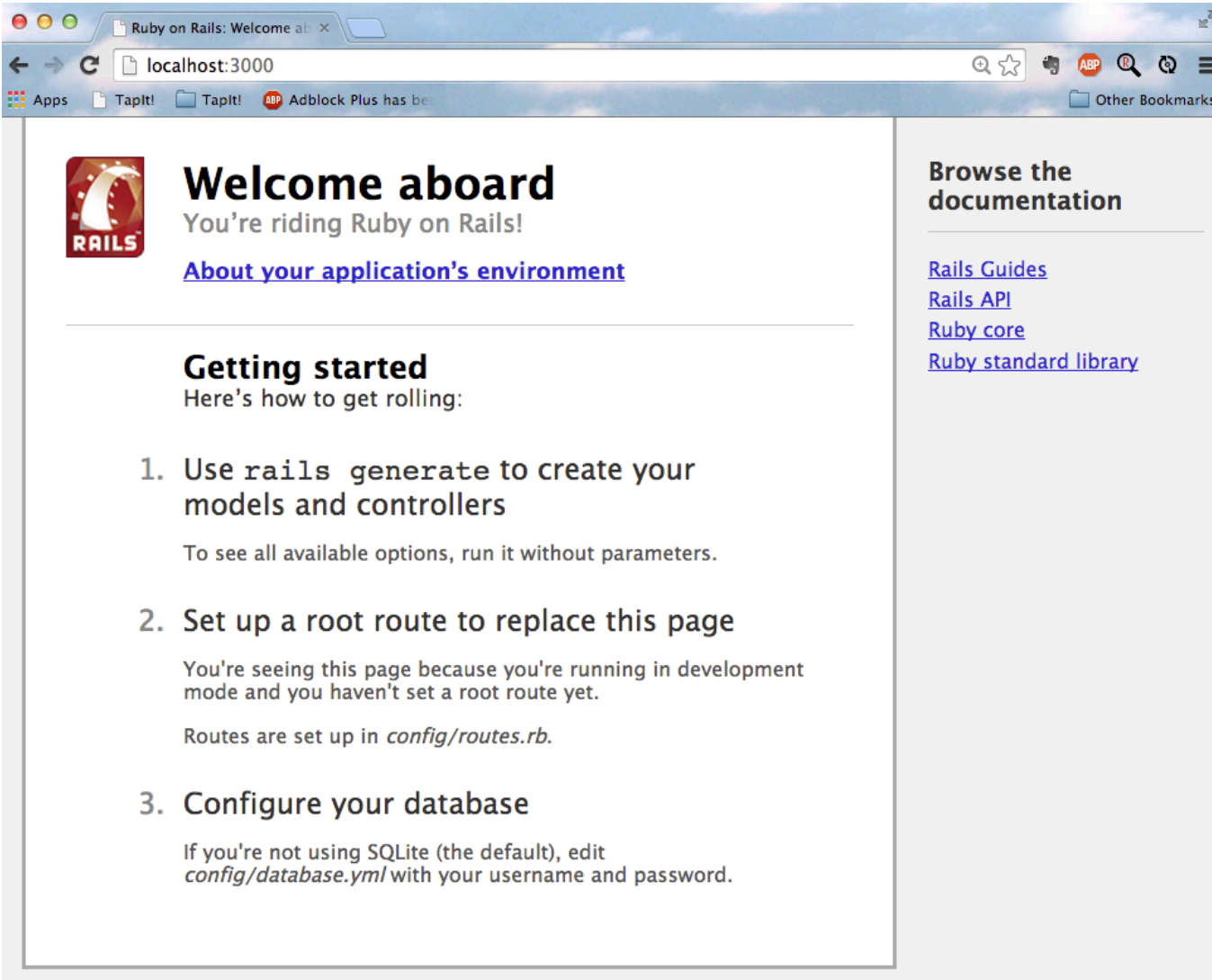




Intro to Ruby on Rails

Part II


Running the app (Continued)



Ruby on Rails: Welcome aboard x

localhost:3000

Apps Tapit! Tapit! Adblock Plus has been installed Other Bookmarks



Welcome aboard

You're riding Ruby on Rails!

[About your application's environment](#)

Getting started

Here's how to get rolling:

1. Use `rails generate` to create your models and controllers
To see all available options, run it without parameters.
2. Set up a root route to replace this page
You're seeing this page because you're running in development mode and you haven't set a root route yet.
Routes are set up in `config/routes.rb`.
3. Configure your database
If you're not using SQLite (the default), edit `config/database.yml` with your username and password.

