WILLARD STERLING

RUBY ON RAILS INTRODUCTION

Introduce Yourself
We will be learning about Ruby on Rails
How to use a web IDE to build our project
We will use GIT for revision control, sharing and publishing
Heroku to host our app

FRAMEWORK

Ruby on Rails

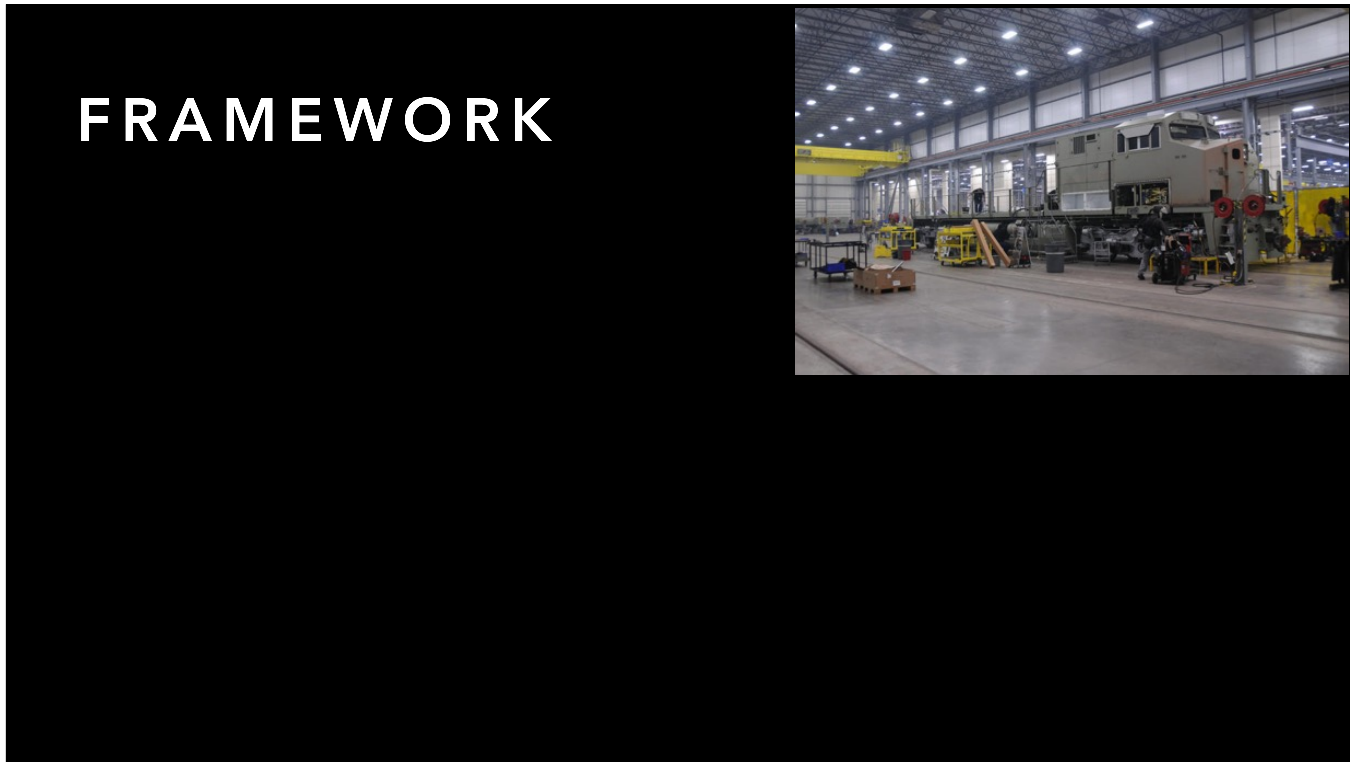
Is a web application framework

Written in Ruby of course

It does all of the common tasks or “infrastructure” that our common among web applications so you can focus on the unique aspects of the application you are building.

