command

- Use `-d` option to **get more details**

```
~$ gem list httparty -d

*** LOCAL GEMS ***

httparty (0.13.5)
    Authors: John Nunemaker, Sandro Turriate
    Homepage: http://jnunemaker.github.com/httparty
    License: MIT
    Installed at: /Users/kalmanhazins/.rbenv/versions/2.1.2/lib/ruby/gems/2.1.0

    Makes http fun! Also, makes consuming restful web services dead
    easy.
```

What Are Restful Web Services?

- **Simple** web services **implemented using HTTP** (and principles of REST) that:
  1. Have a **base URI**
  2. Support a **data exchange format** like XML or JSON (and possibly others)
  3. Support a **set of HTTP operations** (GET, POST etc.)

HTTParty Gem

- ✧ Restful web services **client**
- ✧ **Automatic parsing of JSON and XML into Ruby hashes**
- ✧ Provides **support** for:
  - Basic http authentication
  - Default request query parameters

Lots of Restful APIs Out There…

www.programmableweb.com/apis

Search

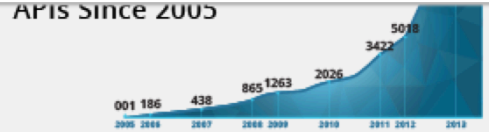
Follow ProgrammableWeb to get API news and alerts as they break | +Follow | Share | Sign In/Sign Up

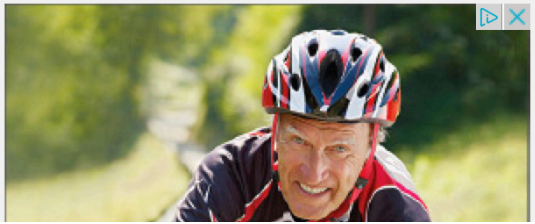
ProgrammableWeb

API News | API Directory | For API Providers | For Developers | Listings | Forum

MOST POPULAR APIs

1. Facebook	Track this API	6. Pinterest	Track this API
2. Google Maps	Track this API	7. Bloomberg	Track this API
3. Fitbit	Track this API	8. Twitter	Track this API
4. LinkedIn	Track this API	9. FedEx	Track this API
5. TripAdvisor	Track this API	10. Instagram	Track this API

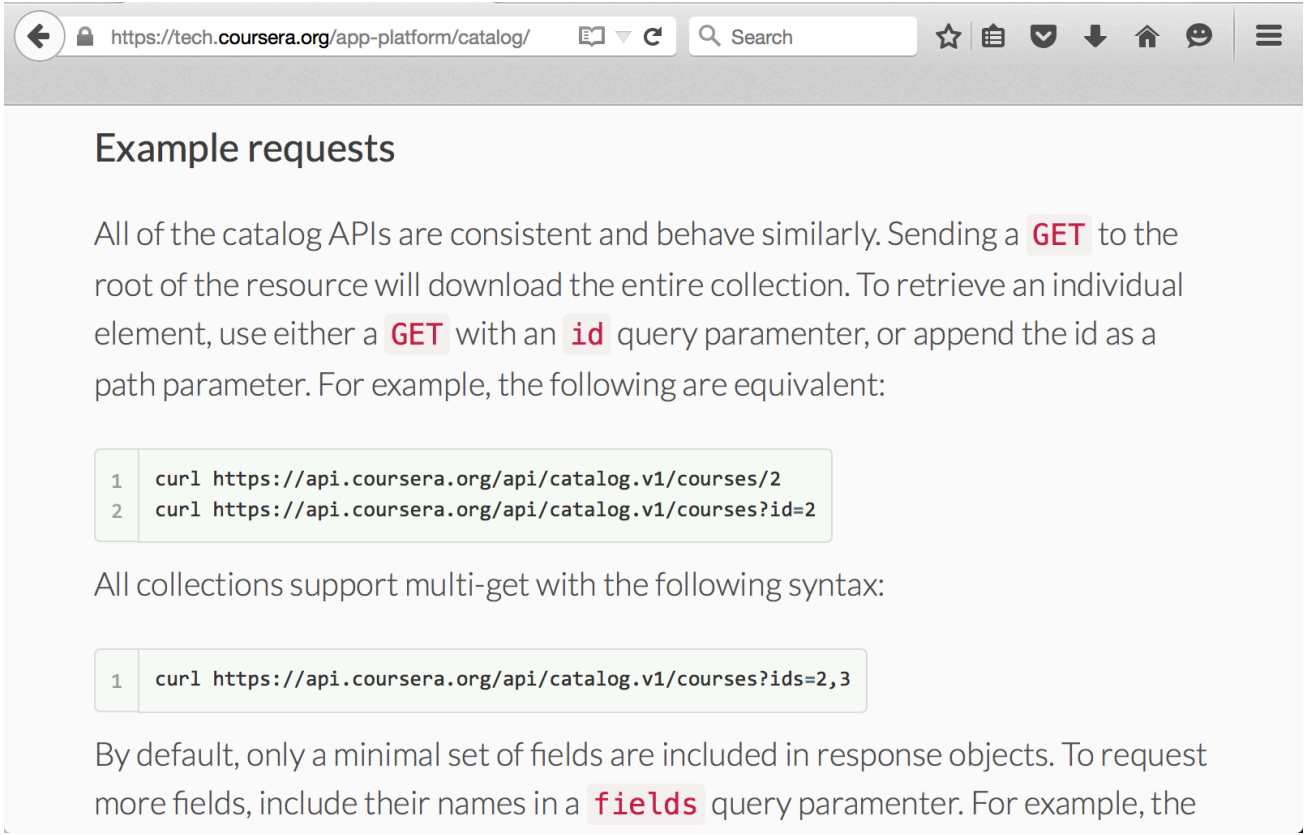




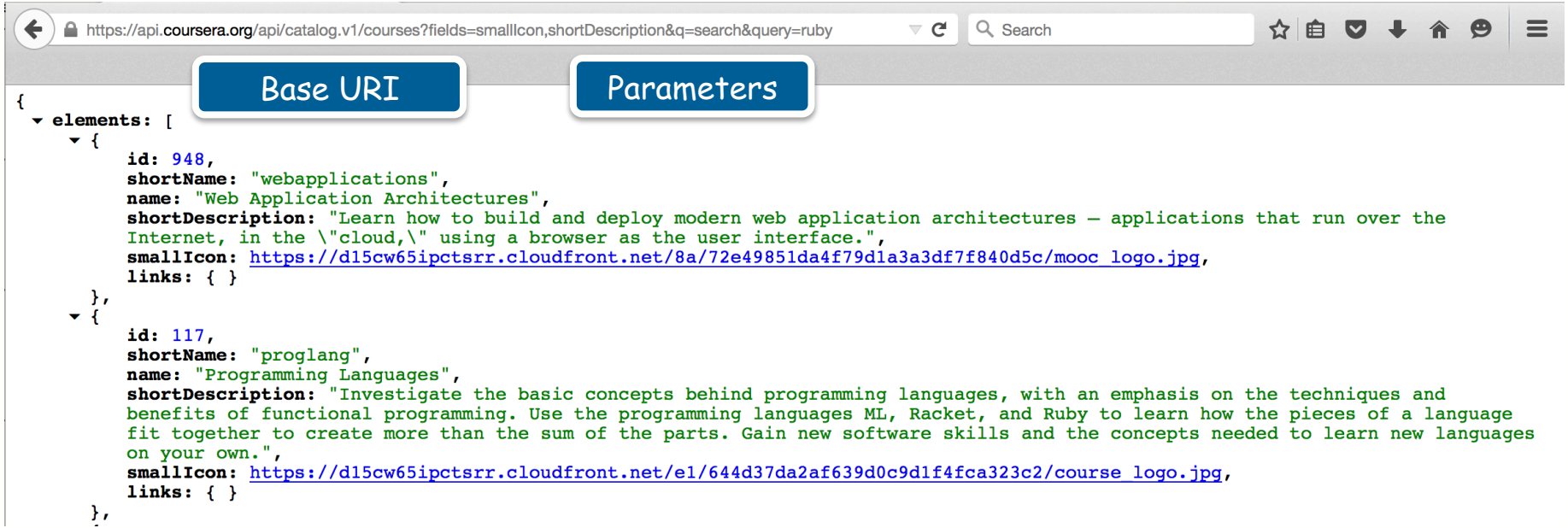
HTTParty Usage

- `include HTTParty` module
  - And you are good to go!
- Can specify:
  - `base_uri` for your **requests**
  - `default_params` (API developer key for example)
  - `format` to tell it **which format the data is in**

