RAILS

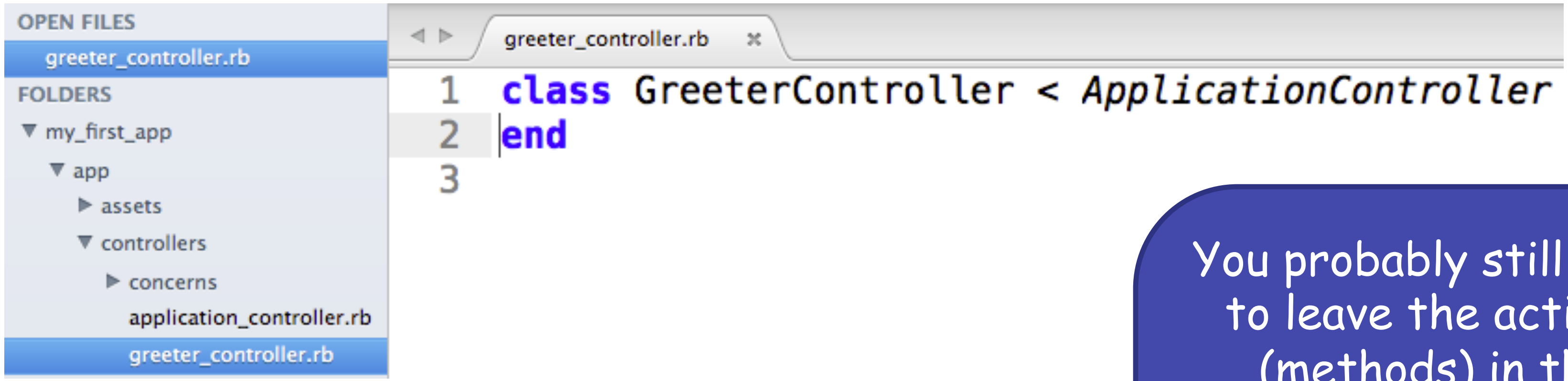
Introduction to Ruby on Rails

Part III

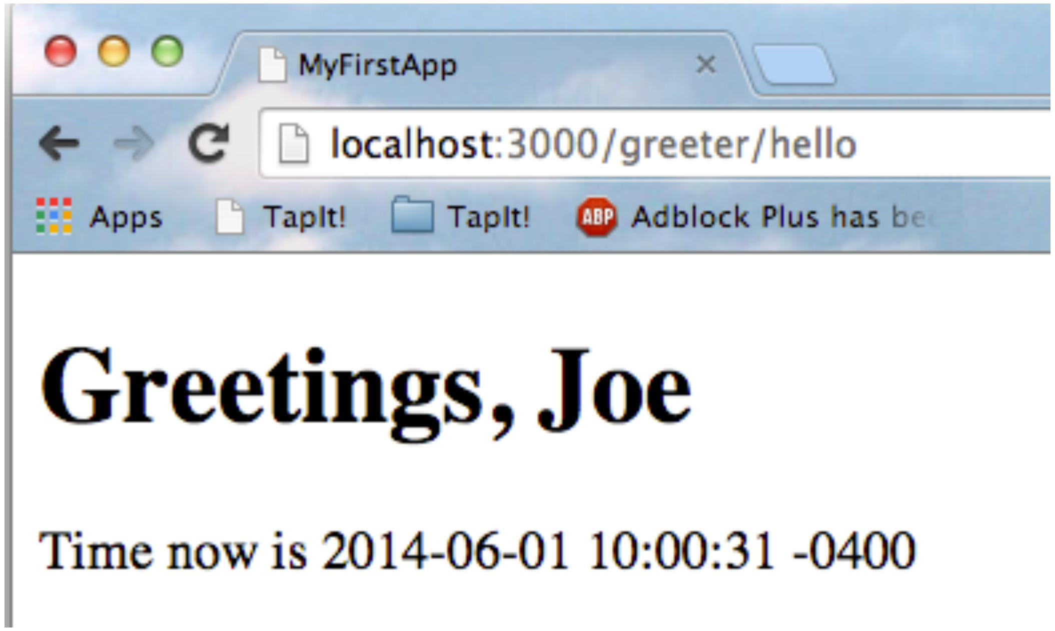
Action methods inside Controller

- If the action method is not really doing anything (i.e. 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 use a convention to find the correct template