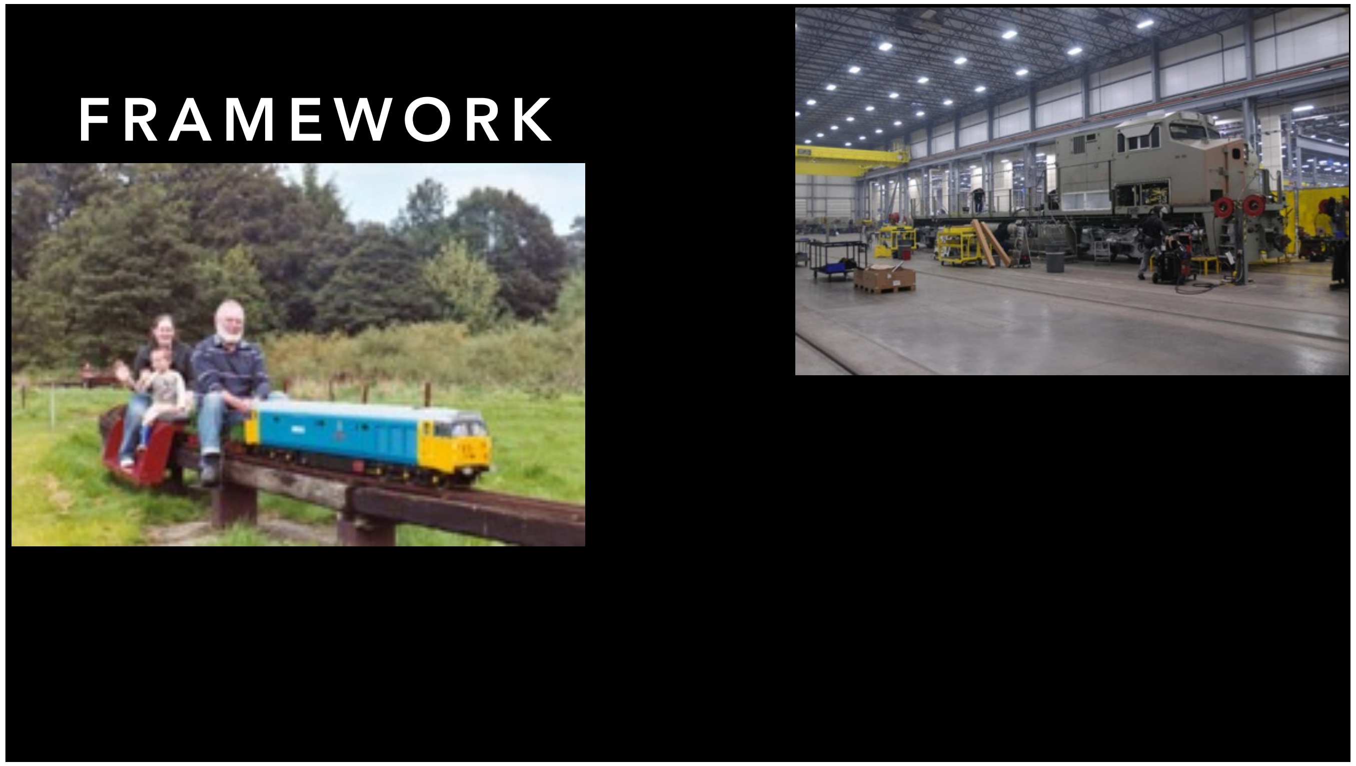
“The RAILS Way”- CoC, DRY, MVC

“Convention over configuration” or “strongly opinionated” Rails will look for your code in specific places in the project structure with out any configuration. So if you stick to convention you just need to stick code snippets in the correct place to make large changes. - hint at models and views but avoid deep discussion