Coursera Restful API



Coursera Restful API



JSONView Browser Plugin (Available for Chrome and Firefox)

HTTParty Example

```ruby
require 'httparty'
require 'pp'

class Coursera
  include HTTParty

  base_uri 'https://api.coursera.org/api/catalog.v1/courses'
  default_params fields: "smallIcon,shortDescription", q: "search"
  format :json

  def self.for term
    get("", query: { query: term})["elements"]
  end
end

pp Coursera.for "python"
```

HTTParty Example Output


```
[{"id"=>2760,
  "shortName"=>"algorithmicthink1",
  "name"=>"Algorithmic Thinking (Part 1)",
  "shortDescription"=>
   "Experienced Computer Scientists analyze and solve computational problems at a level of a
   designed to train students in the mathematical concepts and process of \"Algorithmic Thin
  "smallIcon"=>
   "https://d15cw65ipctsrr.cloudfront.net/70/6ff6409dd811e49bfb61275b434ba9/AlgTh_logo.jpg",
  "links"=>{}},
 {"id"=>3048,
  "shortName"=>"molecularevolution",
  "name"=>"Deciphering Molecular Evolution (Bioinformatics IV)",
  "shortDescription"=>
   "In this course, we will see how evolutionary trees resolve quandaries from finding the o
   from computational proteomics to test whether we can reconstruct Tyrannosaurus rex protei
  "smallIcon"=>
   "https://d15cw65ipctsrr.cloudfront.net/16/55b1b0f3c111e4a6a209002642cb9f/evo.jpg",
  "links"=>{}},
```

Summary

- ✧ HTTParty makes it **extremely easy** to ingest Restful services, converting them to Ruby Hashes
- ✧ **JSON** and **XML** formats **supported**

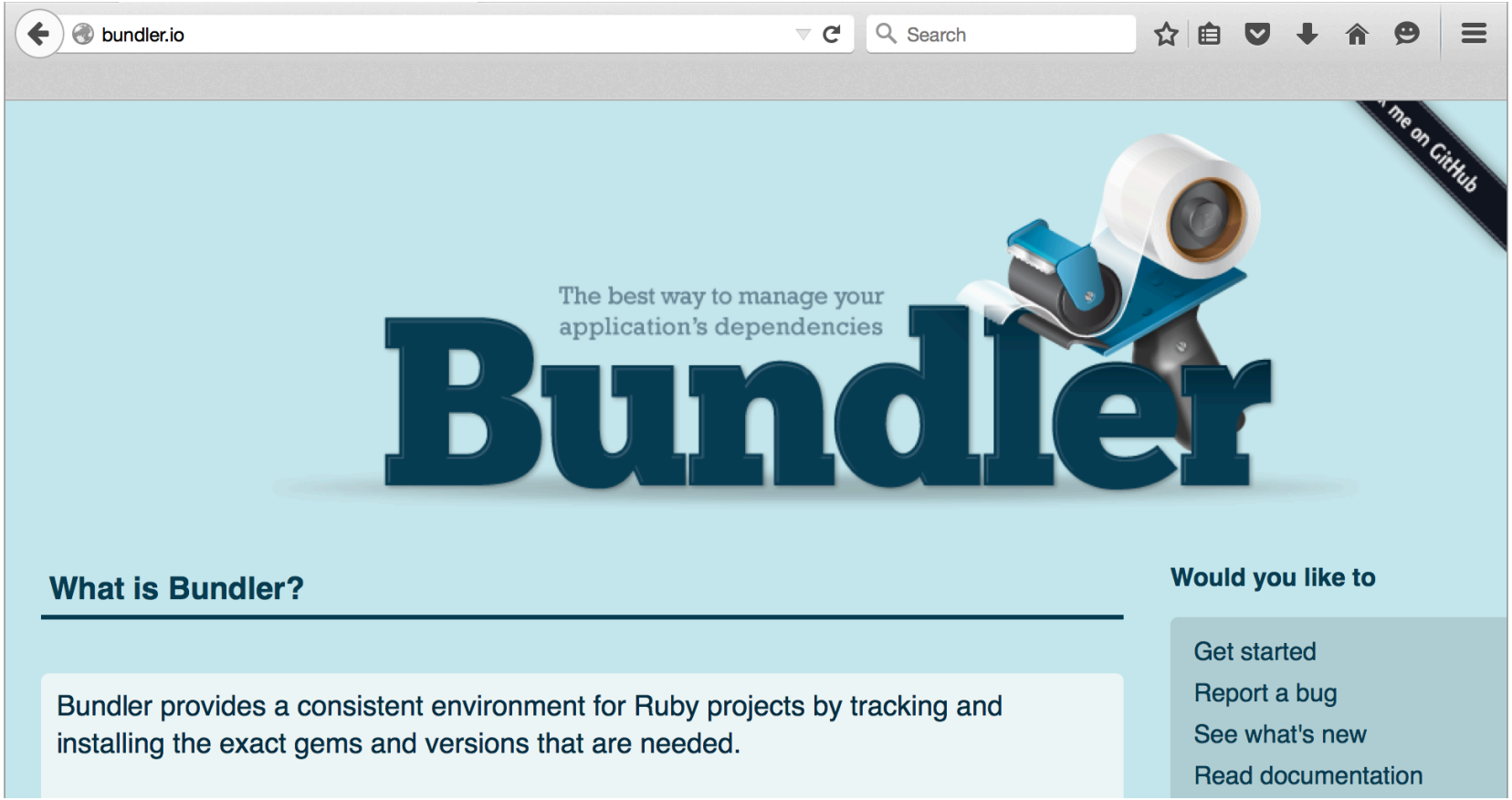
What's next?

- ✧ Bundler

In this lecture, we will discuss…

- Bundler
- Managing gems inside your Rails application

Bundler


bundler.io
Search
me on GitHub
The best way to manage your application's dependencies
Bundler
What is Bundler?
Bundler provides a consistent environment for Ruby projects by tracking and installing the exact gems and versions that are needed.
Would you like to
Get started
Report a bug
See what's new
Read documentation

Bundler

- Lets you **specify gems** (and associated gem dependencies) for this Rails app inside `Gemfile` (in the root of your Rails app)
- **Preferred way** to **manage gem dependencies** in Rails
- Run `bundle install` or simply `bundle` after specifying a new gem in the Gemfile
- Run `bundle update` when **modifying** a version of a gem

Bundler

- You can **instruct Rails** (through `Gemfile`) to **only load certain gems** in specific Rails environments

```
group :development, :test do
  # Call 'byebug' anywhere in the code to stop execution and get a debugger console
  gem 'byebug'

  # Access an IRB console on exception pages or by using <%= console %> in views
  gem 'web-console', '~> 2.0'

  # Spring speeds up development by keeping your application running in the background.
  gem 'spring'
end
```

Bundler – Which Version Of Gem?

