

1. config - default meanJS config files
2. modules - contains "chunks" of functionality for the app.
  1. articles - sample CRUD module interacting with a db. unused.
  2. chat - live chat functionality. unused.
  3. core - the actual application
    1. client - client-facing pages.
      1. app - main angularjs initialization code. unchanged.
      2. config - maps urls to views, controls the menu. unchanged outside of disabling some unused menu options
      3. controllers - angularjs codebehind javascript pages for the app
        1. home.client.controller.js - contains the main javascript logic for the app.
      4. css - styling for the app. unmodified.
      5. directives - some helper functions. unmodified.
      6. img - image resources for the app
      7. services - some angular goop that controls menus, error handling, etc. unmodified.
      8. views - actual html pages for the app. also contains 404 page, etc.
        1. header.client.view.html - partial html for the header bar. contains menus that we aren't using.
        2. home.client.view.html - the html for the home page. contains the view logic for the app.
    2. server - server side functionality, e.g. web services, db access
      1. controllers - provide server side functionality
        1. core.server.controller.js - contains most of the server-side functionality for the app. particularly, db access is here.
      2. routes - maps urls to nodejs functions
        1. core.server.routes.js - maps url GET requests to functions in core.server.controller.js
      3. views - some server side error pages
    4. users - contains currently unused logic for authentication. could be turned back on fairly easily.
  3. mongo-seed - stuff for seeding the database with data
    1. Dockerfile - for docker, tells the db to load the data
    2. mazips.json - the json data to load into mongodb.
  4. public - staging area that node maps to the actual web directory when hosted.
  5. scripts - miscellaneous, unused scripts. mostly for authentication.
  6. app.json - describes the web application for nodejs (our server solution)
  7. Dockerfile - app build file
  8. docker-compose.yml - docker-compose file, entry point to run in docker
  9. gulpfile - gulp file, which is our build system. provides realtime reloading, lint, etc.
  10. karma.conf.js - test runner config file
  11. LICENSE.md - MIT License.
  12. manifest.yml - file used by heroku, if you feel inclined to host this there (it was, early on.)
  13. MAZips.csv - original raw data file, before it was converted to json
  14. package-lock.json, package.json - contains the node packages this app uses
  15. Procfile - tells heroku what to do to start up the application (in this case, npm start)
  16. protractor.conf.js - integration testing framework config
  17. README.md - readme file
  18. server.js - start point of the application. not very interesting, just loads /config and calls app.start()
  19. test.js - start point for testing the application. we're not running unit tests at the moment.