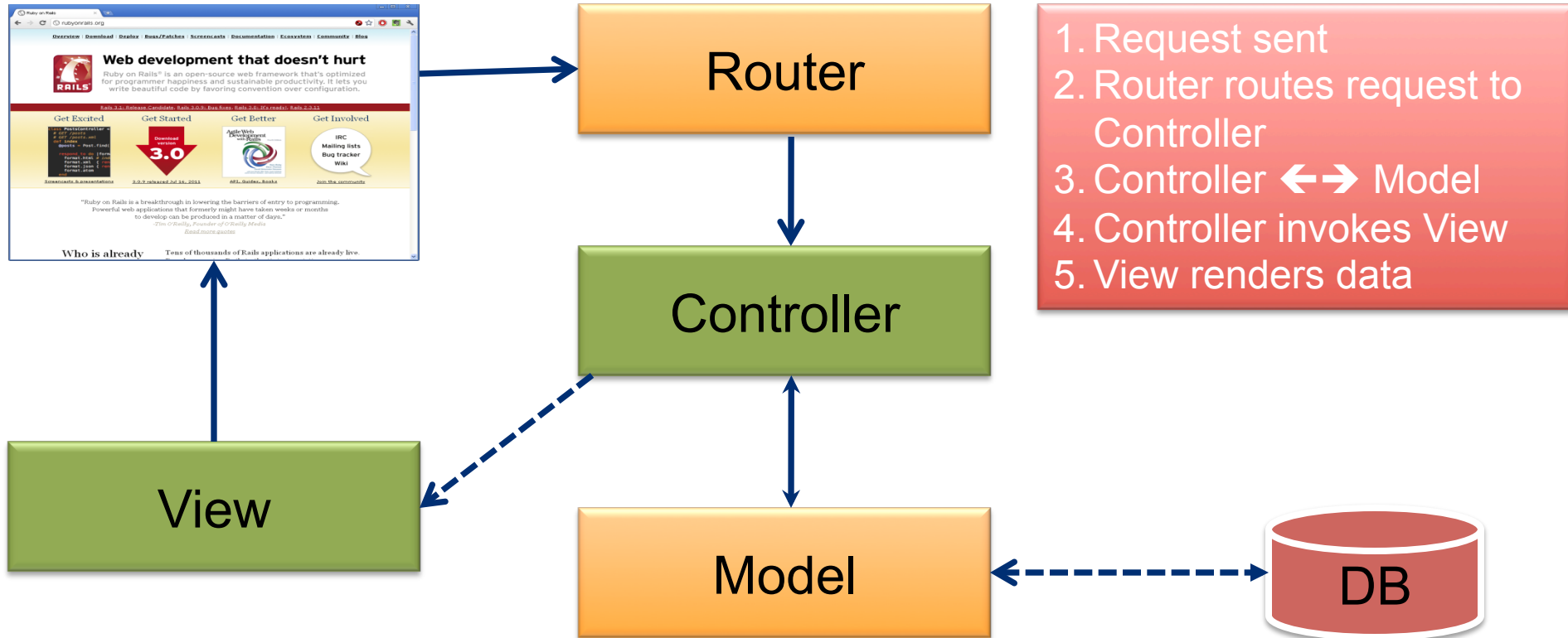


In this lecture, we will discuss...

- ✧ Introduction to Action Pack



ActionController + ActionView = AP



Blog Scaffolding

```
~/my_blog$ rails g scaffold post title content:text
  invoke  active_record
  create   db/migrate/20151001131254_create_posts.rb
  create   app/models/post.rb
  invoke   test_unit
  create   test/models/post_test.rb
  create   test/fixtures/posts.yml
  invoke   resource_route
   route   resources :posts
  invoke   scaffold_controller
  create   app/controllers/posts_controller.rb
  invoke   erb
  create   app/views/posts
  create   app/views/posts/index.html.erb
  create   app/views/posts/edit.html.erb
  create   app/views/posts/show.html.erb
  create   app/views/posts/new.html.erb
  create   app/views/posts/_form.html.erb
```

\$rails new my_blog

```
~/my_blog$ rake db:migrate
== 20151001131254 CreatePosts: migrating =====
-- create_table(:posts)
   -> 0.0009s
== 20151001131254 CreatePosts: migrated (0.0010s)
```



Scaffolding: Explaining The Magic

✧ Scaffolding creates:

1. Migration
2. Model
3. *Routes*
4. *Restful Controller*
5. *Views*
6. More...



Discuss these now...

