This covers how to build a simple CRUD based app with RESTful API interactions using Ruby, Rails and React. The assumption here is that you have Postgres and the minimum prerequisites for Ruby and Rails.

**Prerequisites**

rvm install 2.2.5

rvm use 2.2.5

gem update rails

***Step 1 – Creating a RESTful API Interface with JSON Responses***

**Create Project Database and Models**

rails new restaurants --api –T

cd restaurants

vi Gemfile [remove MySql gem and use Postgres gem instead as shown below]

...

gem 'pg', '~> 0.18'

...

bundle update

bundle install

vi config/database.yml

default: &default

adapter: postgresql

encoding: unicode

pool: <%= ENV.fetch("RAILS\_MAX\_THREADS") { 5 } %>

development:

<<: \*default

database: restaurants\_development

test:

<<: \*default

database: restaurants\_test

production:

<<: \*default

database: restaurants\_production

rails s

Ctrl + c

rails g model Restaurant name:string location:string rating:integer

(this creates a model file and a migration file)

[must add create restaurants\_development database to Postgres]

rails db:migrate

(creates actual table in database)

**Create Controllers**

rails g controller Restaurants

(creates app/controllers/restaurants\_controller.rb)

vi config/routes.rb [add resource line for restaurants]

Rails.application.routes.draw do

resources :restaurants

end

rails routes

vi app/controllers/restaurants\_controller.rb

class RestaurantsController < ApplicationController

before\_action :set\_restaurant, only: [:show, :update, :destroy]

# GET /restaurants

def index

@restaurants = Restaurant.all

json\_response(@restaurants)

end

# POST /restaurants

def create

@restaurant = Restaurant.create!(restaurant\_params)

json\_response(@restaurant, :created)

end

# GET /restaurants/:id

def show

json\_response(@restaurant)

end

# PUT /restaurants/:id

def update

@restaurant.update(restaurant\_params)

head :no\_content

end

# DELETE /restaurants/:id

def destroy

@restaurant.destroy

head :no\_content

end

private

def restaurant\_params

# whitelist params

params.permit(:name, :created\_by)

end

def set\_restaurant

@restaurant = Restaurant.find(params[:id])

end

end

touch app/controllers/concerns/response.rb

vi app/controllers/concerns/response.rb

module Response

def json\_response(object, status = :ok)

render json: object, status: status

end

end

touch app/controllers/concerns/exception\_handler.rb

vi app/controllers/concerns/exception\_handler.rb

module ExceptionHandler

extend ActiveSupport::Concern

included do

rescue\_from ActiveRecord::RecordNotFound do |e|

json\_response({ message: e.message }, :not\_found)

end

rescue\_from ActiveRecord::RecordInvalid do |e|

json\_response({ message: e.message }, :unprocessable\_entity)

end

end

end

vi app/controllers/application\_controller.rb

class ApplicationController < ActionController::API

include Response

include ExceptionHandler

end

**Test REST**

rails s

In Browser / Postman:

GET [http://localhost:3000/restaurants/](http://localhost:3000/resorts/)

POST http://localhost:3000/restaurants/

x-www-form-urlencoded

name Brighton

location Big Cottonwood Canyon

rating 4

GET [http://localhost:3000/restaurants/](http://localhost:3000/resorts/)

GET [http://localhost:3000/restaurants/](http://localhost:3000/resorts/)1

***Step 2 – Transition to a React Project***

**Add Webpack**

vi Gemfile

...

gem 'webpacker'

...

bundle install

rails webpacker:install

rails webpacker:install:react

(/app/javascript/ folder is created)

touch Procfile

vi Procfile

web: bundle exec rails s

webpacker: ./bin/webpack-dev-server

**Add Foreman**

gem install foreman

foreman start

[if foreman doesn’t start, try and start again: bundle binstubs bundler --force]

In Browser / Postman (notice a new port):