- If you **don't specify** … You get the latest version
- Can specify an **exact version** or an **approximate version**

```
gem "nokogiri"
gem "rails", "3.0.0.beta3"
gem "rack", ">=1.0"
gem "thin", ">= 1.1", "< 2.0"
gem "thin", "~>1.1"
```

~> Pessimistic Version Constraint

Drop the final digit, then increment to get the upper limit version number

Bundler: `require`

- Occasionally, the name of the gem to be used inside `require` statement is **different** than the name of the gem

```
gem 'sqlite3-ruby', require: 'sqlite3'
```

Gemfile - Example

```
source 'http://rubygems.org'
gem 'rails', '4.2.3'
# Bundle edge Rails instead:
# gem 'rails', github:'git://github.com/rails/rails.git'
gem 'sqlite3'
…
```

- Our app can even use a **different version** of Rails if you **change the version** and run `bundle update`
- Bundler creates a `Gemfile.lock` file, which **contains the actual gem versions** your app is using with their **associated dependencies**

Bundle exec?


Bundler: The best way to
bundler.io/v1.6/bundle_exec.html
Apps
TapIt!
TapIt!
Adblock Plus has
The best way to manage your application's dependencies
Bundle
bundle exec
Run the command in context of the bundle
$ bundle exec
Exec runs a command, providing it access to the gems in the bundle. While using bundle exec you can require and call the bundled gems as if they were installed into the systemwide Rubygems repository.

Summary

- ✧ Bundler **manages** gem dependencies
- ✧ **Loads** gems on application startup

What's Next

- ✧ Integrating HTTParty and Rails

In this lecture, we will discuss…

- Integrating Rails with HTTParty

HTTParty Integration - Gemfile

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
14  group :development, :test do
15    gem 'byebug'
16    gem 'web-console', '~> 2.0'
17    gem 'spring'
18  end
19
20  gem 'httparty', '0.13.5'
```

HTTParty Integration - Gemfile

```
Installing httparty 0.13.5
Using multi_json 1.11.2
Using jbuilder 2.3.1
Using jquery-rails 4.0.4
Using bundler 1.9.5
Using sprockets 3.2.0
Using sprockets-rails 2.3.2
Using rails 4.2.3
Using rdoc 4.2.0
Using sass 3.4.16
Using tilt 1.4.1
Using sass-rails 5.0.3
Using sdoc 0.4.1
Using spring 1.3.6
Using sqlite3 1.3.10
Using turbolinks 2.5.3
Using uglifier 2.7.1
Using web-console 2.2.1
Bundle complete! 13 Gemfile dependencies, 56 gems now installed.
Use `bundle show [gemname]` to see where a bundled gem is installed.
Post-install message from httparty:
When you HTTParty, you must party hard!
~/my_first_app$
```

bundle (or bundle install)

NEED TO RESTART THE SERVER after running bundler for changes to take effect!!

Course Listing Example

- Let's show Courses based on search term
- Generate Courses Controller
- `rails g controller courses index`

Controller

Action

```
~/my_first_app$ rails g controller courses index
      create  app/controllers/courses_controller.rb
       route  get 'courses/index'
      invoke  erb
      create    app/views/courses
      create    app/views/courses/index.html.erb
      invoke  test_unit
      create    test/controllers/courses_controller_test.rb
      invoke  helper
      create    app/helpers/courses_helper.rb
      invoke    test_unit
      invoke  assets
      invoke    coffee
      create      app/assets/javascripts/courses.coffee
      invoke    scss
      create      app/assets/stylesheets/courses.scss
```

Courses Listing – `config/routes.rb`

routes.rb

```
Rails.application.routes.draw do
  get 'courses/index'