Controller - new look



```ruby
class GreeterController < ApplicationController
end
```



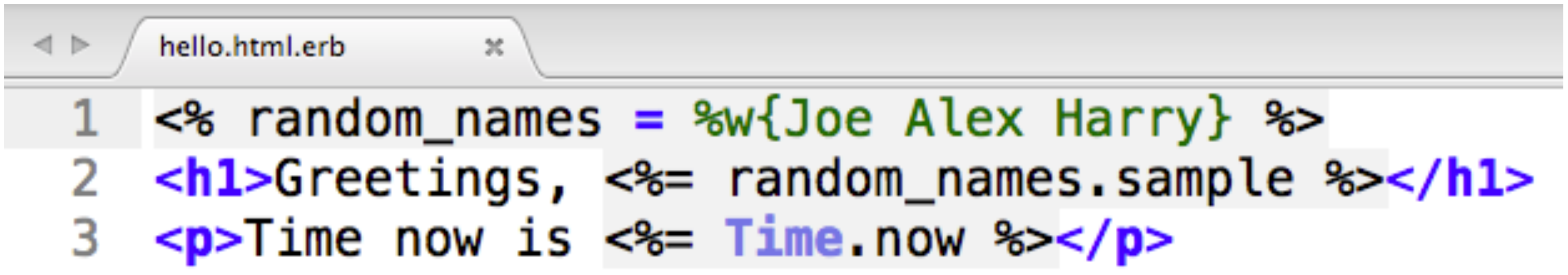
Greetings, Joe

Time now is 2014-06-01 10:00:31 -0400

You probably still want to leave the actions (methods) in the Controller (to make it easier on another person looking for them), but the point of this slide is that you don't *have* to have the methods if they are empty as long as the routes are properly defined and the views exist

Moving business logic out

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

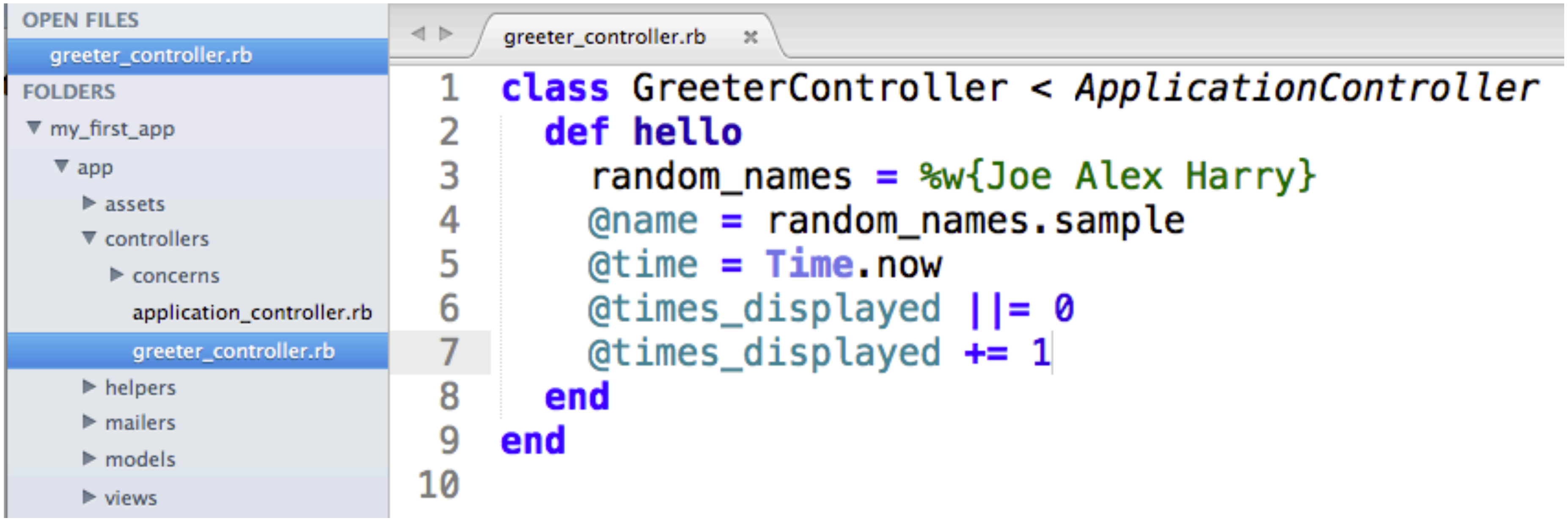


```
<% random_names = %w{Joe Alex Harry} %>
<h1>Greetings, <%= random_names.sample %></h1>
<p>Time now is <%= Time.now %></p>
```

- Let’s move some code to the controller!

