

README

Introduction

Zip Status Board is an Express web application for the servicing of zips. It is a tool which manages zips and their tasks. The Zip Status Board features UI and API functionality.

Getting started

Install dependencies:

```
npm install
```

Dependencies:

```
Express >= v4.16.4  
Handlebars >= v4.0.3  
Mongoose >= v5.4.19  
MongoDB
```

Extra dependencies for development:

```
Nodemon >= v1.18.10  
MongoDB installed locally
```

To run the web app (requires process.env.MONGODB_URL):

```
npm run start
```

To run the web app on dev:

```
npm run dev
```

To run tests:

```
npm run test
```

Important items of note

This project uses ES6 conventions and thus takes advantage of async/await over promises when possible.

This web applicaiton contains a **hardcoded** MongoDB connection with a username and password. This is not proper convention and dangerous! The intention behind this is solely for the rapid spin up of the web app and ease of use!

API

Zips and Task Models share a 1:many relationship. All tasks have a zipNumber which is tied to a Zip model. Deleting a zip will delete all associated tasks. Updating a task **will** affect the health of a zip and so patching a zip's health is **highly** discouraged and will result in undescribed behavior as it is subject to change upon a new task being added for that zip.

Zips

GET '/api/zips'

Returns all zips.

Optional parameters:

- sortBy=[isHealthy, zipNumber]_[asc, desc]
- isHealthy=[true, false]

Responses: 200 - successful and Zip object 500 - error querying DB

GET '/api/zips/'

Returns unique zip.

Parameters:

- None

Responses: 200 - successful and Zip object 404 - no Zip found 500 - error querying DB

POST '/api/zips'

Creates a new zip.

Required parameters:

- body (Zip Model)
 - zipNumber (required - must be a positive integer)

Responses:

- 201 - successful and Zip object
- 400 - error and error object

PATCH '/api/zips/'

Updates unique zip.

Required parameters:

- body
 - isHealthy (required - true or false)

Responses: 200 - successful and Zip object 404 - no Zip found 500 - error querying DB

DELETE '/api/zips/'

Deletes unique zip and associated tasks with zip.

Required parameters:

- None

Responses: 200 - successful and Zip object 404 - no Zip found 500 - error querying DB

Tasks

GET '/api/tasks'

Returns all tasks.

Optional parameters:

- sortBy=[description, isComplete, isBlocking, zipNumber]_[asc, desc]
- isComplete=[true, false]
- isBlocking=[true, false]
- zipNumber=[positive integer]

Responses: 200 - successful and Zip object 500 - error querying DB

GET '/api/tasks/'

Returns unique task.

Parameters:

- None

Responses: 200 - successful and Task object 404 - no Task found 500 - error querying DB

POST '/api/tasks'

Creates a new task.

Required parameters:

- body (Task Model)
 - zipNumber (required - must be a positive integer AND valid zip)
 - description (required)
 - isBlocking (required - true or false)
 - isComplete (optional - true or false (default))

Responses:

- 201 - successful and Zip object
- 400 - error and error object

PATCH '/api/tasks/'

Updates unique task and will update associated zip's health.

Required parameters:

- body
 - isComplete (required - true or false)

Responses: 200 - successful and Zip object 404 - no Zip found 500 - error querying DB

DELETE '/api/tasks/'

Deletes unique task.

Required parameters:

- None

Responses: 200 - successful and Zip object 404 - no Zip found 500 - error querying DB

File structure

```
config/
DB URL for dev and test environment configured here! (MONGODB_URL)

node_modules/
Contains all package dependencies.

public/
Contains all static public files (css/js) for client-side.
  css/
    styles.css contains CSS for all webpages to keep things compact.
  js/
    Each .js file has a .hbs counterpart in the templates/views directory.

src/
Contains all server-side handling and rendering.
  app.js, index.js
    Contains web application logic for Zip Status Board.
  db/
    mongoose.js handles the MongoDB connection using the Mongoose package.
  models/
    Defines the data models Zip and Task.
  routers/
    Contains all route logic for the web application (file structure reflects
route URLs).
    /api
      Router logic for the API.
    doc.js, fleet.js, maintenance.js
      Router logic for the UI.
  templates/
    Contains all the UI template files. Zip Status Board uses Handlebars as its
view engine therefore all files use the .hbs extension rather than the standard .html.
    /partials
      header.hbs controls the header displayed on every page. The title is
rendered from router .js files.
    /views
      Each file has a counterpart view on the website. Dynamic data is rendered
through client .js file counterparts.
    /tests
      Contains tests associated with the API

package.json
Configuration file for NodeJS.
```

People

Created by Drew Taylor, March 2019.