  # get 'greeter/hello'

  # SAME AS ABOVE
  get 'greeter/hello' => "greeter#hello"
  get 'greeter/goodbye'

  # You can have the root of your site routed with "root"
  # root 'welcome#index'
end
```

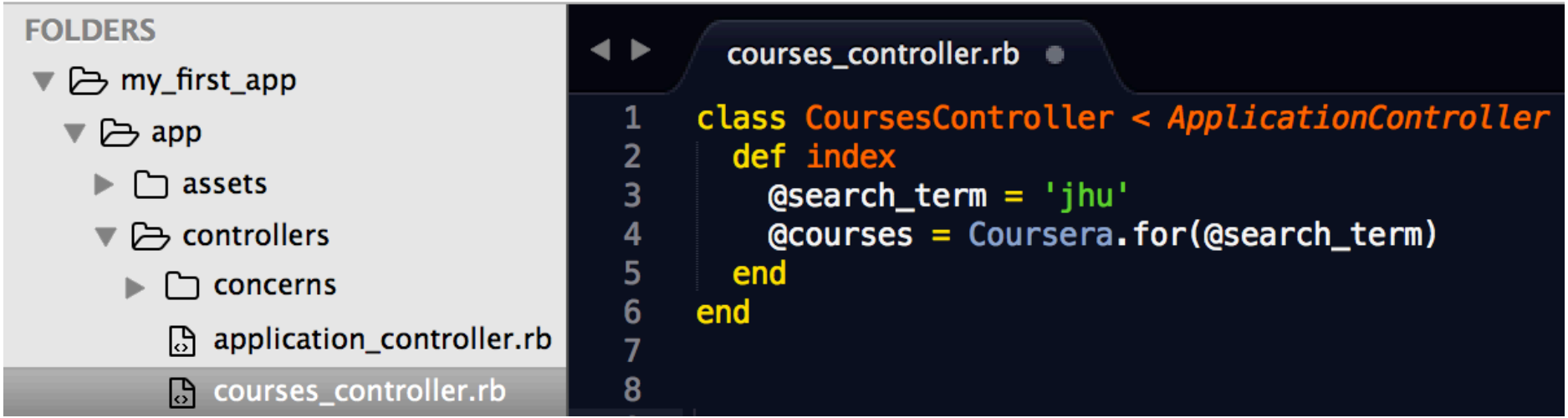
Coursera Model

By convention, controllers are named plural and model is named singular…
Also, note how HTTParty does not have to be **required** (thanks Bundler!)

Courses Controller

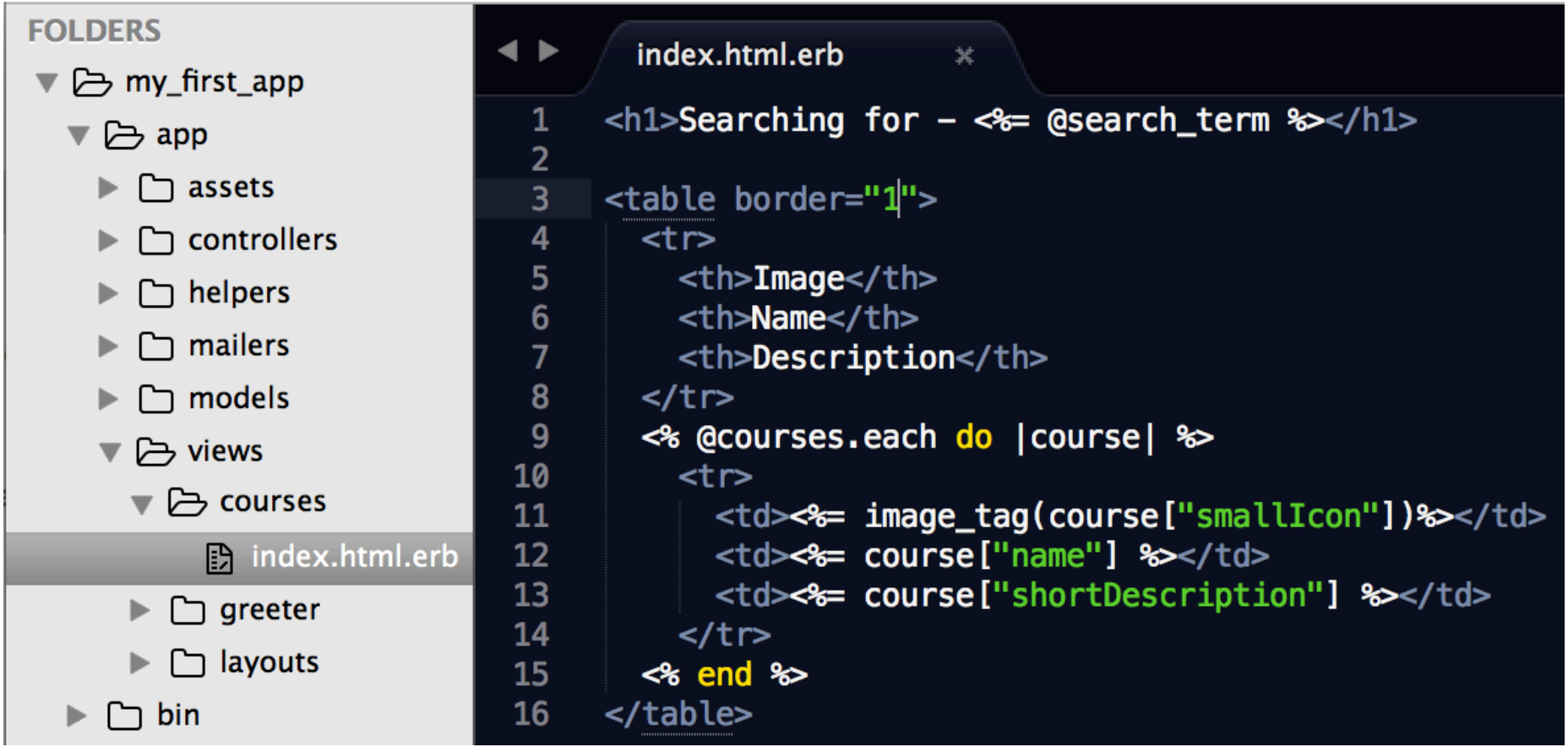
- Fill in `index` action

Assume jhu for now…



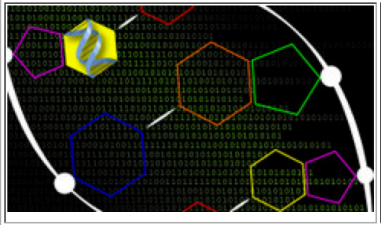
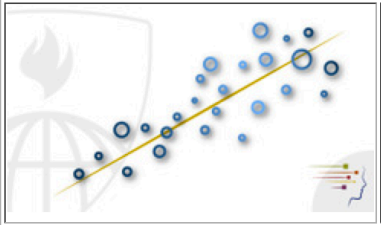
Notice, how the Controller did not have to **require** Coursera…

courses/index.html.erb



What Does It Look Like?

Searching for - jhu

Image	Name	Description
	Introduction to Genomic Technologies	The basic biology of modern genomics and the experimental tools used for measurement. This is the first course in the Genomic Big Data Science Specialization.
	Regression Models	Learn how to use regression models, the most important statistical analysis tool in the data scientist's toolkit. This is the seventh course in the Johns Hopkins Data Science Specialization.
	Statistics for Genomic Data	An introduction to the statistics behind the most popular genomic data science projects. This is the sixth course in

Summary

- By convention, Controller names are **plural** while Model names are **singular**
- Files under `app` directory are **automatically required**

What's Next?

- Making minor, but useful, improvements to our app

In this lecture, we will discuss…

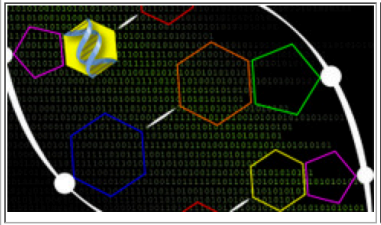
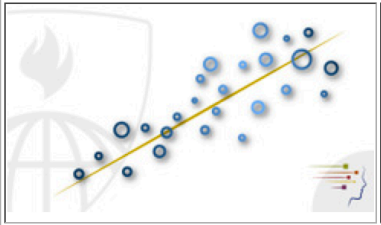
