

# 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"`



Git repo should be INSIDE

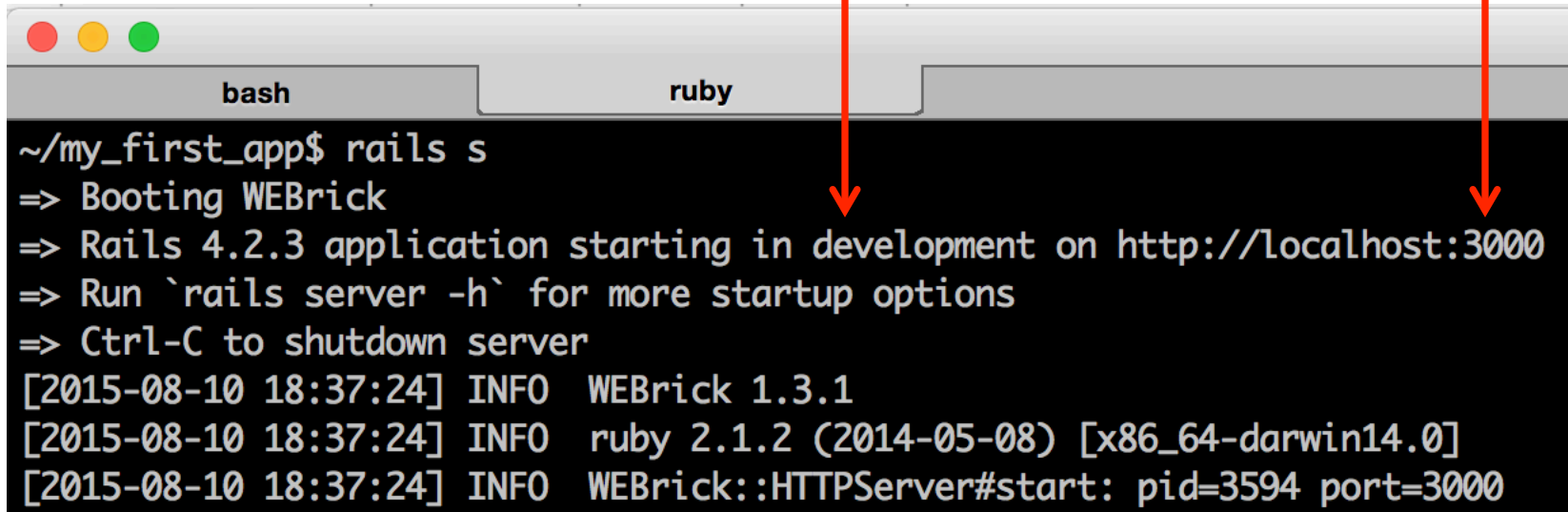
# Running the App

- ✧ 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)

