



Getting Started with Ruby on Rails

Victoria Gonda

[@TTgonda](#)

victoria@collectiveidea.com



Setup

- Find Rails Workshop Materials at:
https://github.com/vgonda/rails_workshop
- We're going to use a cloud editor to minimize setup
- Sign up for a Code Anywhere account with the instructions found at the above link.



What's Ruby?

- Programming language created by Yukihiro Matsumoto (Matz) for fun
- Interpreted, object-oriented, dynamically typed, scripting language
- Open source
- Syntax cheatsheet on the [github](#)



What's Rails?

- Full stack web framework
- Database agnostic
- Open source
- Used by Basecamp, GitHub, Shopify, Airbnb, Twitch, SoundCloud, Hulu, Zendesk, Square, Highrise



What are we making?

- A timeline app, similar to a Twitter feed
- Functional over pretty in order to cover more of the framework



Let's make something!



- `$ rails new timeline_app`
- `$ cd timeline_app`



Back-end: Database

- Rails doesn't really care what database we use. We get to interact with them all the same
- We are going to create an event

- `$ rails generate migration CreateEvent message:text`
- `Add t.timestamps to migration`
- `$ rake db:migrate`



Front-end: MVC

- **Model-View-Controller is a common pattern in application development**
- **It's a way of organizing the different parts of the project**

Model

- Can be thought of as “things” or “data”
- Objects that handle the business logic
- DB models are tied to these
- Doesn't know anything about the view



In `app/models/event.rb`

- Create an Event class that extends ApplicationRecord
- `$ rails console`
- Try
 - Initializing an instance of Event
 - Saving event to DB
 - Finding an event
 - Experiment with Ruby syntax (from cheat sheet)



Controller

- Hooks up the models and views
- Decides what to do and display when you go to a URL
- Handles actions sent from the view
- CRUD: Create, Read, Update, Delete



In
`app/controllers/events_controller.rb`

- **Create EventsController that extends ApplicationController**
- **Define an index method**
- `@events = Event.all`



Routes

- What URLs are available
- What controller method for each URL
- All actions are an HTTP request, and this is where you say what they are

In config/routes.rb

- `root "events#index"`
- `resources :events`
- `$ rake routes`



View

- The stuff the user sees
- Minimal, if any, logic
- Sends any actions to the controller
- `erb <% %>` and `<%= %>`



In

app/views/events/index.html.erb

- ```
<% @events.each do |event| %>
 <%= event.message %>
<% end %>
```
- **\$ rails server**
- **Ctrl+C stops server**



# MVC Review

- **Model: business logic**
- **View: pages**
- **Controller: actions**
- **Mostly connected by names, but can be set explicitly**



# Create “show” page

- Like “index” but named “show”
- Only want one event
- Hints
  - Need controller method and view file
  - `@event = Event.find(params[:id])`

# Add link to “show”

- Rails provides URL helpers
- `<%= link_to "Show", event_path(event) %>`

# Create “new” page

- **Controller methods** new, create, and event\_params
- ```
<%= form_for @event do |form| %>  
  <%= form.text_field :message %>  
  <%= submit_tag "Save" %>  
<% end %>
```
- ```
<%= link_to "New", new_event_path %>
```



# Create “edit” page

- Define edit and update
- Hint:  
`@event = Event.find(params[:id])`  
`@event.update(event_params)`
- Link to `edit_event_path(event)`

# Partials

- A way to reuse view code
- `<%= render partial: "form" %>`



# Delete an Event

- What do we need in order to do this?

# Model Validations

- **In model:** `validates :message, presence: true`
- **In controller:**

```
@event = Event.new(event_params)
if @event.save
 redirect_to events_path
else
 render :new
end
```
- **In view:**

```
<% if @event.errors.any? %>
 <% @event.errors.full_messages.each do |message| %>
 <%= message %>
 <% end %>
<% end %>
```



# Validate “update” too

- This will mean changing the controller’s update method
- Hint: `event.update` calls and returns the result of save

# Adding attributes

- `$ rails generate migration AddRelevanceToEvent relevance:integer`
- Add default: 0 to migration
- `$ rake:db migrate`
- Add to all the different views
- Hint: `form.number_field`
- Hint: `.permit(:message, :relevance)`



# Adding custom actions

- In `routes.rb`  

```
resources :events do
 member do
 get :add_relevance
 end
end
```
- `$ rake routes`
- Create controller method
- Add link to views



# Now do decrement

- Add route
- Add controller method
- Add links



# What's next?

- **Expand on this, or create something on your own**
- **Contribute on GitHub**
- **Learn more!**



# References

- **Bruce A. Tate. “Seven Languages in Seven Weeks**
- **Michael Hartl. “Ruby On Rails Tutorial (Rails 5): Learn Web Development with Rails”**
- **Rails cheatsheet: <http://www.pragtob.info/rails-beginner-cheatsheet/>**





# Contact

- Email: [victoria@collectiveidea.com](mailto:victoria@collectiveidea.com)
- Twitter: @TTgonda
- Swing by the office!

