What is defined in the controllers

Coordinates server- In the coordinates server the function requests the body of the address and feeds it to the google api to determine if the address listing is valid or not. If the address is invalid, it’ll throw an error otherwise it will determine the address on the map based on the address from the listing and save it to the results on a success.

Listings server- In the listings server the function first creates a list, instantiates the listing, and saves the coordinates based on the results from the coordinates server. From there, we have functions defined that will show the current listing, update a listing, delete a listing, retrieve the listings and sort them alphabetically and finally find a listing by its ID. If any error occurs in any of the functions, the function will throw a 404 error.

how the router makes use of the controllers to determine the flow of request handling

There are two different method calls that are based on the which request handler will be used. The first router method call contains get which gets a listing (in this case, all of the listings) and post which creates a listing, routing through the root route. When either get or post request handlers are called, the program chooses the first router method call. The second router method call is based on the listing ID that is specified. Once the listing ID is found, the routing method will allow it to get, update, or delete a listing. The method call router.param sorts out which routing function will be used based on the listingID.

how middleware is used throughout the application to modularize the code

Middleware was used many times in the application to simplify code. I noticed that most of the middleware was used when action was required or for routing to perform an action such as post, get, or delete. Middleware makes the coding easier by acting as a glue between the kernel and application. It allows for easier implementation when it comes to input and output.