ActionView: ERB

- ✧ HTML file with an .erb extension
- ✧ ERb is a **templating** library (similar to JSP) that lets you **embed** Ruby into your html
- ✧ Two tag patterns to learn:
 - `<% ...ruby code... %>` - evaluate Ruby code
 - `<%= ...ruby code... %>` - output evaluated Ruby code

...And of course, there are many view helpers that assist in code creation, like `link_to...`



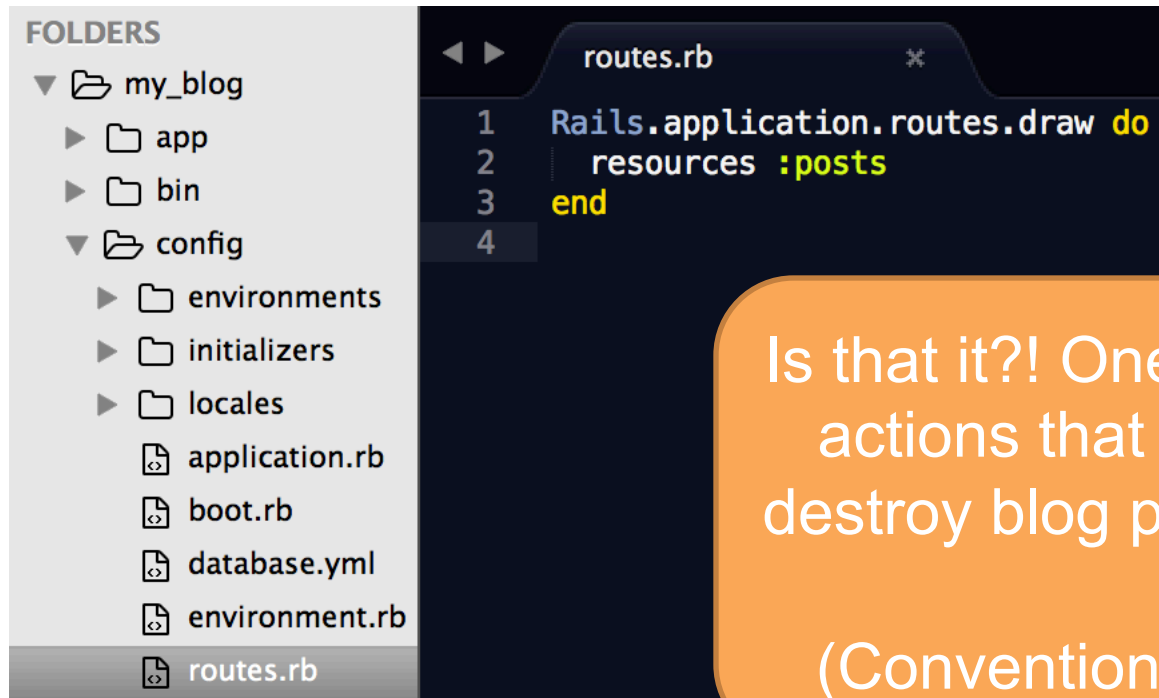
Action Controller

- ✧ Ruby class containing **one or more** actions
- ✧ Each action is responsible for **responding to a request** to perform some task
- ✧ Unless otherwise stated – when an action is **finished firing** (or even if the action is not physically present) it **renders a view** with the same name as the action
- ✧ The action always needs to be mapped in **routes.rb**



Speaking of Routes...

✧ Let's see scaffolded routes



The image shows a screenshot of a Rails application's file explorer on the left and the `routes.rb` file in the editor on the right. The file explorer shows a project named `my_blog` with folders `app`, `bin`, and `config`. The `config` folder is expanded, showing `environments`, `initializers`, `locales`, `application.rb`, `boot.rb`, `database.yml`, `environment.rb`, and `routes.rb`. The `routes.rb` file is open in the editor, showing the following code:

```
1 Rails.application.routes.draw do
2   resources :posts
3 end
4
```

Is that it?! One line lets you route to actions that list, create, edit and destroy blog postings? Impossible...

(Convention Over Configuration)

Summary

- ✧ Action Pack is Controller and View work together to let you interact with resources in the Model layer

What's Next?

- ✧ REST and Rails

