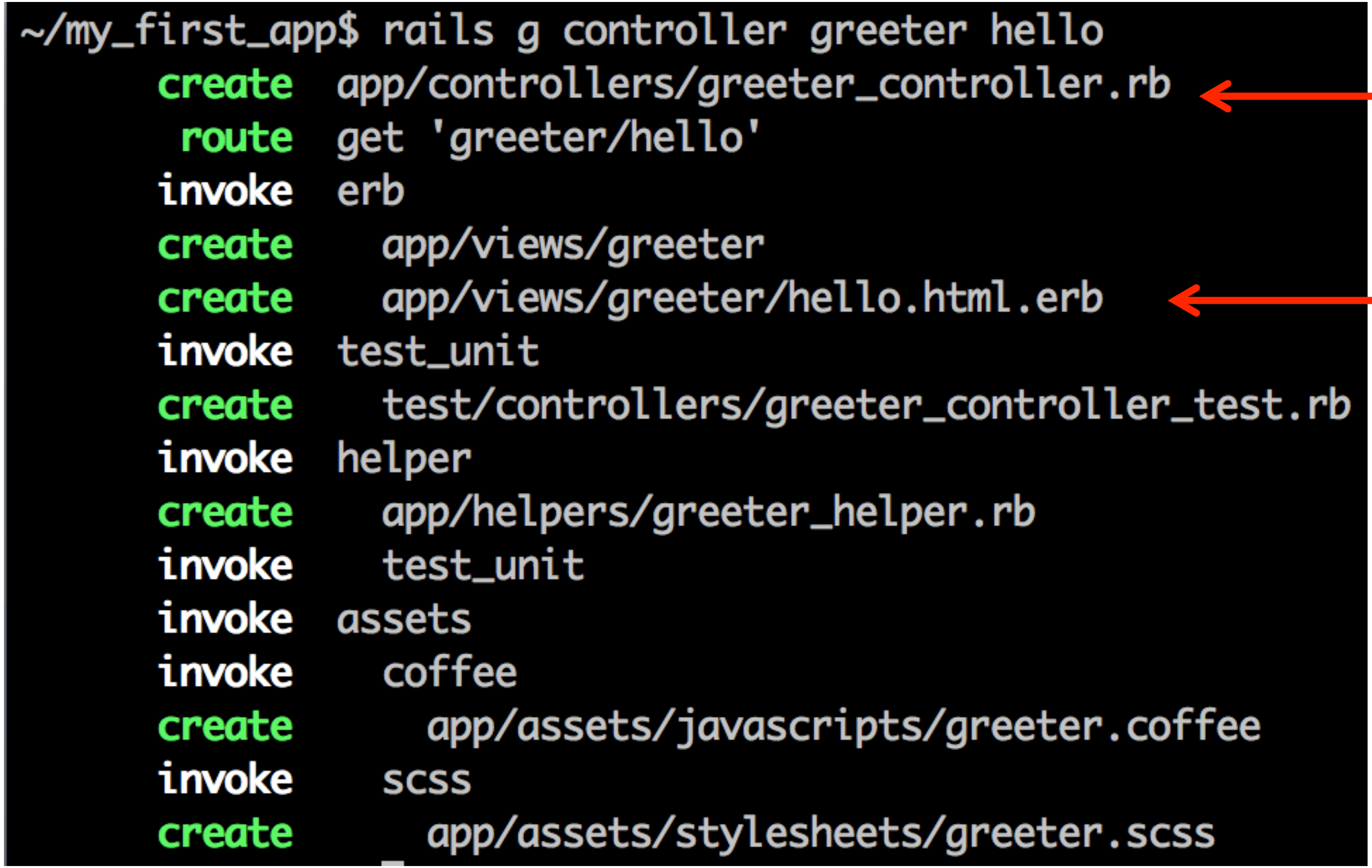
In this lecture, we will discuss…

- 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

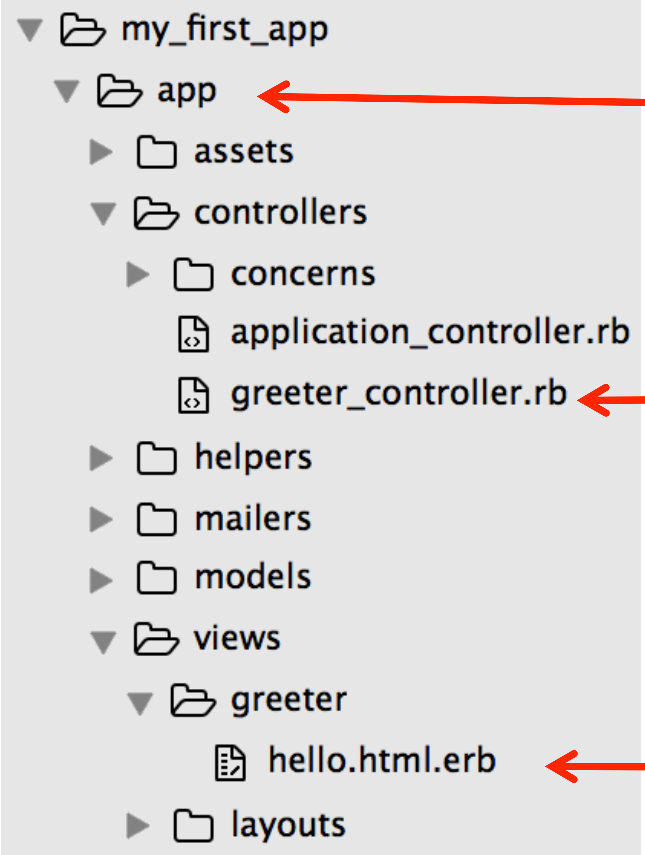


```
~/my_first_app$ rails g controller greeter hello
      create  app/controllers/greeter_controller.rb
       route  get 'greeter/hello'
      invoke  erb
      create    app/views/greeter
      create    app/views/greeter/hello.html.erb
      invoke  test_unit
      create    test/controllers/greeter_controller_test.rb