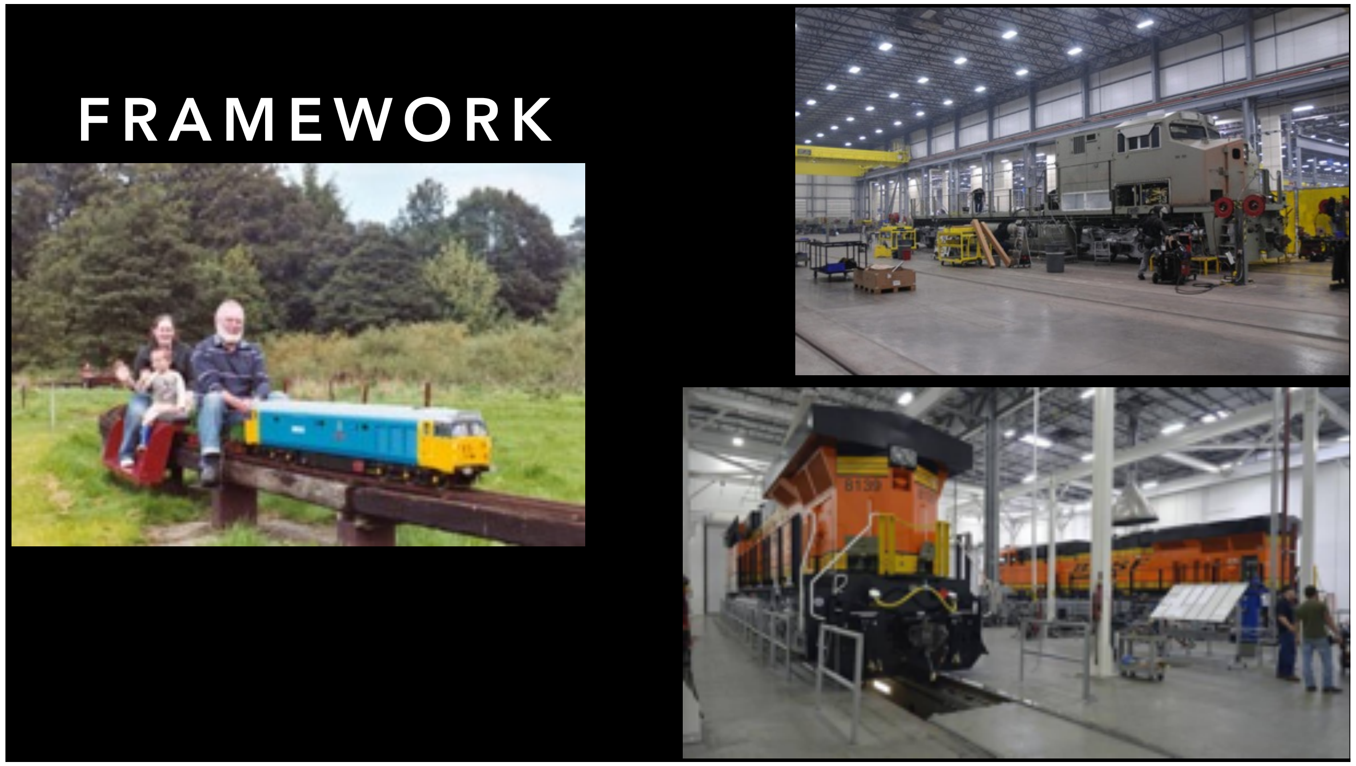
FRAMEWORK

Ruby on Rails
Is a web application framework
Written in Ruby of course
It does all of the common tasks or “infrastructure” that our common among web applications so you can focus on the unique aspects of the application you are building.
“The RAILS Way”- CoC, DRY, MVC
“Convention over configuration” or “strongly opinionated” Rails will look for your code in specific places in the project structure with out any configuration. So if you stick to convention you just need to stick code snippets in the correct place to make large changes. - hint at models and views but avoid deep discussion



FRAMEWORK

Ruby on Rails
Is a web application framework
Written in Ruby of course
It does all of the common tasks or “infrastructure” that our common among web applications so you can focus on the unique aspects of the application you are building.
“The RAILS Way”- CoC, DRY, MVC
“Convention over configuration” or “strongly opinionated” Rails will look for your code in specific places in the project structure with out any configuration. So if you stick to convention you just need to stick code snippets in the correct place to make large changes. - hint at models and views but avoid deep discussion