Browse the documentation

- [Rails Guides](#)
- [Rails API](#)
- [Ruby core](#)
- [Ruby standard library](#)

Directory structure convention

▼ my_first_app

► app

► bin

► config

► db

► lib

► log

► public

► test

► tmp

► vendor

.gitignore

config.ru

Gemfile

Gemfile.lock

Rakefile

README.rdoc

Controllers, Views, Models (and helpers)

Configuration files

Files related to your db and migration "scripts"

Third-party code or your code that isn't directly
a model view or controller (or helper)

Log files

Static files

Unit, functional and integration tests

Dependencies managed by Bundler

public/hello_static.html

- **Rails 4+ UPDATE:** index.html is no longer in the public directory
- Server looks into `public` directory before looking anywhere else
- So... if we want to add a completely static web page (or any file) to our application - we can add it under `public` directory

Adding hello_static.html to public dir



No need to restart the server - just refresh your browser



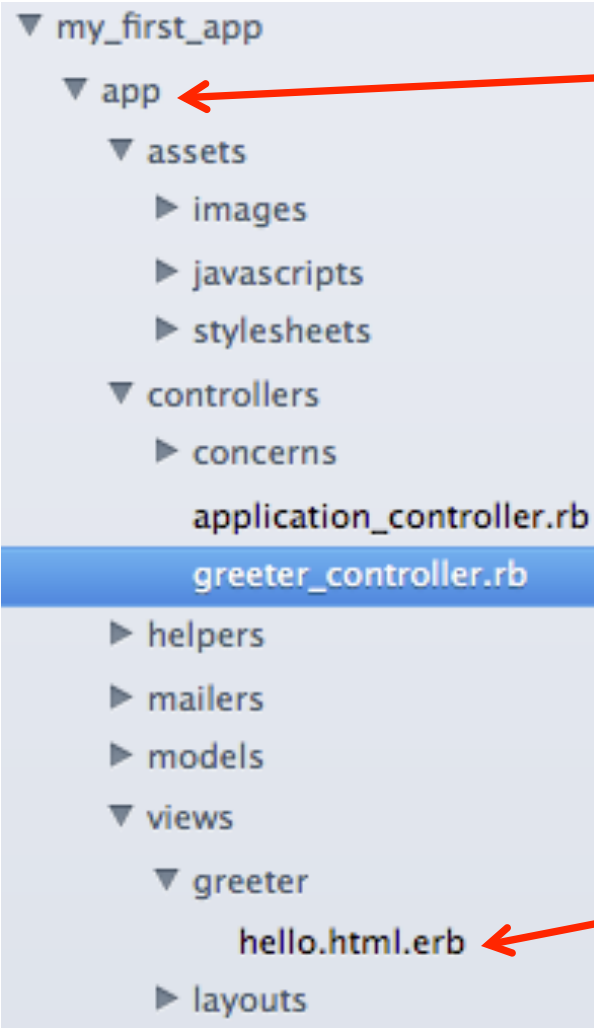
Generating a controller

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

Generating a controller

ruby	bash
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
create	test/helpers/greeter_helper_test.rb
invoke	assets
invoke	coffee
create	app/assets/javascripts/greeter.js.coffee
invoke	scss
create	app/assets/stylesheets/greeter.css.scss

Generating a controller



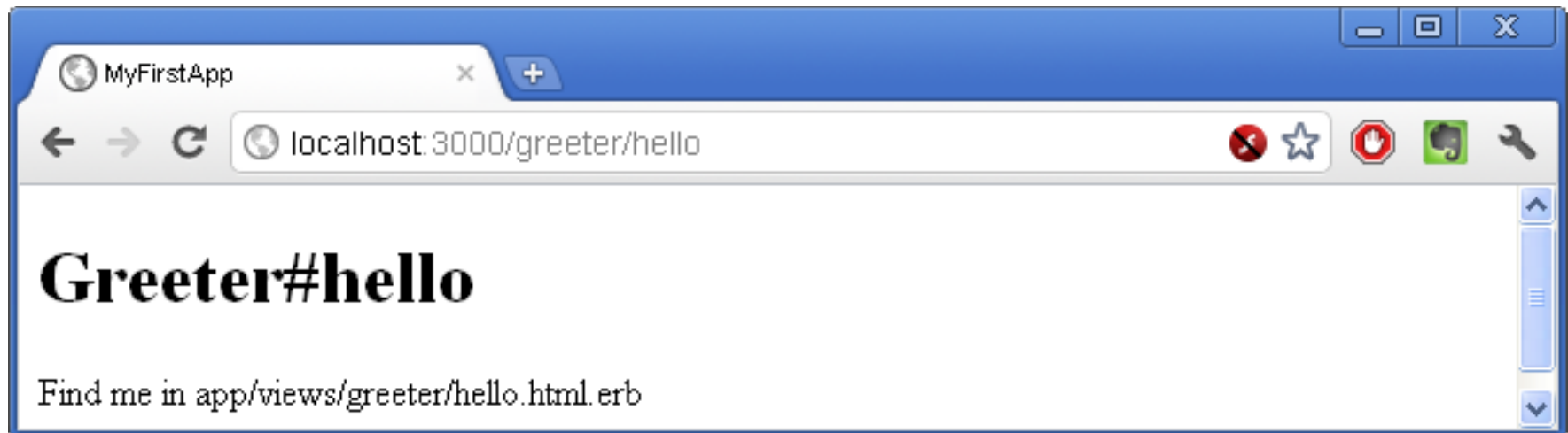
... app directory is where you will be spending most of your time ...