- ✧ Adding **basic styling** to our view
- ✧ Making the app **more dynamic** with a request parameter
- ✧ **Routing** with a root path

Layout

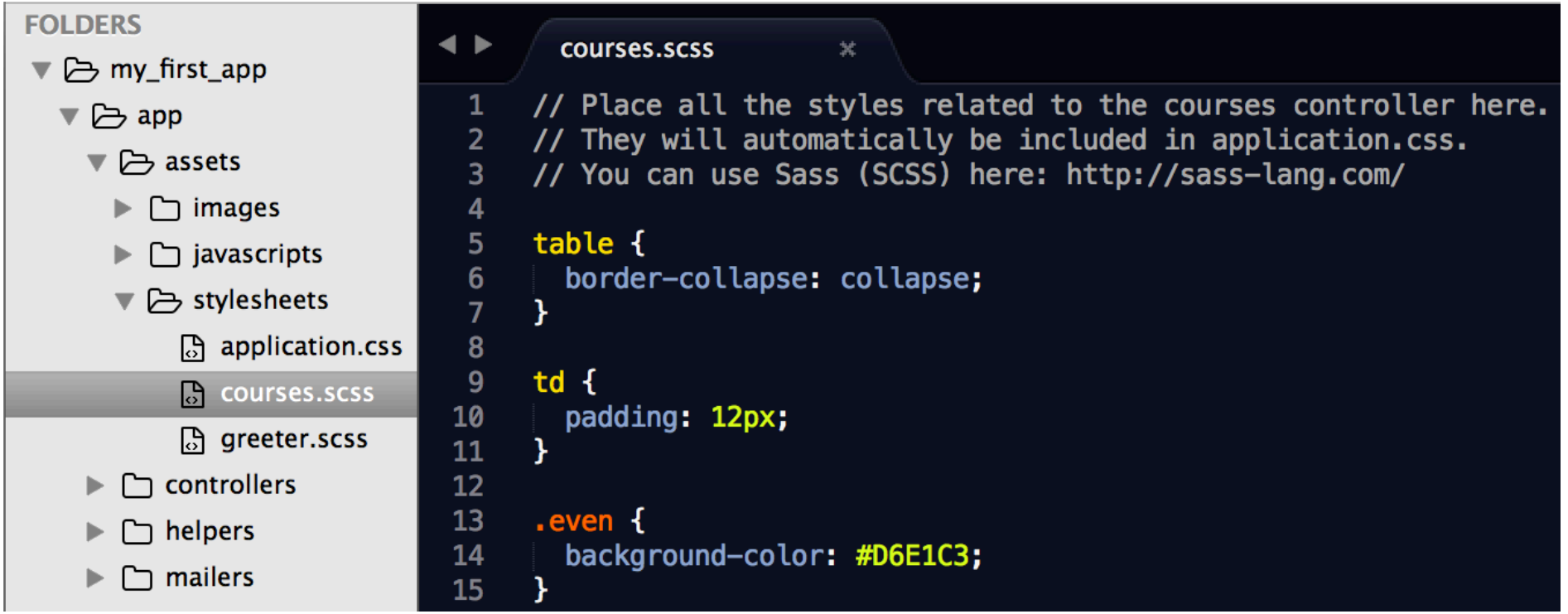
- `views/layout/application.html.erb` serves as view's container (unless overriden)

What Does It Look Like Now?

Searching for - jhu

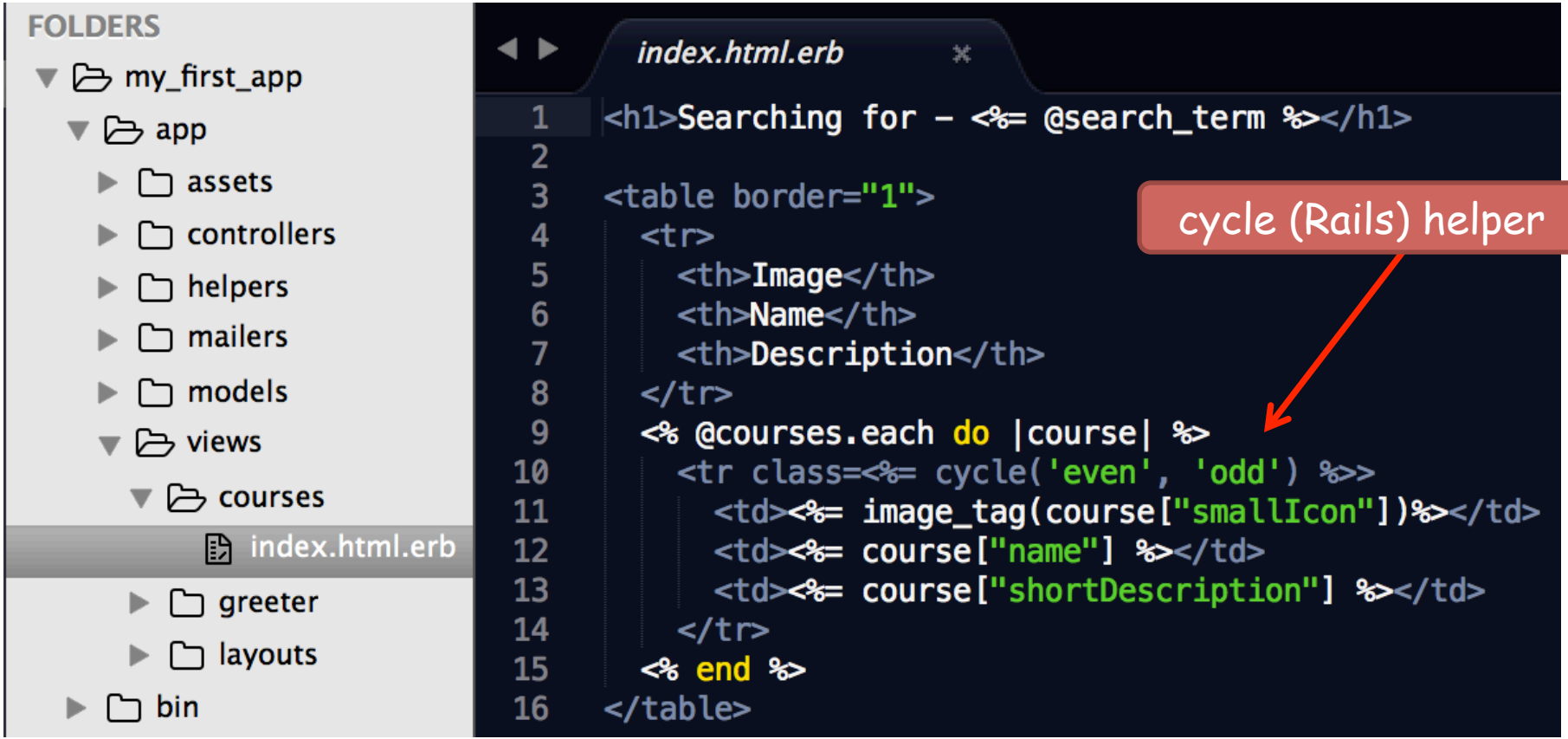
Image	Name	Description
	Introduction to Genomic Technologies	The basic biology of modern genomics and the experimental tools used for measurement. This is the first course in the Genomic Big Data Science Specialization.
	Regression Models	Learn how to use regression models, the most important statistical analysis tool in the data scientist's toolkit. This is the seventh course in the Johns Hopkins Data Science Specialization.
	Statistics for Genomic Data	An introduction to the statistics behind the most popular genomic data science projects. This is the sixth course in

Adding Some CSS



If you are new to CSS, go to Course 4 - HTML, CSS and Javascript for Web Developers

Modify View to Include CSS Classes

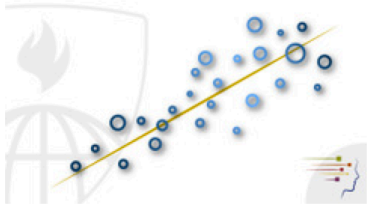


What Does It Look Like Now?

localhost:3000/courses/index

Search

Searching for - jhu

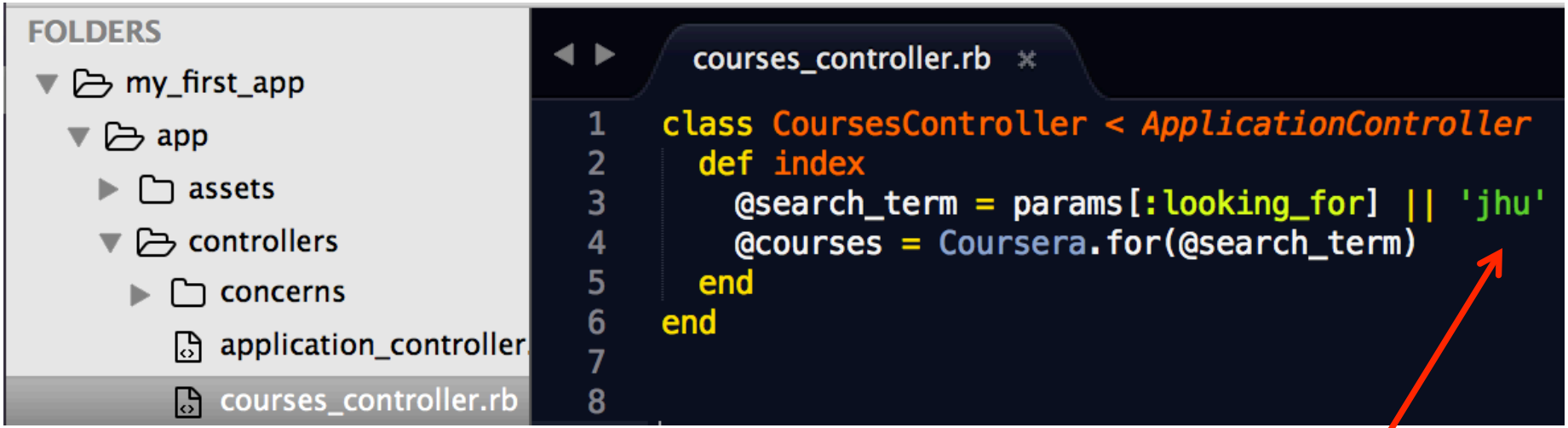
Image	Name	Description
	Introduction to Genomic Technologies	The basic biology of modern genomics and the experimental tools used for measurement. This is the first course in the Genomic Big Data Science Specialization.
	Regression Models	Learn how to use regression models, the most important statistical analysis tool in the data scientist's toolkit. This is the seventh course in the Johns Hopkins Data Science Specialization.