FRAMEWORK

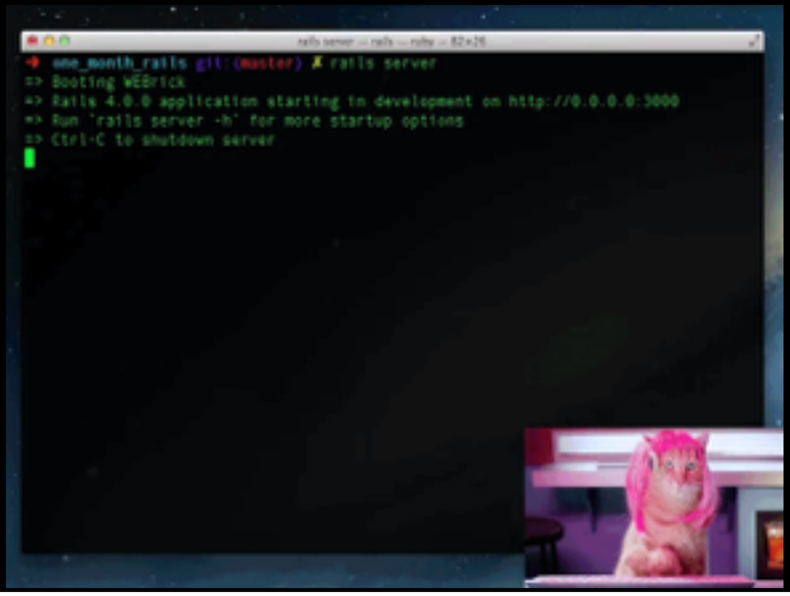
Ruby on Rails
Is a web application framework
Written in Ruby of course
It does all of the common tasks or “infrastructure” that our common among web applications so you can focus on the unique aspects of the application you are building.
“The RAILS Way”- CoC, DRY, MVC
“Convention over configuration” or “strongly opinionated” Rails will look for your code in specific places in the project structure with out any configuration. So if you stick to convention you just need to stick code snippets in the correct place to make large changes. - hint at models and views but avoid deep discussion

LET'S GET STARTED

EXERCISE 1- DEVELOPMENT ENVIRONMENT

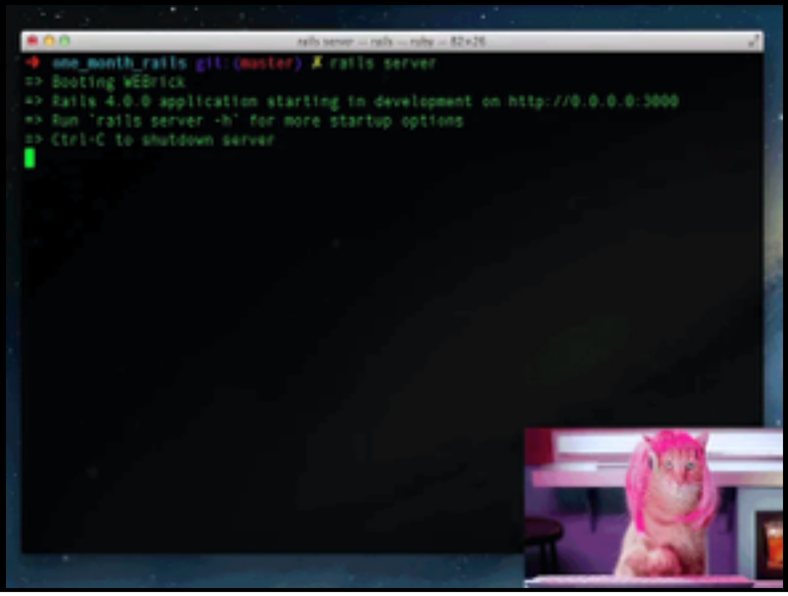
- Follow Exercise 1 README - https://github.com/will-sterling/RAILS-Intro/blob/exercise1/README.md

RAILS COMMAND LINE



```
➜  one_month_rails git:(master) ✗ rails server
=> Booting WEBrick
=> Rails 4.0.0 application starting in development on http://0.0.0.0:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server
```

