

Apple Exercise

Greetings,

I would have liked to put the finishing touches on this documentation. But in the interest of time, I am sending it over with full functionality and including the notes I've drafted along the way.

Please have a look at the screenshots provided by the hyperlinks below:

[OpenWeatherMap API](#)

[OpenWeatherMap API for 5 day / 3 hour forecast](#)

[Example 1 from forecast insert](#)

[Example 2 from forecast insert](#)

[Example 1 from Current weather data insert](#)

[Model \(2\) Schema](#)

[Forecast data table](#)

[Locations model main index](#)

[New Locations Creation](#)

[Show Model Render](#)

[Forecast Table](#)

[Show Location Model](#)

[Locations Model](#)

Directions:

Unit Tests!!

Detailed Comments/Documentation within the code + README.md

Include *Decomposition* of the Objects in the Documentation

Design patterns where Applicable

Scalability Considerations where Applicable

Naming Conventions as if it were enterprise-scale

Encapsulation (do not have 1 method doing 55 things)

Code Re-Use (don't over engineer or under engineer solution)

Best practices from the industry

Specification:

Forecast Application

1. Accept address as input
 - Form w/ Submit
 2. One Forecast per given Address, includes (at minimum):
 - Current Temperature
 - Bonus points:
 - High and Low Temperatures
 - Extended Forecast
 3. Display the Forecast details to the user
 4. Cache the Forecast for 30 minutes for all subsequent requests by zip code.
 - Display indicator for result if it came from the cache
-

Design

- Name: Forecast App (forecast_app)
- Data Models:
 1. Address
 - Unique ID
 - Street
 - City
 - Zip Code
 2. Forecast
 - Unique ID
 - Address Unique ID
 - Current Temperature
 - Day of Year

3. Forecast Extended

models descriptions

1. (table) Location(s)

- has a unique index `id`, same as `location_id` below, and index as `zipcode`, can have__many `metric(s)`

2. (table) Metric

- has a id
- one per day

Create `location` model with:

```
rails g model location zipcode:string:index city:string country:string{3}  
street:string lat:decimal long:decimal  
or use scaffold  
rails g scaffold Location zipcode:string:index street:string city:string  
country_code:string lat:decimal long:decimal
```

```
location  
-----  
id  
zipcode  
street  
city  
country  
lat  
long
```

Generate Scaffold:

```
invoke active_record  
create db/migrate/20221026074025_create_locations.rb  
create app/models/location.rb  
invoke test_unit  
create test/models/location_test.rb  
create test/fixtures/locations.yml  
invoke resource_route  
route resources :locations  
invoke scaffold_controller  
create app/controllers/locations_controller.rb  
invoke erb  
create app/views/locations  
create app/views/locations/index.html.erb
```

```
create      app/views/locations/edit.html.erb
create      app/views/locations/show.html.erb
create      app/views/locations/new.html.erb
create      app/views/locations/_form.html.erb
create      app/views/locations/_location.html.erb
invoke      resource_route
invoke      test_unit
create      test/controllers/locations_controller_test.rb
create      test/system/locations_test.rb
invoke      helper
create      app/helpers/locations_helper.rb
invoke      test_unit
invoke      jbuilder
create      app/views/locations/index.json.jbuilder
create      app/views/locations/show.json.jbuilder
create      app/views/locations/_location.json.jbuilder
```

```
rails g scaffold metric location_id:integer:index zipcode:string:index
temp:decimal min:decimal max:decimal day:integer
```

```
invoke      active_record
create      db/migrate/20221026074226_create_metrics.rb
create      app/models/metric.rb
invoke      test_unit
create      test/models/metric_test.rb
create      test/fixtures/metrics.yml
invoke      resource_route
  route      resources :metrics
invoke      scaffold_controller
create      app/controllers/metrics_controller.rb
invoke      erb
create      app/views/metrics
create      app/views/metrics/index.html.erb
create      app/views/metrics/edit.html.erb
create      app/views/metrics/show.html.erb
create      app/views/metrics/new.html.erb
create      app/views/metrics/_form.html.erb
create      app/views/metrics/_metric.html.erb
invoke      resource_route
invoke      test_unit
create      test/controllers/metrics_controller_test.rb
create      test/system/metrics_test.rb
invoke      helper
create      app/helpers/metrics_helper.rb
invoke      test_unit
invoke      jbuilder
create      app/views/metrics/index.json.jbuilder
create      app/views/metrics/show.json.jbuilder
create      app/views/metrics/_metric.json.jbuilder
```

Create `metric` model with:

`rails g model metric location_id:integer:index zipcode:string:index temp:decimal min:decimal max:decimal day:integer cached_at:datetime`

```
metric
-----
id
location_id
zipcode
temp
min
max
cached_at
```

Seed data:

- `<app_root>/db/seeds.rb`

Setup DB:

- `export RAILS_ENV=development && rails db:setup && rails db:migrate`

Test models with:

- from the rails app's root folder
- run: `rails test test/models/location_test.rb`
- run: `rails test test/models/metric_test.rb`

3. (array) Metric

- individual days correspond with the index of an Array of metric
- today is $d=0$, and each subsequent day is $d+1$ index of the Array

Controllers

1. Location

- `create__new`

To take form/submit data and create a new model

- `download__weather` (today and future)

To query the API and populate the database

- `weather`

To get weather for today and into the future

Run with:

```
rails g controller Location create_new download_weather weather
```

```
create  app/controllers/location_controller.rb
route   get 'location/create_new'
        get 'location/download_weather'
        get 'location/weather'
invoke  erb
create  app/views/location
create  app/views/location/create_new.html.erb
create  app/views/location/download_weather.html.erb
create  app/views/location/weather.html.erb
invoke  test_unit
create  test/controllers/location_controller_test.rb
invoke  helper
create  app/helpers/location_helper.rb
invoke  test_unit
```

2. Metric

- by_zipcode

Run with:

```
rails g controller Metric by_zipcode
```

```
create  app/controllers/metric_controller.rb
route   get 'metric/by_zipcode'
invoke  erb
create  app/views/metric
create  app/views/metric/by_zipcode.html.erb
invoke  test_unit
create  test/controllers/metric_controller_test.rb
invoke  helper
create  app/helpers/metric_helper.rb
invoke  test_unit
```