Moving business logic out (Cont.)

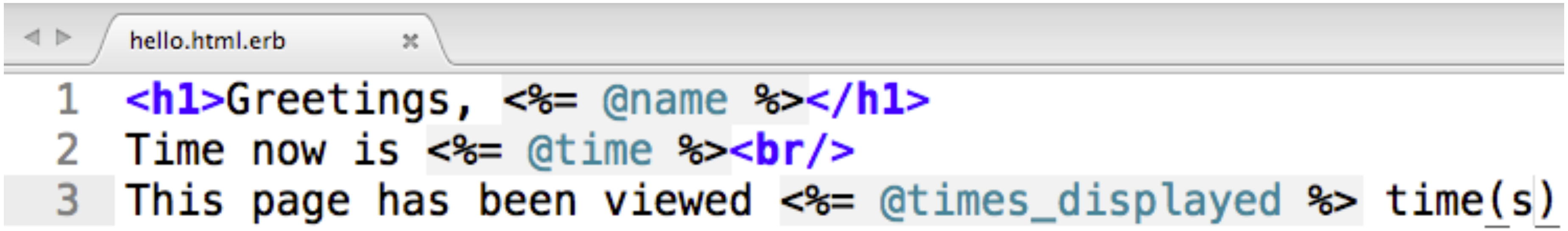
- Instance variables from the controller are available inside the view



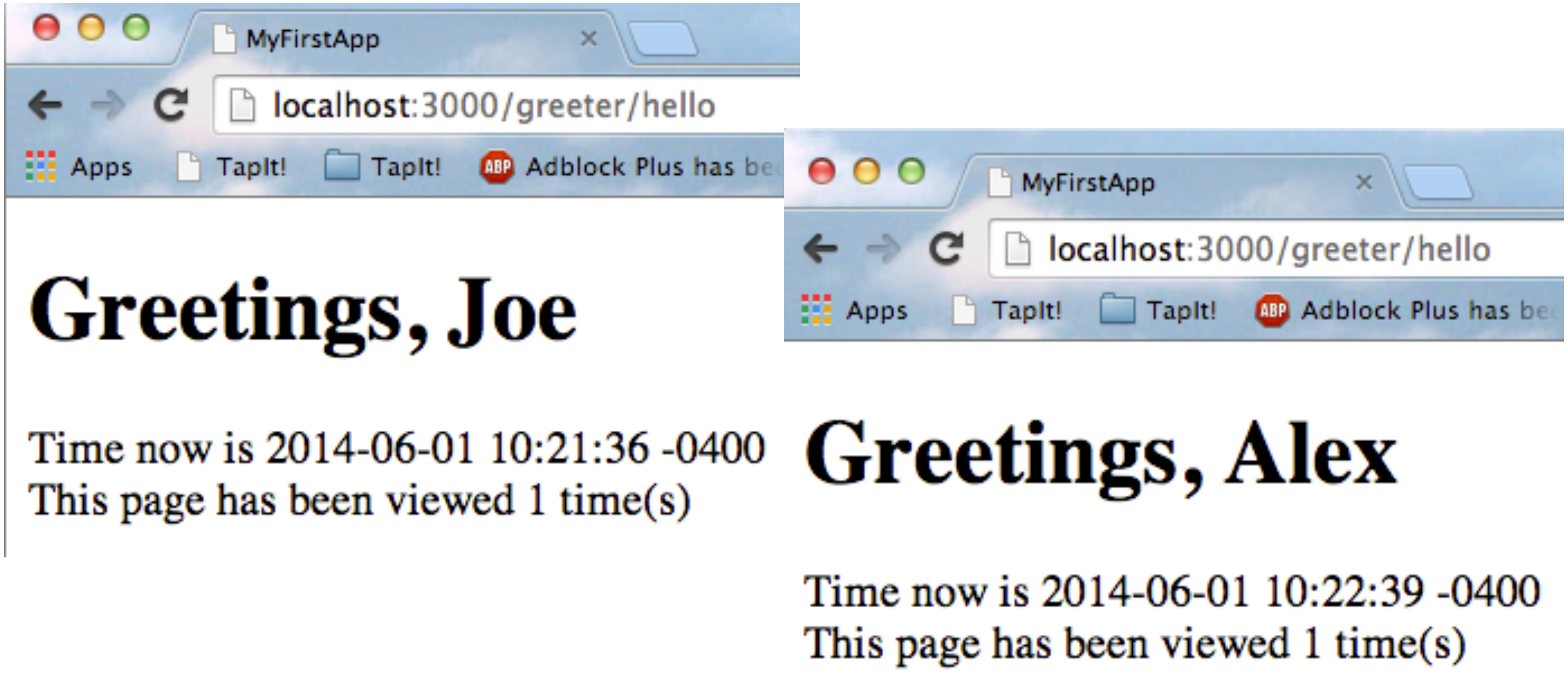
```ruby
class GreeterController < ApplicationController
  def hello
    random_names = %w{Joe Alex Harry}
    @name = random_names.sample
    @time = Time.now
    @times_displayed ||= 0
    @times_displayed += 1
  end
end
```

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

New views/greeter/hello.html.erb



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



Greetings, Joe

Time now is 2014-06-01 10:21:36 -0400
This page has been viewed 1 time(s)

Greetings, Alex

Time now is 2014-06-01 10:22:39 -0400
This page has been viewed 1 time(s)

Instance variables in Rails

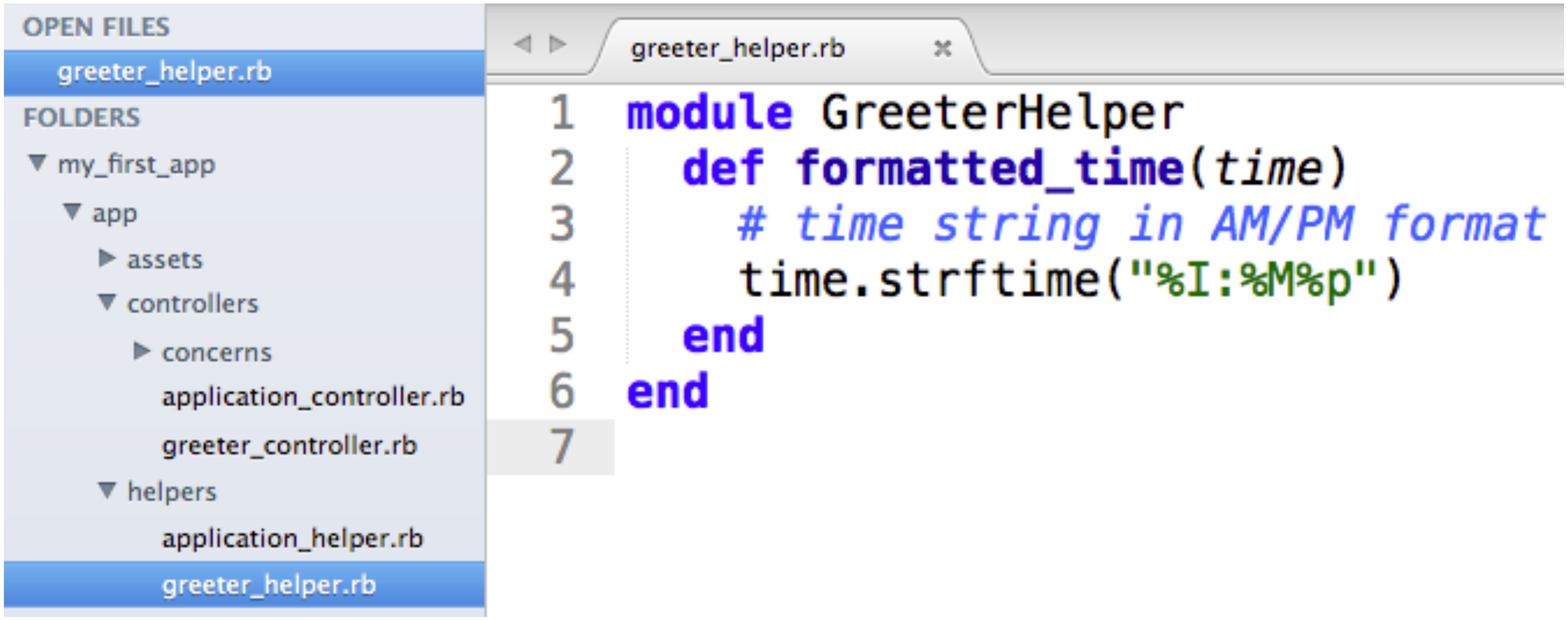
- Unlike Servlets/JSP – a new instance of the Controller is created **every time a request is made**
- Therefore, you can't "store" values in the Controller's instance variables in between requests
- Alternative?
  – Session
  – Database