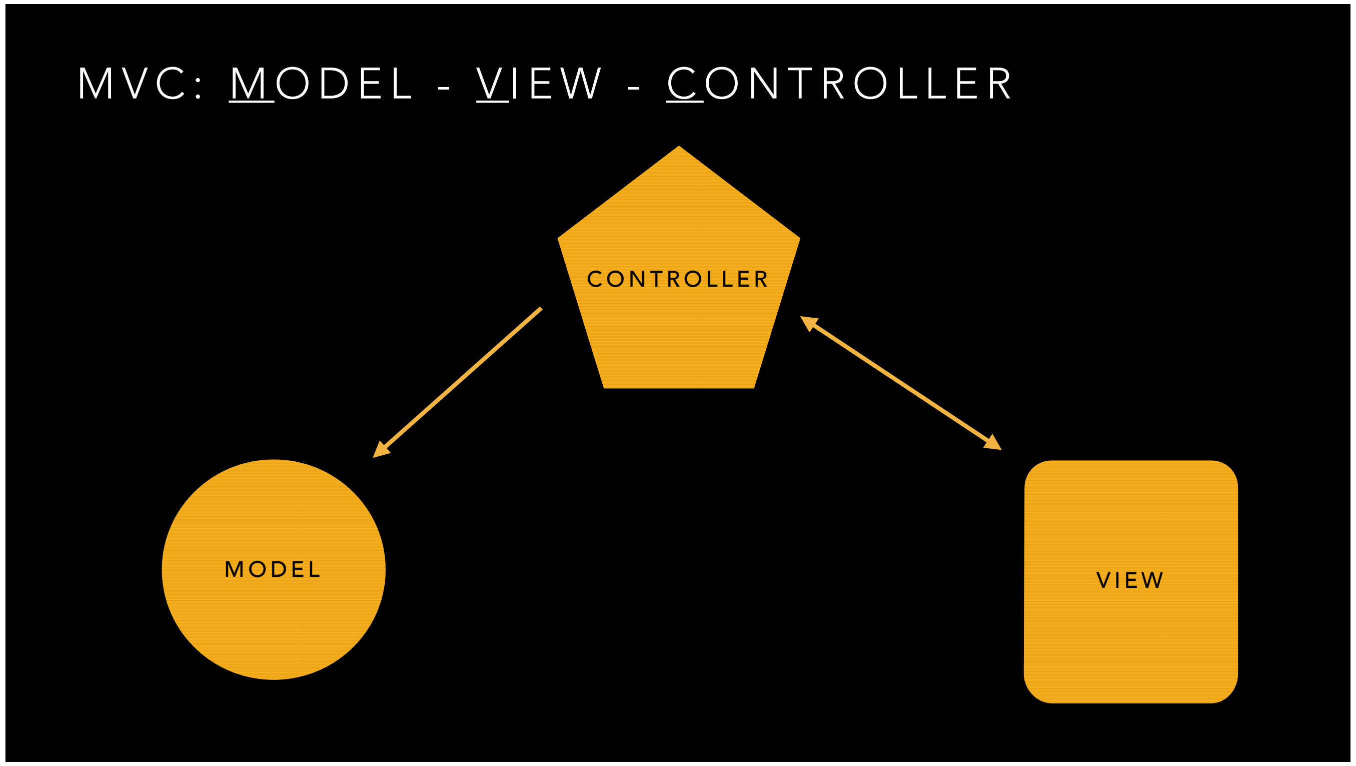
RAILS COMMAND LINE



```
➜  one_month_rails git:(master) ✗ rails server
=> Booting WEBrick
=> Rails 4.0.0 application starting in development on http://0.0.0.0:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server
```

EXERCISE 2- CREATE A RAILS APP

- Follow Exercise 2 README - https://github.com/will-sterling/RAILS-Intro/blob/exercise2/README.md



MVC: MODEL - VIEW - CONTROLLER

Just cover the basics of what each of these are. We will visit each later on and in the exercises.

VIEWS

HTML/JS page templates presented to user
Should only contain minimal code. e.x. looping through array to display list of data
Should not contain business logic
Template languages ERB:default, HAML

ROUTES

- rake routes
- config/routes.rb
- Operation on resource Method & URI Controller action
- Index (list) movies GET /movies index
- Read (show) existing movie GET /movies/: id show
- Display fill-in form for new movie GET /movies/ new new
- Create new movie from filled-in form POST /movies create
- Display form to edit existing movie GET /movies/: id/ edit edit
- Update movie from fill-in form PUT /movies/: id update
- Destroy existing movie DELETE /movies/: id destroy