	Statistics for Genomic Data	An introduction to the statistics behind the most popular genomic data science projects. This is the sixth course

`params` helper

- It would be nice to **specify the search term**
- Use `params` “Hash” to **retrieve the value** (name of parameter becomes a symbol/key in the Hash)
- Returns `nil` if request parameter **not passed in** (standard Hash behavior)

params helper

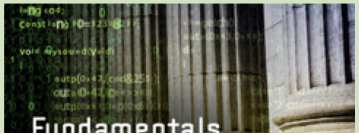
- No changes to the Model or the View, **only** to the Controller



Default to 'jhu' if request parameter not passed in

What Does It Look Like Now?

Searching for - python

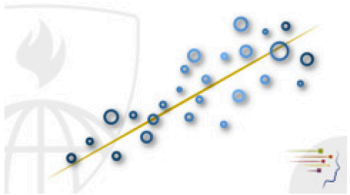
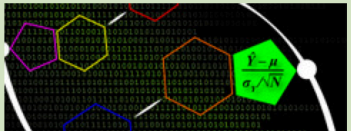
Image	Name	Description
	Algorithmic Thinking (Part 1)	Experienced Computer Scientists analyze and solve computational problems at a level of abstraction that is beyond that of any particular programming language. This two-part class is designed to train students in the mathematical concepts and process of "Algorithmic Thinking", allowing them to build simpler, more efficient solutions to computational problems.
	Deciphering Molecular Evolution (Bioinformatics IV)	In this course, we will see how evolutionary trees resolve quandaries from finding the origin of a deadly virus to locating the birthplace of modern humans. We will then use methods from computational proteomics to test whether we can reconstruct Tyrannosaurus rex proteins and prove that birds evolved from dinosaurs.
	The Fundamentals of Computing Specialization Capstone	The "Fundamentals of Computing" series concludes with a two-week course that reviews the series material and whose primary assessment is a cumulative "capstone" exam.