Controller

View

What does it look like? (View)

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



(No DOCTYPE, head or body elements...)

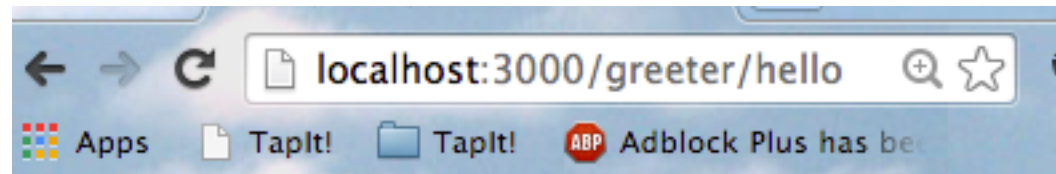
ERB (Embedded Ruby)

- 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 say JSP) that lets you embed Ruby into your html
- 2 tag patterns to learn:
 - `<% ...ruby code... %>` - evaluate Ruby code
 - `<%= ...ruby code... %>` - output evaluated Ruby code

New hello.html.erb

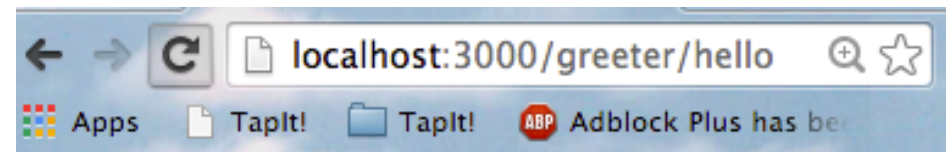
hello.html.erb

```
1 <% random_names = %w{Joe Alex Harry} %>
2 <h1>Greetings, <%= random_names.sample %></h1>
3 <p>Time now is <%= Time.now %></p>
```