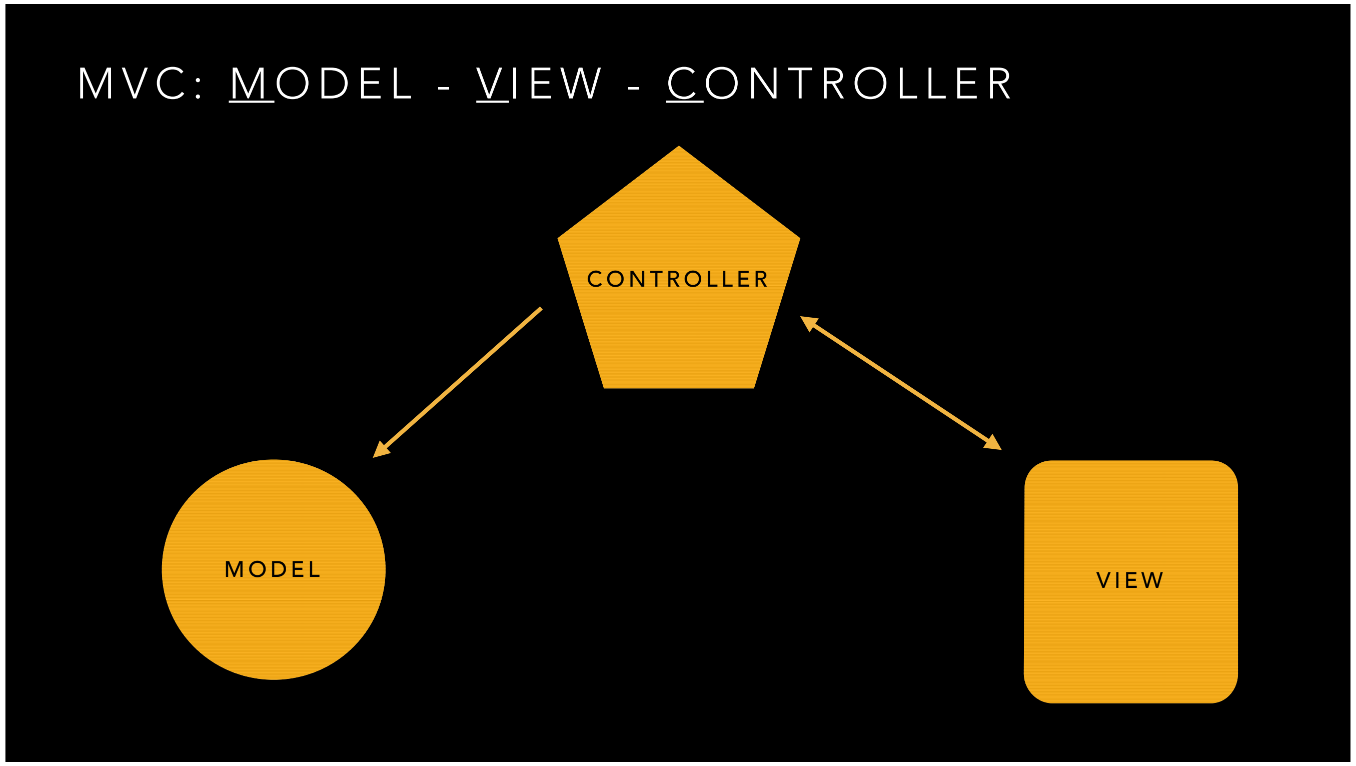
- Discuss that the routes are RESTful
- Discuss the need for new and create. New is only needed for interactive user facing web app. API user sends all required data to create without calling new.

EXERCISE 3- HELLO WORLD

- Follow Exercise 3 README - https://github.com/will-sterling/RAILS-Intro/blob/exercise3/README.md

EXERCISE 4 - HEROKU

- Follow Exercise 4 README - https://github.com/will-sterling/RAILS-Intro/blob/exercise4/README.md



MVC: MODEL - VIEW - CONTROLLER

We saw view earlier. Now we are going to talk about models and controllers.