      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
```

Controller

View

Generating a Controller



```
▼ my_first_app
    ▼ app
        ▶ assets
        ▼ controllers
            ▶ concerns
              application_controller.rb
              greeter_controller.rb
        ▶ helpers
        ▶ mailers
        ▶ models
        ▼ views
            ▼ greeter
                  hello.html.erb
            ▶ layouts
```

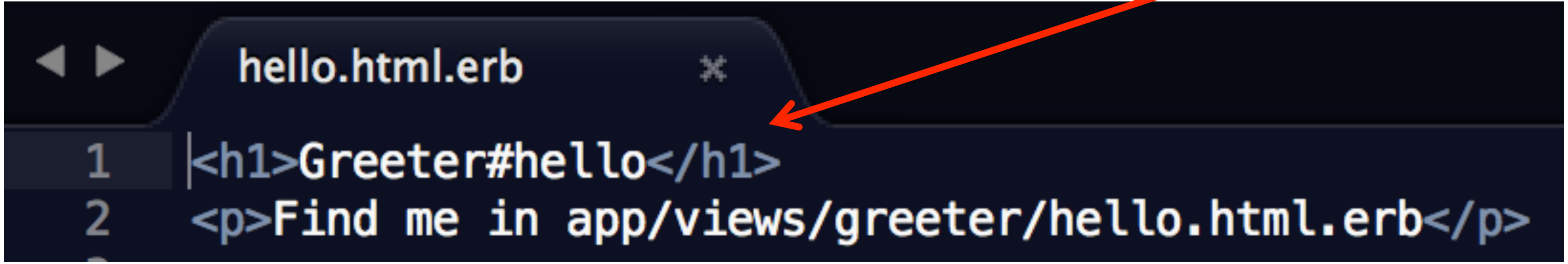
… app directory is where you will be spending most of your time …