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? Then, we can't reuse it. So… maybe the Controller? That doesn't seem correct either – since the Controller should be "view format" agnostic
- Helpers to the rescue!

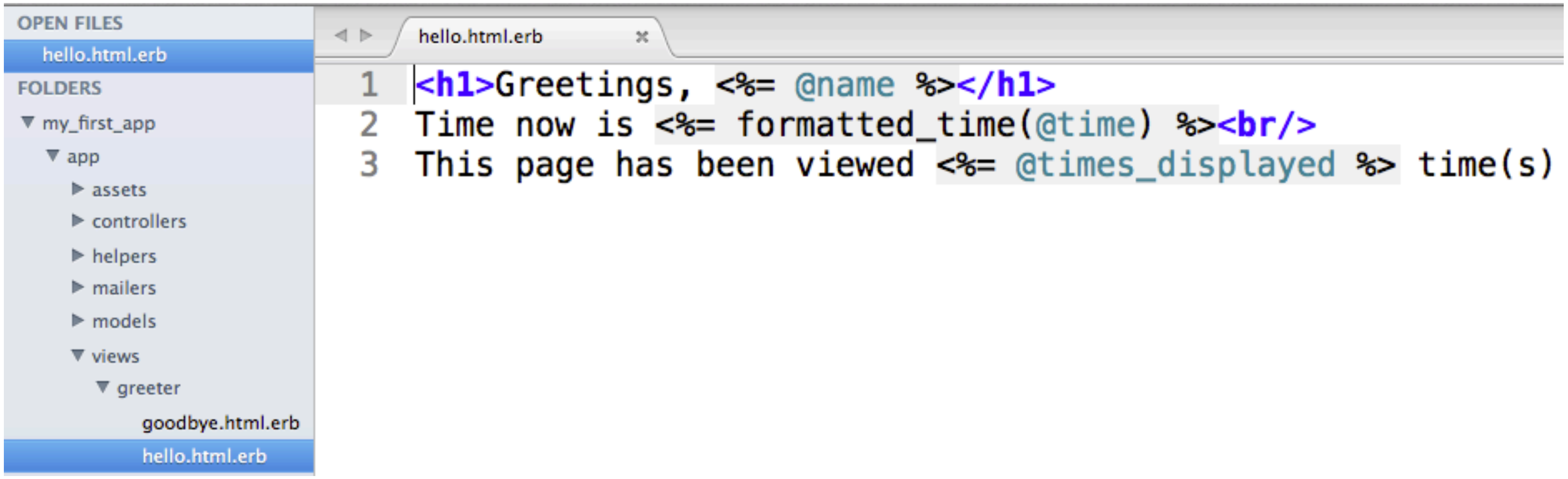
Helpers (Continued)

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

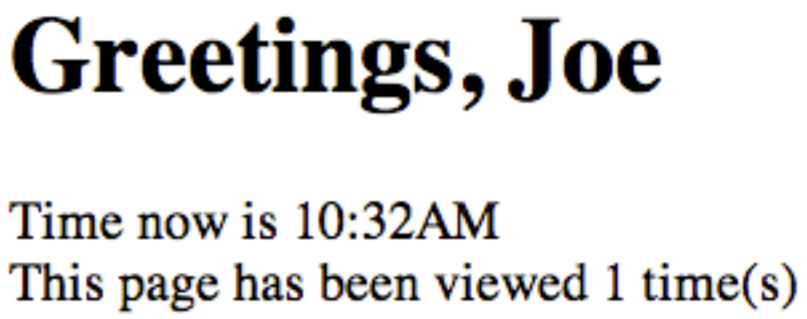


```ruby
module GreeterHelper
  def formatted_time(time)
    # time string in AM/PM format
    time.strftime("%I:%M%p")
  end
end
```