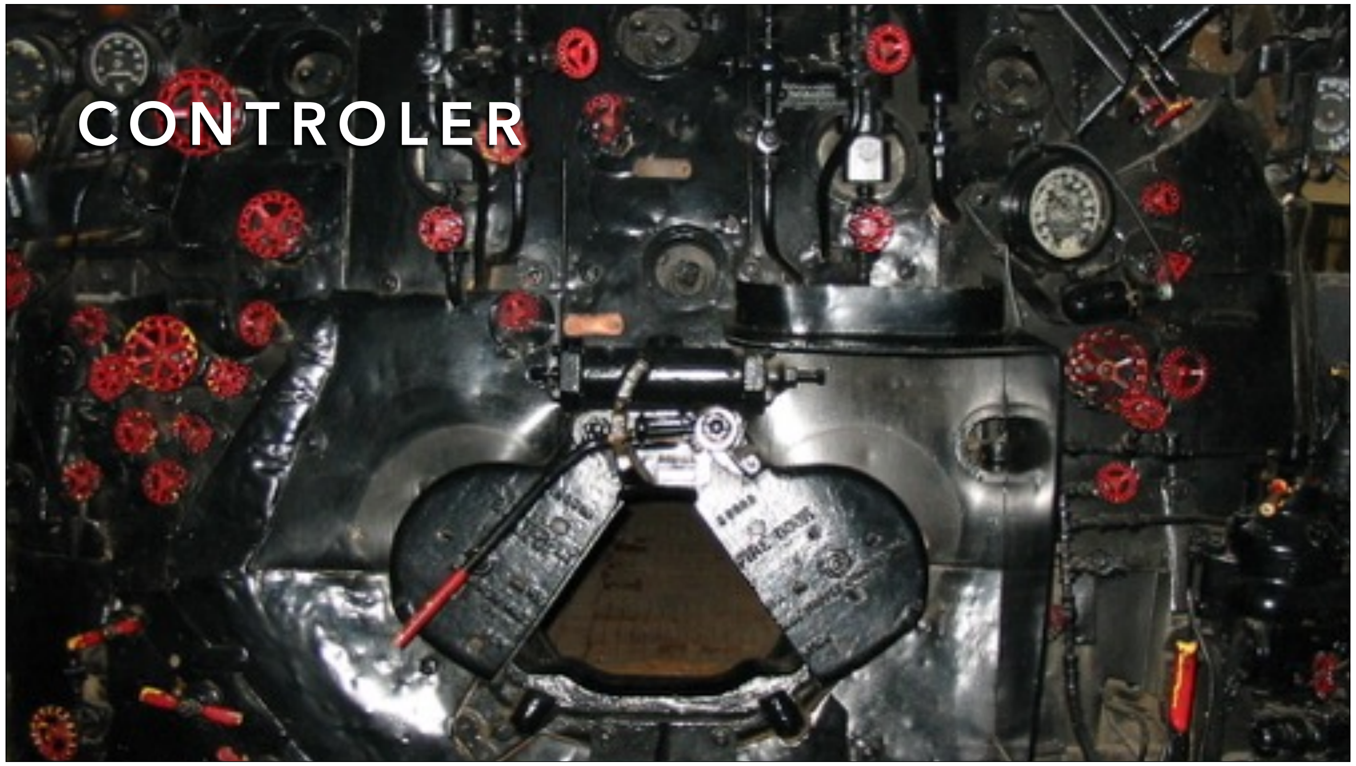
MODEL

ACTIVE RECORD

https://en.wikipedia.org/wiki/Active_record_pattern - “In software engineering, the active record pattern is an architectural pattern found in software that stores in-memory object data in relational databases. It was named by Martin Fowler in his 2003 book Patterns of Enterprise Application Architecture.[1] The interface of an object conforming to this pattern would include functions such as Insert, Update, and Delete, plus properties that correspond more or less directly to the columns in the underlying database table.”