Greetings, Alex

Time now is 2014-05-23 11:26:22 -0400



Greetings, Joe

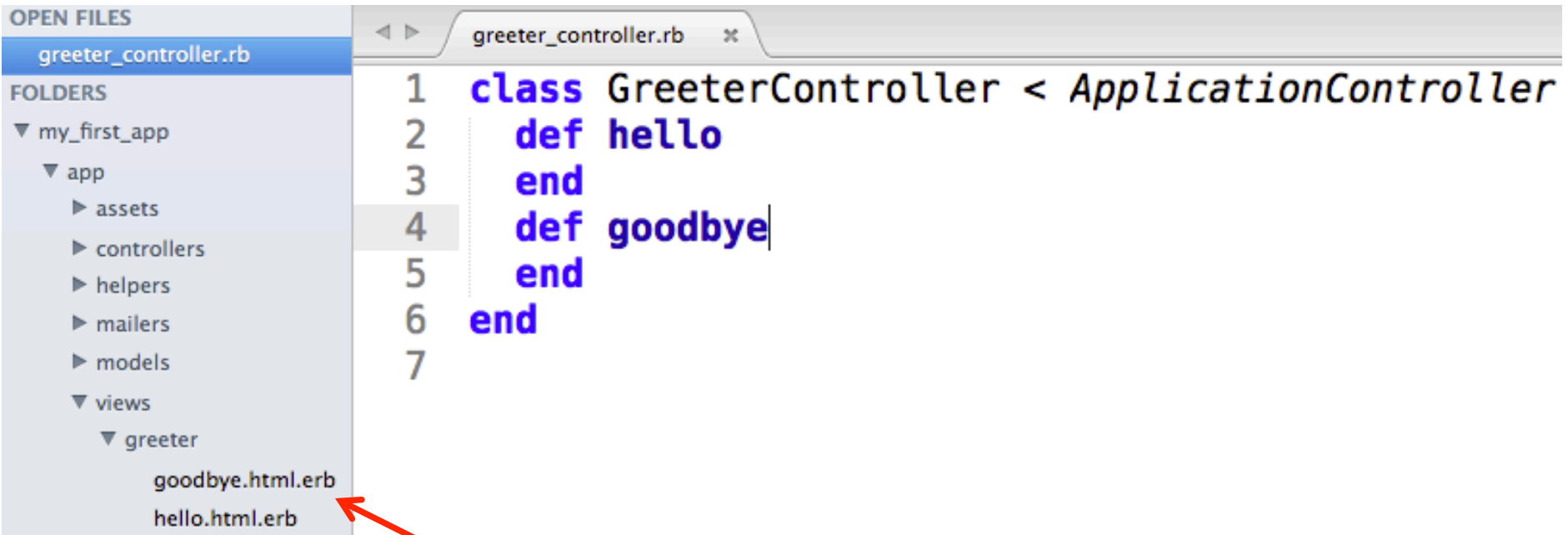
Time now is 2014-05-23 11:29:12 -0400

What does it look like? (Controller)

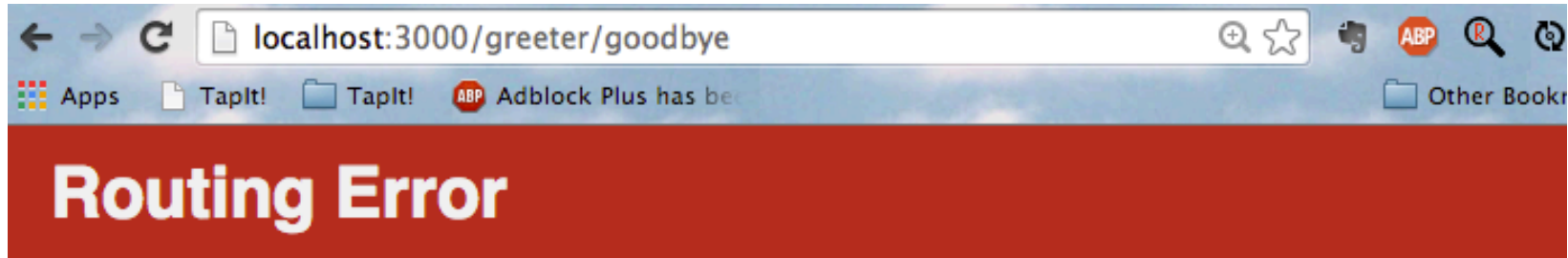
```
greeter_controller.rb
1 class GreeterController < ApplicationController
2   def hello
3   end
4 end
```

- `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?

Adding goodbye action



Adding Goodbye Action



No route matches [GET] "/greeter/goodbye"

Rails.root: /Users/hazink1/my_first_app

[Application Trace](#) | [Framework Trace](#) | [Full Trace](#)