And Without A Request Parameter?

localhost:3000/courses/index

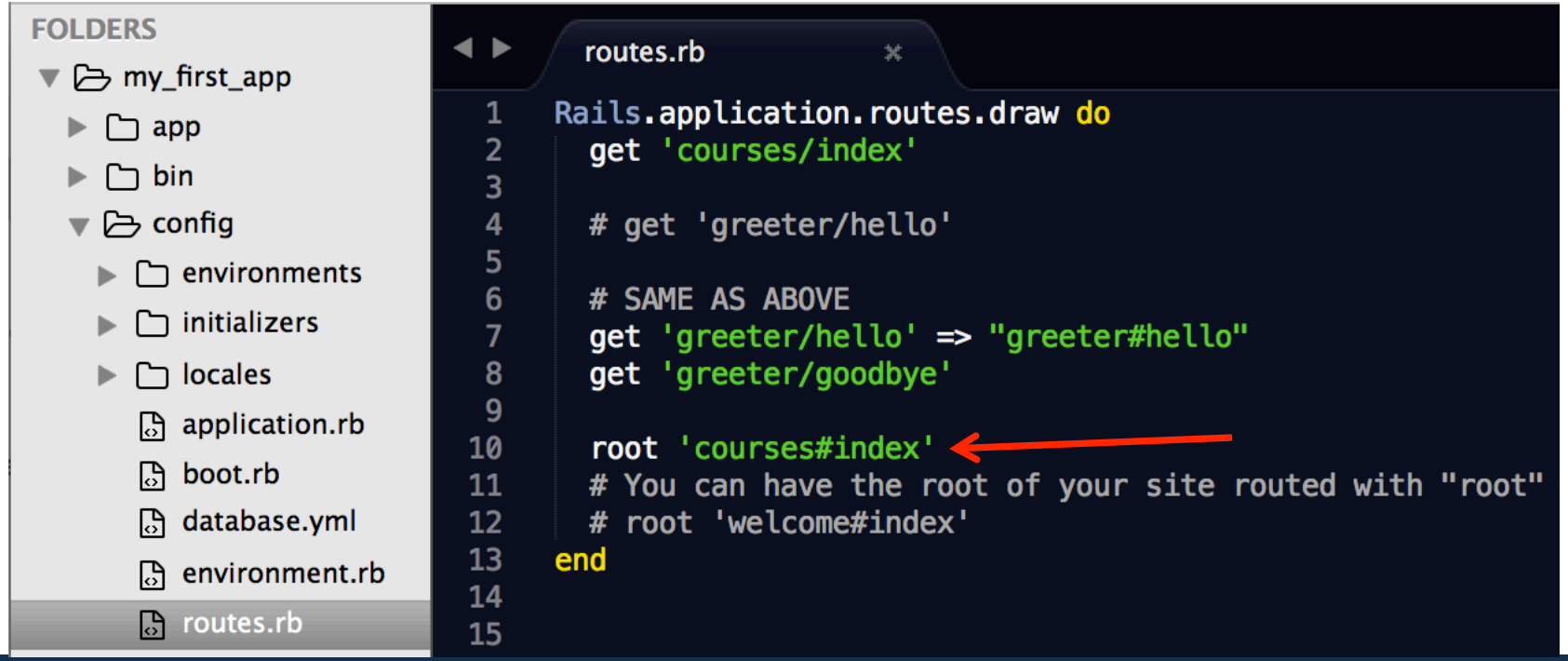
Search

Searching for - jhu

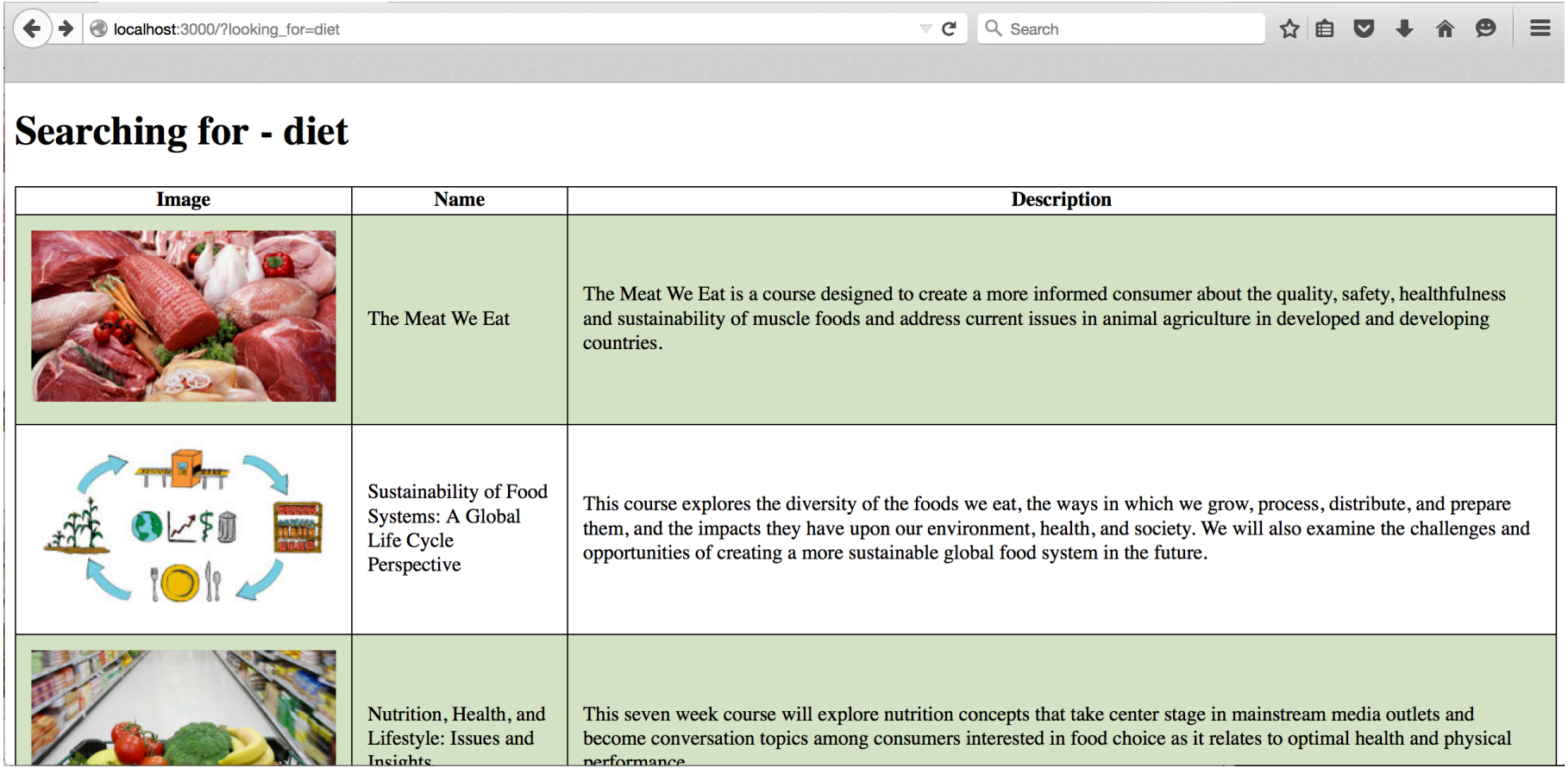
Image	Name	Description
	Introduction to Genomic Technologies	The basic biology of modern genomics and the experimental tools used for measurement. This is the first course in the Genomic Big Data Science Specialization.
	Regression Models	Learn how to use regression models, the most important statistical analysis tool in the data scientist's toolkit. This is the seventh course in the Johns Hopkins Data Science Specialization.
	Statistics for Genomic Data Science	An introduction to the statistics behind the most popular genomic data science projects. This is the sixth course in the Genomic Big Data Science Specialization from Johns Hopkins University.

One Final Twist: Root Path

- What if we wanted the root path go to the `index` action? Just modify `routes.rb`



One Final Twist – Root Path



Summary

- ✧ Minor CSS changes can **dramatically** enhance the app
- ✧ `params` helper **parses** request parameters
- ✧ Easy to change the root path by **tweaking** `routes.rb`

What's Next?

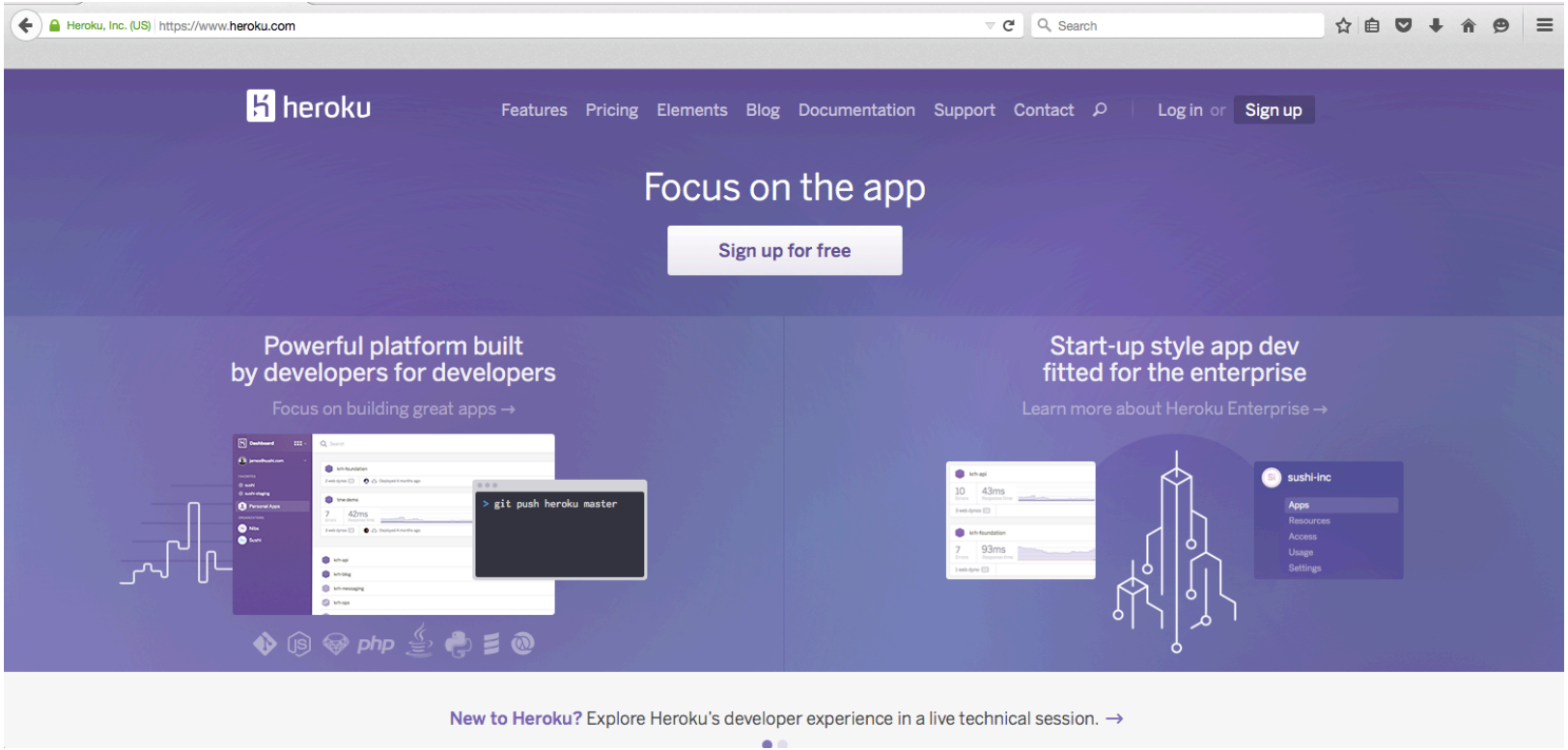
- ✧ Deploying to Heroku

In this lecture, we will discuss…

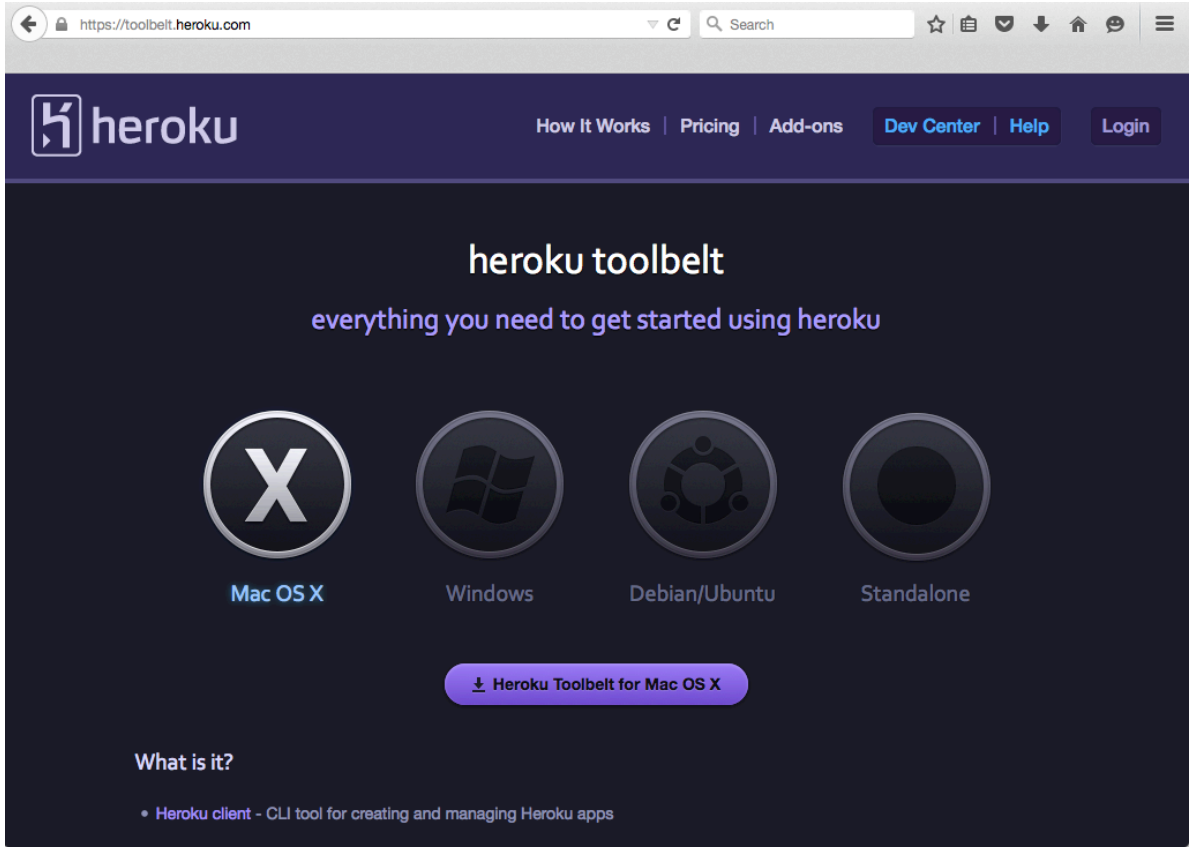
- Deploying to Heroku

Deploying to Heroku

- Sign up for new account at heroku.com

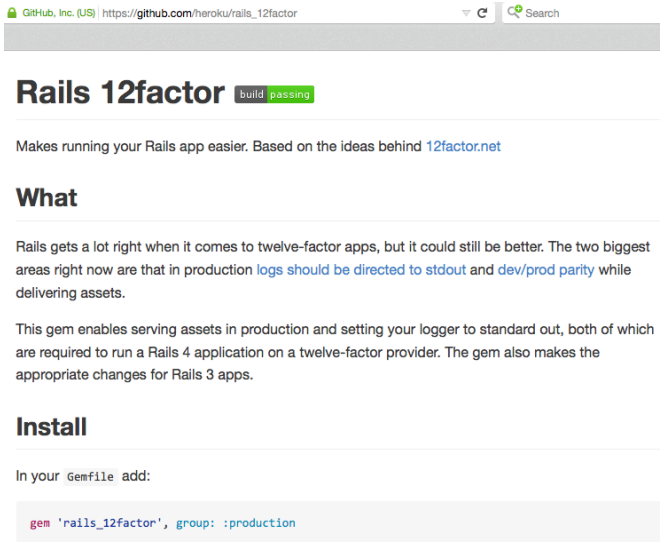


Heroku Toolbelt



Final Steps

- Heroku uses Postgres and recommends `rails_12factor` gem
- Put `sqlite` gem into development group and “heroku” gems in production



Final Steps

Gemfile

