Chris Loomis

04/24/2016

CS496 Assignment 3 Part 2

Note - I am using NodeJS with Heroku. Free Heroku apps go to sleep if inactive for 30mins. Please be patient on your first load as it is most likely asleep and will load. Also, free apps with Heroku can only be active 18 hours a day, not that I think this will pose any problems.

Inside server.js you will see the api structure.

- /api/units
  - GET – fetches all units
  - POST – adds new unit contained in JSON payload
- /api/weapons
  - GET – fetches all weapons
  - POST – adds new unit contained in JSON payload
- /api/units/:unit_id
  - GET – fetches unit with _id
  - PUT – updates unit with _id, new data contained in JSON payload
  - DELETE – deletes unit with _id
- /api/units/:unit_id/weapon
  - PUT – supposed to add weapon reference to unit (more on this later)

I met some RESTful constraints, but not all. This certainly operates client-server, meaning the server will take commands from a client and execute them if they are good and valid. It is stateless, as the server does not tack where or what the client is commanding. My server responses are not explicitly labeled as cacheable or non-cachable. There are not many layers, but the client has no idea what or how my server is generating the responses. There is no uniform interface as my server only responds in JSON and it gives no information on parents, children, and siblings, so one could not navigate my server by just making requests. They would need to have knowledge of it before hand.

In the file test.txt you will find test cases. They are split by the HTTP verbs. I tested using Advanced REST Client which is a Google Chrome app. There are JSON for payloads where appropriate. The contents of test.txt is as follows:

GET==========

https://serene-depths-48279.herokuapp.com/api/units

https://serene-depths-48279.herokuapp.com/api/weapons

https://serene-depths-48279.herokuapp.com/api/units/571d09c8909a779439b05bef

POST==========

https://serene-depths-48279.herokuapp.com/api/units

{"faction":"UCM", "name":"Rapier", "a":10, "mv":4, "dp":1, "pts":45}

POST=FAIL==========

https://serene-depths-48279.herokuapp.com/api/units

{"faction":"", "name":"Rapier", "a":10, "mv":4, "dp":1, "pts":45}

{"faction":"UCM", "name":"", "a":10, "mv":4, "dp":1, "pts":45}

{"faction":"UCM", "name":"Rapier", "a":0, "mv":4, "dp":1, "pts":45}

{"faction":"UCM", "name":"Rapier", "a":11, "mv":4, "dp":1, "pts":45}

{"faction":"UCM", "name":"Rapier", "a":10, "mv":-1, "dp":1, "pts":45}

{"faction":"UCM", "name":"Rapier", "a":10, "mv":4, "dp":0, "pts":45}

{"faction":"UCM", "name":"Rapier", "a":10, "mv":4, "dp":1, "pts":-1}

PUT==========

https://serene-depths-48279.herokuapp.com/api/units/571d3a03b4cdb0030049d156

{"faction":"UCM", "name":"Rapier", "a":11, "mv":4, "dp":1, "pts":45}

PUT=FAIL==========

https://serene-depths-48279.herokuapp.com/api/units/571d3a03b4cdb0030049d156/weapon

{"unit_id":"571d33e5f2f69bb4388a039b","weapon_id":"571cffb008d150141a183808"}

DELETE==========

https://serene-depths-48279.herokuapp.com/api/units/571d3a03b4cdb0030049d156

The POST fails that I marked in blue failed due to the fringe cases. They will report a 400 Bad Request, as they are either missing data or the data inside is bad (example - ‘a’ needs to be between 1 and 10).

The PUT fail marked in red is failing due to my inability at this time to get a proper array of references working with Mongo. This means that the two objects cannot reference one another.

Both schemas have changed. This is due to time. Unit is mostly there, I dropped squad size as I realized I was running out of time and decided it wasn't absolutely necessary for the requirements. Weapon is stripped just to name, as I spent too long fussing with Unit.

I tackled this project completely wrong. I got too excited about learning Mongo and Angular and spent way too much time making a front end rather than focusing on the RESTful API. This is why I cannot properly reference a Weapon to a Unit. I did manage to learn a lot about Angular and you can view my endeavors at https://serene-depths-48279.herokuapp.com/ though it is a bit of a mess as pressure increased due to time running out I started getting messier with my coding.

If I had to do it over again I would do my best to focus on the assignment requirements and not what I wish they were. I definitely feel silly, as a lot of time was spent on things I thought I needed that I later realized I did not after reading the assignment again. Also, changing languages from the past few assignments was probably a bad idea. Ruby on Rails was new, and I ditched it to study the MEAN stack. I really like what I learned and am itching to get back to it (though I should sticker closer to what my classes are actually teaching me and asking for). So overall I am disappointed with myself in relation to the task set me. I am fully confident that if I had focused on making an API and not the front end for a web site I would have met all the criteria.