Controller

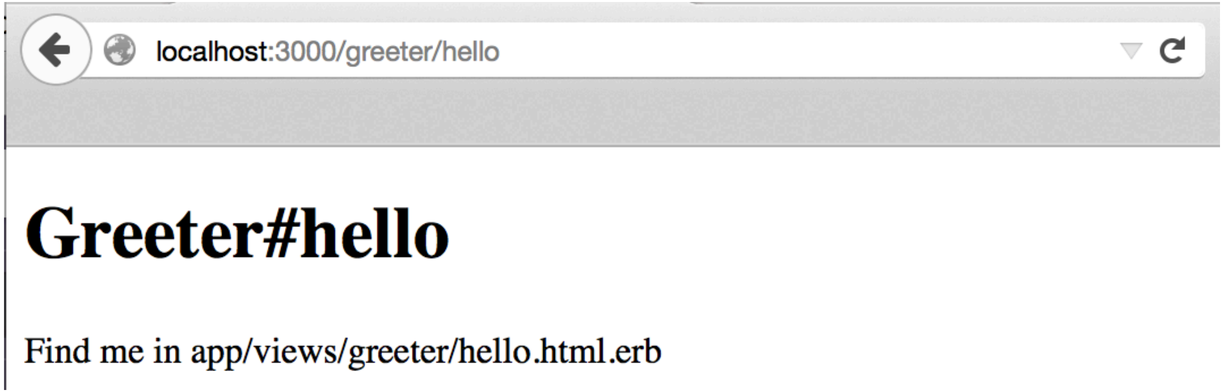
View

What Does It Look Like? (View)

No DOCTYPE, head or body elements…



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



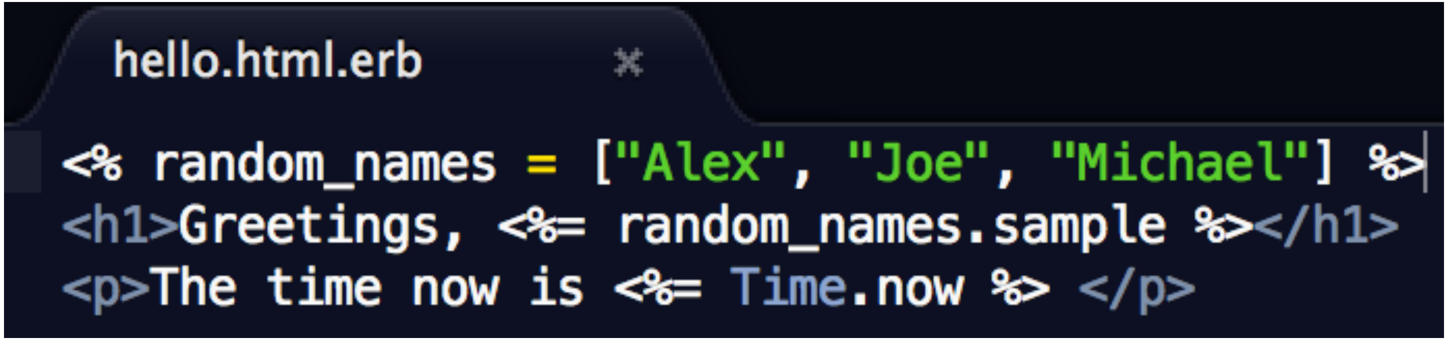
Greeter#hello

Find me in app/views/greeter/hello.html.erb

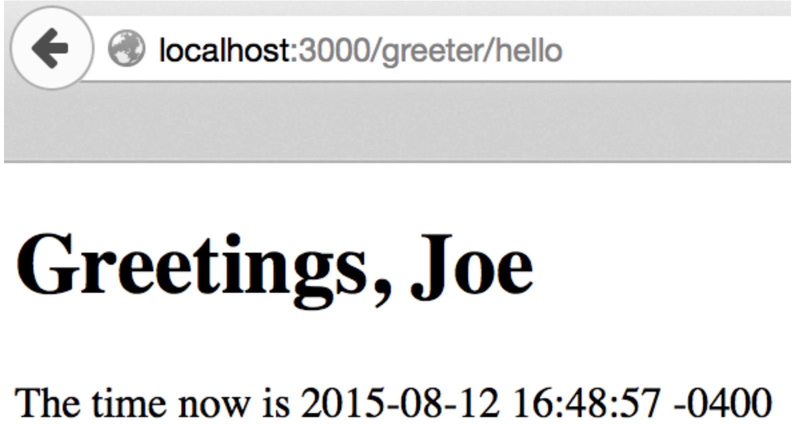
ERB (Embedded Ruby)