```
source 'https://rubygems.org'

gem 'rails', '4.2.3'
gem 'sqlite3', group: :development
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

group :development, :test do ...
end

group :production do
  gem 'pg'
  gem 'rails_12factor'
end

gem 'httparty', '0.13.5'
```

After adding the gems, run bundle and commit changes to repo!

Final Steps

```
~/my_first_app$ heroku login
Enter your Heroku credentials.
Email: kalmanh@gmail.com
Password (typing will be hidden):
Authentication successful.
~/my_first_app$ heroku create search-coursera-jhu
Creating search-coursera-jhu... done, stack is cedar-14
https://search-coursera-jhu.herokuapp.com/ | https://git.heroku.com/search-coursera-jhu.git
Git remote heroku added
~/my_first_app$ git remote -v
heroku  https://git.heroku.com/search-coursera-jhu.git (fetch)
heroku  https://git.heroku.com/search-coursera-jhu.git (push)
origin  https://github.com/jhu-ep-coursera/fullstack-course1-module3.git (fetch)
origin  https://github.com/jhu-ep-coursera/fullstack-course1-module3.git (push)
```

Final Steps

```
~/my_first_app$ git push heroku master
Counting objects: 159, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (144/144), done.
Writing objects: 100% (159/159), 155.86 KiB | 0 bytes/s, done.
Total 159 (delta 29), 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 1.9.7
remote:        Running: bundle install --without development:test --path vendor/bundle --binstubs vendor/bundle/bin -j4 --deployment
remote:        Fetching gem metadata from https://rubygems.org/.............
remote:        Fetching version metadata from https://rubygems.org/...
remote:        Fetching dependency metadata from https://rubygems.org/..
remote:        Rubygems 2.0.14 is not threadsafe, so your gems must be installed one at a time. Upgrade to Rubygems 2.1.0 or higher to enable parallel gem installation.
```

Make sure that the top level directory of your git repo contains `app` directory and other Rails files and directories!

Final Steps

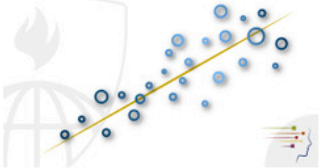
```
remote: ###### WARNING:
remote:        You have not declared a Ruby version in your Gemfile.
remote:        To set your Ruby version add this line to your Gemfile:
remote:        ruby '2.0.0'
remote:        # See https://devcenter.heroku.com/articles/ruby-versions for more information.
remote:
remote: ###### WARNING:
remote:        No Procfile detected, using the default web server (webrick)
remote:        https://devcenter.heroku.com/articles/ruby-default-web-server
remote:
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://search-coursera-jhu.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/search-coursera-jhu.git
 * [new branch]      master -> master
~/my_first_app$ heroku open
Opening search-coursera-jhu... done
```

Final Steps

https://search-coursera-jhu.herokuapp.com

Search

Searching for - jhu

Image	Name	Description
	Introduction to Genomic Technologies	The basic biology of modern genomics and the experimental tools used for measurement. This is the first course in the Genomic Big Data Science Specialization.
	Regression Models	Learn how to use regression models, the most important statistical analysis tool in the data scientist's toolkit. This is the seventh course in the Johns Hopkins Data Science Specialization.
	Statistics for Genomic Data Science	An introduction to the statistics behind the most popular genomic data science projects. This is the sixth course in the Genomic Big Data Science Specialization from Johns Hopkins University.

Final Steps

https://search-coursera-jhu.herokuapp.com/?looking_for=diet

Search

Searching for - diet

Image	Name	Description
	The Meat We Eat	The Meat We Eat is a course designed to create a more informed consumer about the quality, safety, healthfulness and sustainability of muscle foods and address current issues in animal agriculture in developed and developing countries.
	Sustainability of Food Systems: A Global Life Cycle Perspective	This course explores the diversity of the foods we eat, the ways in which we grow, process, distribute, and prepare them, and the impacts they have upon our environment, health, and society. We will also examine the challenges and opportunities of creating a more sustainable global food system in the future.
	Nutrition, Health, and Lifestyle: Issues and Insights	This seven week course will explore nutrition concepts that take center stage in mainstream media outlets and become conversation topics among consumers interested in food choice as it relates to optimal health and physical performance.

Summary

- ✧ Sign up for a Heroku account and download the toolbelt
- ✧ Don’t forget to change the `Gemfile`

What’s Next?

- ✧ Blackbox testing with RSpec and Capybara