Helpers (Continued) – New View



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



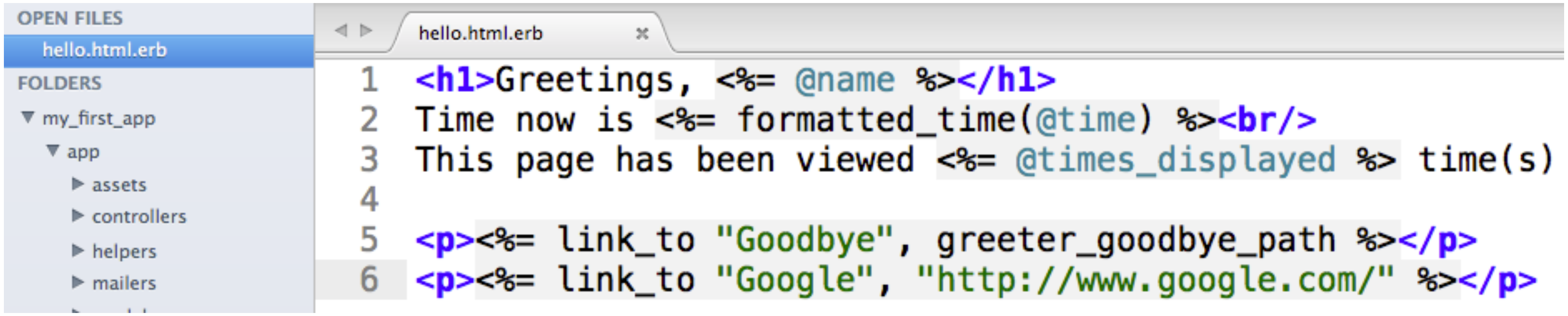
```
Greetings, Joe

Time now is 10:32AM
This page has been viewed 1 time(s)
```

Rails built-in helpers – link_to

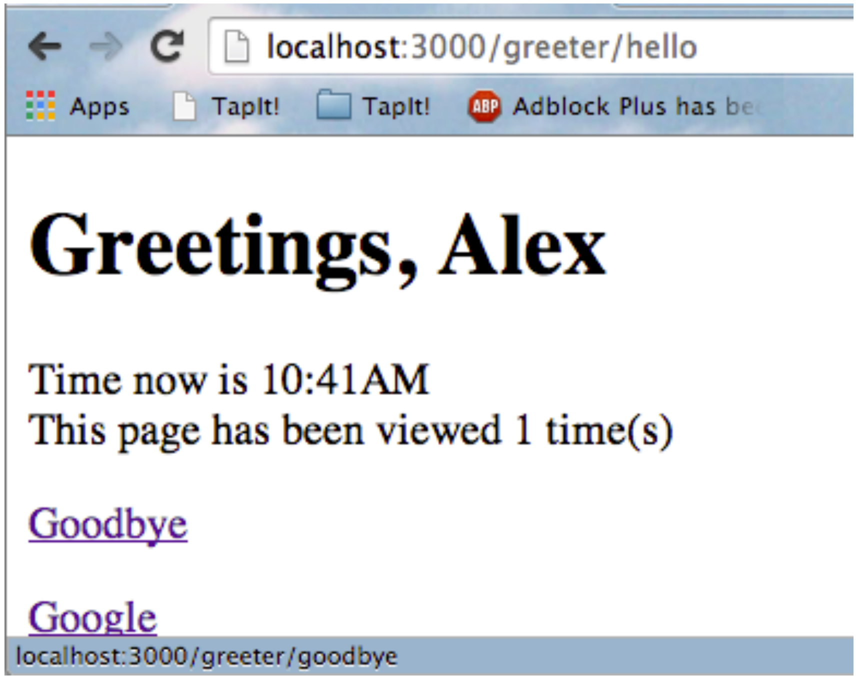
- Rails provides many built_in helpers
- `link_to name, path`
  - Hyperlink generator that displays the `name` and links to the `path`
  - Path could either be a regular string or a route defined in the routes.rb file ending with `_url` (full path) or `_path` (relative path)
  - `_url` and `_path` used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

link_to in action



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

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



Greetings, Alex

Time now is 10:41AM
This page has been viewed 1 time(s)

Goodbye

Google

localhost:3000/greeter/goodbye

greeter_goodbye derived from routes.rb ($rake routes)