- 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

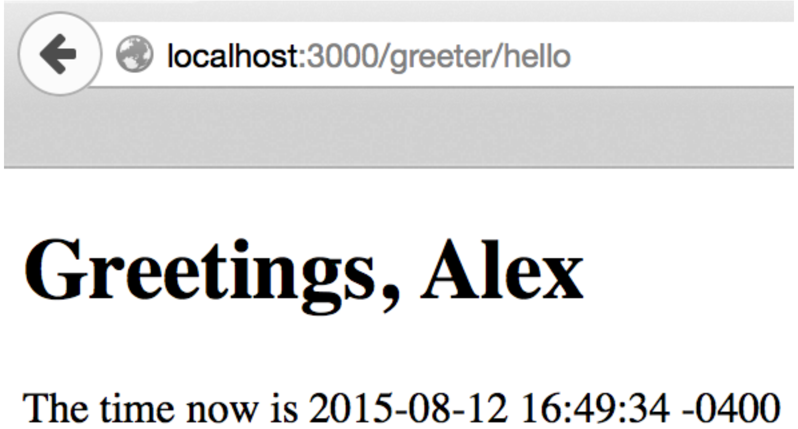


```
<% random_names = ["Alex", "Joe", "Michael"] %>
<h1>Greetings, <%= random_names.sample %></h1>
<p>The time now is <%= Time.now %> </p>
```



Greetings, Joe

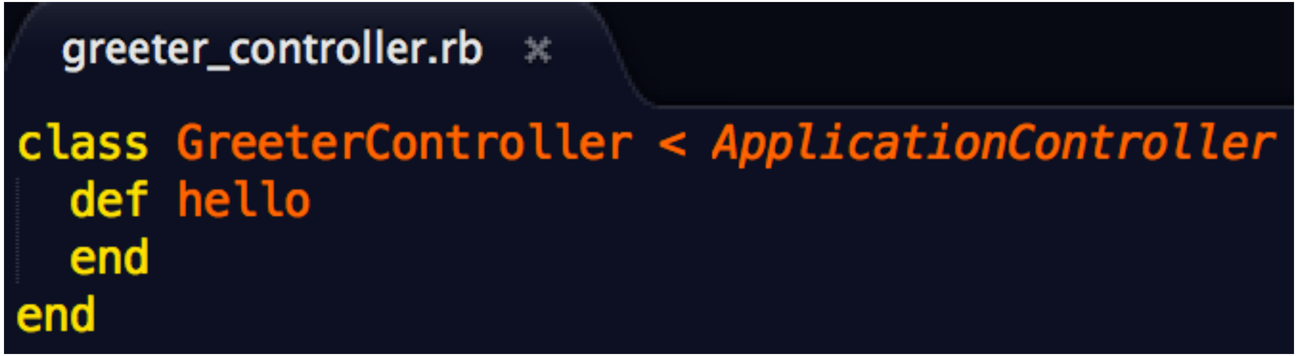
The time now is 2015-08-12 16:48:57 -0400



Greetings, Alex

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

What Does It Look Like? (Controller)

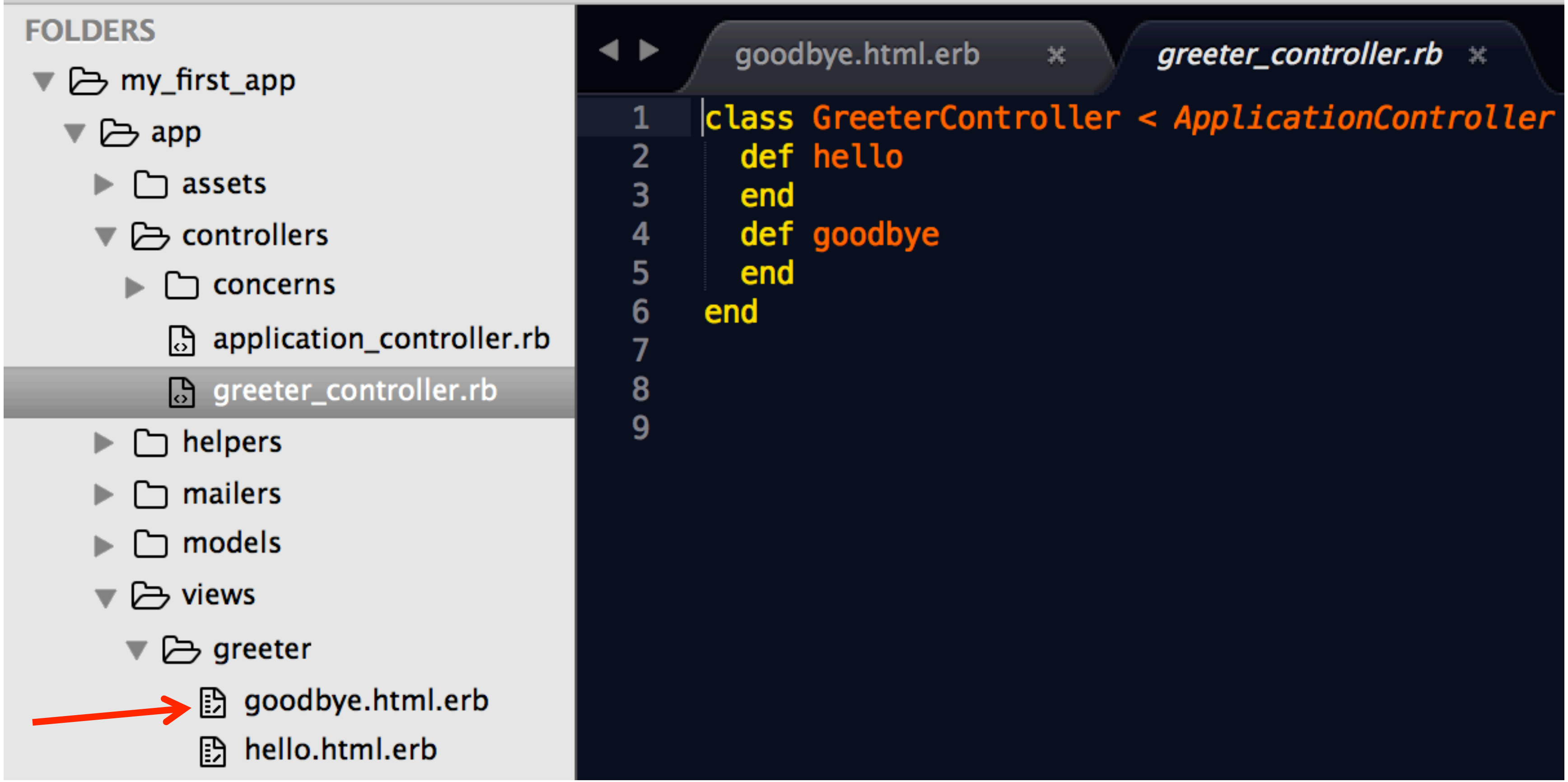


```ruby
class GreeterController < ApplicationController
  def hello
  end
end
```

- `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?