GET [http://localhost:5000/restaurants/](http://localhost:3000/resorts/)

Ctrl + c

**Create Landing Page**

rails generate controller Home index

(creates app/controllers/home\_controller.rb and changes routes.rb file)

rails routes

Change config/routes.rb to the following:

...

root to: 'home#index'

...

mkdir app/views/home

touch app/views/home/index.html.erb

vi app/views/home/index.html.erb

<h1>hello rails</h1>

touch app/views/layouts/application.html.erb

vi app/views/layouts/application.html.erb

<!DOCTYPE html>

<html>

<head>

<title>Restaurants</title>

<%= csrf\_meta\_tags %>

<%= javascript\_pack\_tag 'application' %>

</head>

<body>

<%= yield %>

</body>

</html>

vi javascript/packs/application.js

import React from 'react';

import { render } from 'react-dom';

import MyReactComponent from '../components/MyReactComponent';

document.addEventListener('DOMContentLoaded', () => {

console.log('dom content loaded');

const container = document.body.appendChild(document.createElement('div'));

render(<MyReactComponent/>, container);

});

mkdir app/javascript/components

touch app/javascript/components/MyReactComponent.js

vi app/javascript/components/MyReactComponent.js

import React from 'react';

export default class MyReactComponent extends React.Component {

render() {

return (

<h1>My React component.</h1>

);

}

}

vi app/controllers/application\_controller.rb

(you might just be changing the first line from “API” to “Base”)

class ApplicationController < ActionController::Base

include Response

include ExceptionHandler

end

foreman start

In Browser:

http://localhost:5000/

Ctrl-C

***Step 3 – React Calls RESTful API for Data***

**Add Bootstrap and Fetch**

vi app/views/layouts/application.html.erb

(add the following to the <head> portion of the page)

...

<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css"/>

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"/>

<script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.2/js/bootstrap.min.js"/>

<script src="//cdnjs.cloudflare.com/ajax/libs/fetch/2.0.3/fetch.min.js"/>

...

**Add a New App Component**

touch app/javascript/components/App.js

vi app/javascript/components/App.js

import React from 'react';

export default class App extends React.Component {

constructor() {

super();

this.state = {

data:[]

};

}

componentDidMount() {

fetch("/restaurants")

.then(res => res.json())

.then(res => this.setState({data: res}))

.catch(e => alert(e));

}

render() {

return (

<div className='table-responsive col-sm-offset-1 col-sm-10'>

<table className='table table-condensed table-striped table-bordered table-hover no-margin'>

<thead>

<tr>

<th>Restaurant</th>

<th>Location</th>

<th align='center'>Rating</th>

</tr>

</thead>

<tbody>

{this.state.data.map((restaurant, i) => <TableRow key = {i} data = {restaurant} />)}

</tbody>

</table>

</div>

);

}

}

class TableRow extends React.Component {

render() {

return (

<tr>

<td>{this.props.data.name}</td>

<td>{this.props.data.location}</td>

<td align='center'>{this.props.data.rating}</td>

</tr>

);

}

}

**Use the New App.js Component**

vi javascript/packs/application.js

(modify this file to use the new App.js component)

import React from 'react';

import { render } from 'react-dom';

import App from '../components/App';

document.addEventListener('DOMContentLoaded', () => {

console.log('dom content loaded');

const container = document.body.appendChild(document.createElement('div'));

render(<App/>, container);

});

foreman start

In Browser:

http://localhost:5000/

**Useful Links**

https://scotch.io/tutorials/build-a-restful-json-api-with-rails-5-part-one

https://jeremykratz.com/blog/using-react-with-rails-and-webpack/

https://www.tutorialspoint.com/reactjs/index.htm

https://www.pluralsight.com/courses/reactjs-on-rails-building-full-stack-web-app

https://blog.jetbrains.com/ruby/2016/04/webinar-recording-react-js-from-a-rails-developers-perspective/

https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi