

Welcome 🎉

# About

Third meeting - welcome back!

Website: <http://railsclub.me>

Facebook group: <http://fb.railsclub.me>

# Normal Meeting Procedure

- Say what we are doing that day
- Learn a new skill or two
- Practice the skills
- Work on own applications

Any first-time club people today?

# Today's Meeting Procedure

- Rails MVC
- Rails generate
- The Database
- Practice Application

Come up with something you want  
to build. 🧠

# Come up with something you want to build.

Can be a blog, portfolio website, Pinterest clone, whatever you want.

The goal is to have this project built using skills you learn here.

Okay. Here we go!



# Rails MVC

- A way to organize data
- Used by Rails and many other popular frameworks

# MVC: The Basics

- Model
- View
- Controller

# Part 1: The Model

- An attribute, or data object
- SINGULAR
- Examples: Book, User, Movie

Models are stored here:

```
/app/models/book.rb
```

```
/app/models/user.rb
```

```
/app/models/movie.rb
```

## Part 2: The Controller

- Does something with the MODEL and shows it to you using a VIEW
- PLURAL
- Examples: BooksController, UsersController, MoviesController

Controllers are stored here:

```
/app/controllers/books_controller.rb  
/app/controllers/users_controller.rb  
/app/controllers/movies_controller.rb
```

# Controller Actions

- The actions of a controller
- The actual place where you manipulate the MODEL
- Examples:
  - `BooksController#new`
  - `BooksController#show`

## Example Controller

```
class BooksController < ApplicationController

  def new
    @book = Book.new
  end

  def show
    @book = Book.find_by(params[:id])
  end
end
```

## Part 3: Views

- Receives the MODEL from the CONTROLLER and shows it
- Directly corresponds to a controller action
- Examples: BooksController#new, UsersController#new, MoviesController#new

Controllers are stored here:

```
/app/views/books/new.html.erb  
/app/views/users/new.html.erb  
/app/views/movies/new.html.erb
```

## Example View

Controller Action (BooksController#show):

```
class BooksController < ApplicationController
  def show
    @book = Book.find_by(params[:id])
  end
end
```

View (app/views/books/show.html.erb):

```
<h1> <%= @book.title %> </h1>
```



# Rails Commands:

Rails has awesome commands designed to help you create  
MODELS, VIEWS, and CONTROLLERS

# rails generate

- RAILS can GENERATE things for you!

## rails generate model book

- RAILS will GENERATE a MODEL named BOOK for you
- Won't create a controller or any views for you to manipulate the model, though.

## rails generate controller books

- RAILS will GENERATE a CONTROLLER named BOOK for you
- Won't create a model or any views for you, though.

# You cannot generate views by themselves

- Whenever you create a controller action, make sure you have a view that corresponds to it!

Generate a controller with actions (and views that go with them!)

```
rails g controller books show
```

- Books is your controller
- Show is your action (will create app/views/books/show.html.erb for you!)

The awesomest command in all of  
Rails

rails g scaffold



# A scaffold is the culmination of MVC in Rails.

If you create a scaffold, it will create what is known as a CRUD resource

- Create
- Read
- Update
- Destroy

Also, it comes with a model, controller, and corresponding views

# The Database

- A place to store data!

There are different kinds of data types

- String (a string of characters)
- Text (a large amount of text)
- Integer (a number)
- Datetime (a date and a time)
- Boolean (true or false)

REMEMBER:

ALL DATA IS STORED ON THE MODEL

# Example 1

Say you are creating a Post model and you want to have a post have a title that is a string.

```
rails g model post title:string
```

**It's that easy!**

## Example 2

Say you are creating a Post scaffold and you want to have a post have a title that is a string.

```
rails g scaffold post title:string
```

**It's that easy!**

With your new knowledge...

# Log into Nitrous



cd workspace

# Today's Project:

- Generate a new Rails application named test
- Generate a post scaffold
- That post should have title:string and body:text
- Create a post and then show me.

```
rails server -b 0.0.0.0
```

OMG

# Before we go

- Make sure you join our Facebook group: <http://fb.railsclub.me>
- Next meeting is TBD! (I have to talk with the office)
- Message me on FB or email me if you have any questions
- Feel free to read up about Rails or learn skills in your freetime

See you soon! 🙌