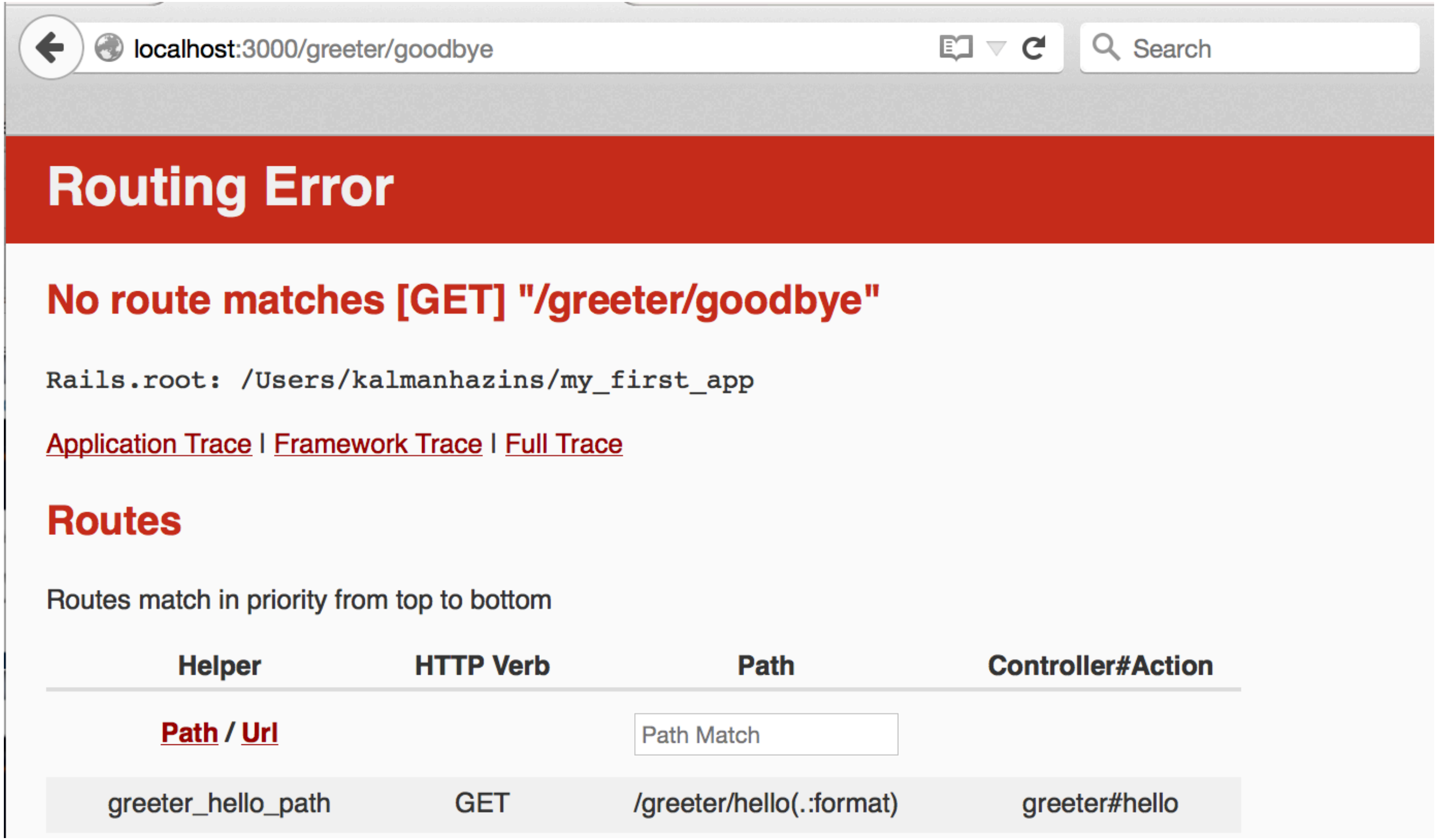
Should work… ?

Adding Goodbye Action



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

Adding Goodbye Action



Routing Error

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

```
Rails.root: /Users/kalmanhazins/my_first_app
```

Application Trace | Framework Trace | Full Trace

Routes

Routes match in priority from top to bottom

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

Summary

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

What's Next?

- Routes