Routes

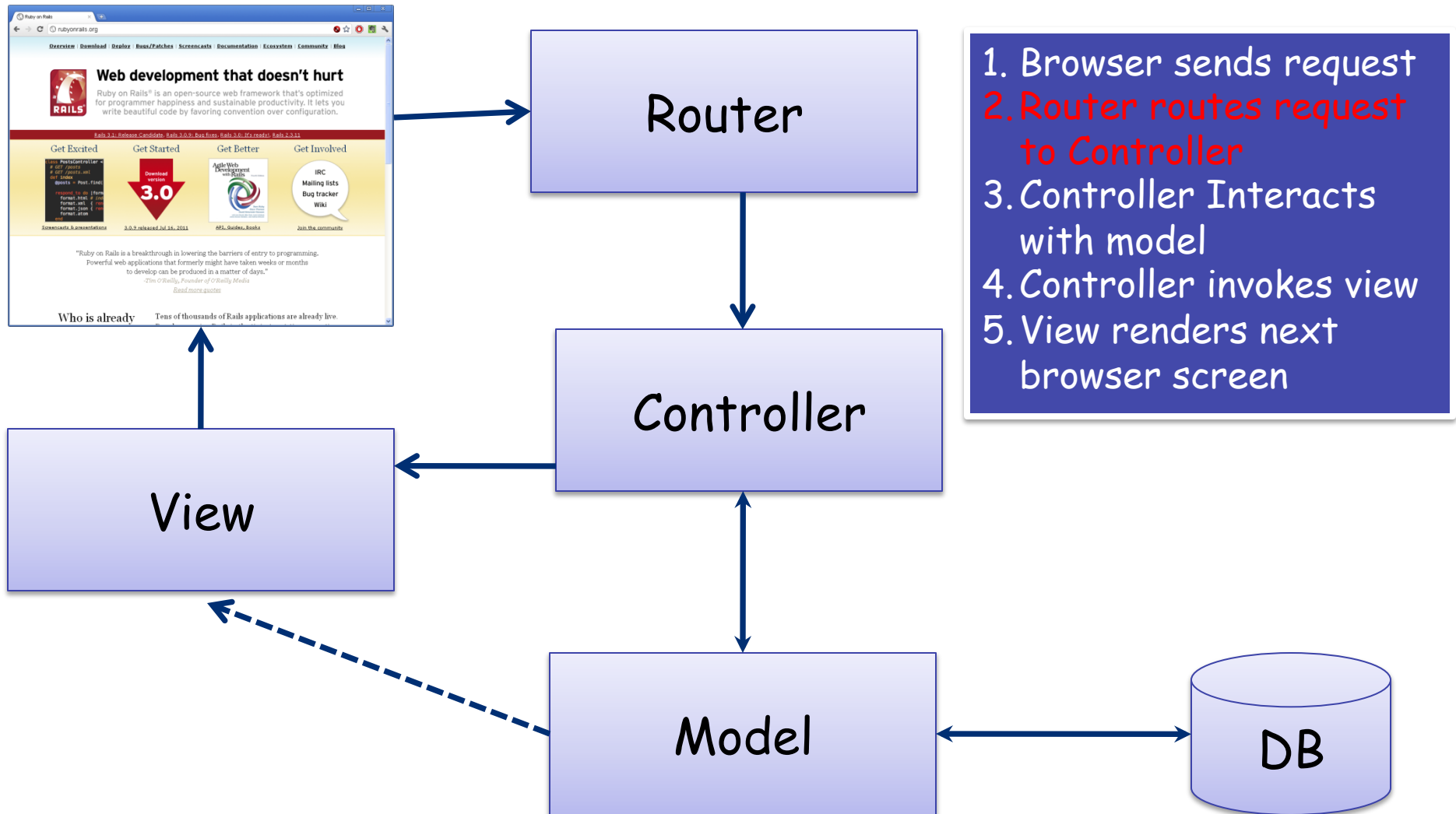
Routes match in priority from top to bottom

Helper	HTTP Verb	Path	Controller#Action
<u>Path / Url</u>		Path Match	
greeter_hello_path	GET	/greeter/hello(.:format)	greeter#hello

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 then?
- The route for `hello` action was automatically generated (during `$rails g controller`)

MVC(R) Cycle - Revised!



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

```
routes.rb
1 Rails.application.routes.draw do
2   get 'greeter/hello'
3
4   # The priority is based upon order of creation: first created -> highest priority.
5   # See how all your routes lay out with "rake routes".
6
7   # You can have the root of your site routed with "root"
8   # root 'welcome#index'
9
10  # ...|
11 end
```

- Let's add the route for the `goodbye` action

New config/routes.rb

The diagram illustrates the relationship between routes and controllers in a Rails application. It features a code editor window showing the contents of `routes.rb`, with two callout boxes: "Action" and "Controller".

Code Editor (routes.rb):

```
1 Rails.application.routes.draw do
2   # get 'greeter/hello'
3
4   # SAME AS ABOVE
5   get "greeter/hello" => "greeter#hello"
6   get "greeter/goodbye"
7
8   # The priority is based upon order of creation: first created -> highest priority.
9   # See how all your routes lay out with "rake routes".
10
11  # You can have the root of your site routed with "root"
12  # root 'welcome#index'
13
14  # ...
15 end
```

Callout Boxes:

- Action:** A blue box with a red arrow pointing to the `greeter#hello` part of the route definition on line 5.
- Controller:** A blue box with a red arrow pointing to the `greeter` part of the route definition on line 5.

Browser Screenshot:

A screenshot of a web browser showing the URL `localhost:3000/greeter/goodbye`. The browser's address bar displays the URL, and the page content shows the word `goodbye`. The browser's taskbar at the bottom includes icons for "Apps", "TapIt!", and "Adblock Plus has be".