- Serialize/Deserialize Data
- Provides methods of describing data relationships and access to data
- Provides methods for updating RDBMS Schema
- Manipulate, change, delete data

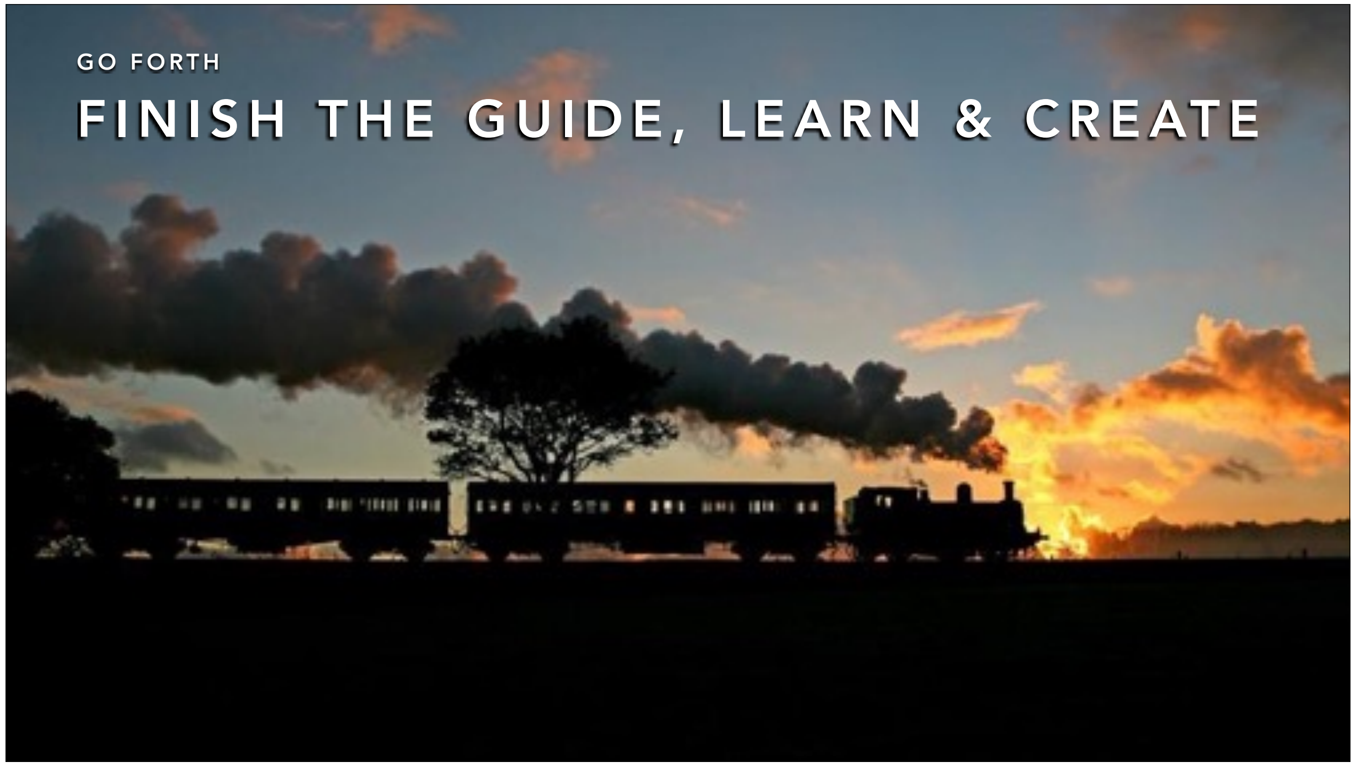
CONTROLER



Prepare data for Presentation
Contain Business Logic
Methods Matching CRUD operations

EXERCISE 5 - FOLLOW THE GUIDE

- http://guides.rubyonrails.org/ Getting Started - Chapter 5
- When done push to Heroku



GO FORTH

FINISH THE GUIDE, LEARN & CREATE

Finish the RAILS Guide, getting started is just the beginning of the guide, there is in depth material on each topic we discussed today plus more. Make sure to read the Debugging chapter early, don’t wait till the end!

Create your own side project! Put it on GitHub! Put it on Heroku!

Hang out in IRC

RESOURCES

- sterling.codes
- slides with notes: http://will-sterling.github.io/Rails-Intro-Slides/slides_with_notes.pdf
- http://guides.rubyonrails.org
- http://rubyonrails.org/community
- https://devcenter.heroku.com/articles/getting-started-with-ruby
- 🐦 @will-sterling