Development  
environment

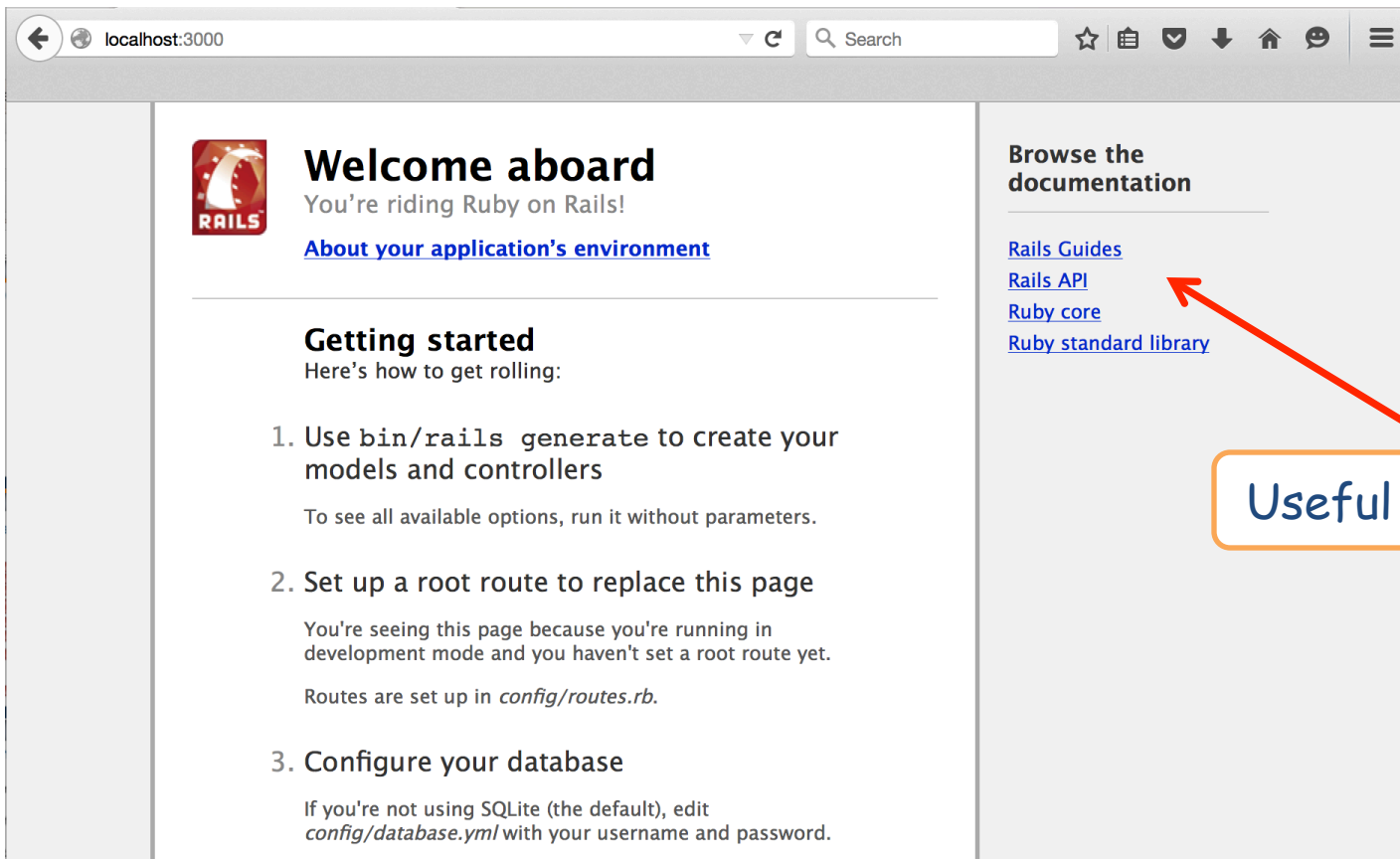
localhost on port  
3000




A terminal window with a title bar containing three colored buttons (red, yellow, green) and two tabs labeled 'bash' and 'ruby'. The 'ruby' tab is active. The terminal output shows the command `rails s` being executed, followed by several informational messages from WEBrick and Ruby. Two orange boxes with arrows point to the terminal text: one points to 'Development environment' and the other points to 'localhost on port 3000'.

```
~/my_first_app$ rails s
=> Booting WEBrick
=> Rails 4.2.3 application starting in development on http://localhost:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server
[2015-08-10 18:37:24] INFO  WEBrick 1.3.1
[2015-08-10 18:37:24] INFO  ruby 2.1.2 (2014-05-08) [x86_64-darwin14.0]
[2015-08-10 18:37:24] INFO  WEBrick::HTTPServer#start: pid=3594 port=3000
```

# Running the App (Continued)



localhost:3000

 **Welcome aboard**  
You're riding Ruby on Rails!  
[About your application's environment](#)

---

**Getting started**  
Here's how to get rolling:

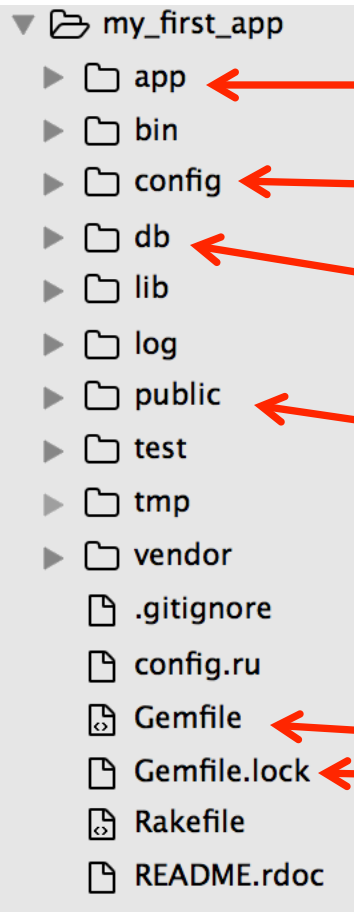
1. Use `bin/rails generate` to create your models and controllers  
To see all available options, run it without parameters.
2. Set up a root route to replace this page  
You're seeing this page because you're running in development mode and you haven't set a root route yet.  
Routes are set up in `config/routes.rb`.
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](#)

**Useful Resources!**

# 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

**FOLDERS**

- ▼ my\_first\_app
  - ▶ app
  - ▶ bin
  - ▶ config
  - ▶ db
  - ▶ lib
  - ▶ log
  - ▼ public
    - 404.html
    - 422.html
    - 500.html
    - favicon.ico
    - hello\_static.html

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

Totally Static

localhost:3000/hello\_static.html

**Will never generate dynamic content :(**

# 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

