Test Driven Development Roadmap

When is TDD Occurring?

TDD is occurring at backend API. API is written in Golang. All project packages are being tested.

Project packages (top to bottom):

> **Server** - Responsible for validating query parameters, routing requests to appropriate controllers.
>
> **Controllers** – Receives call-backs from the server, handles request logic, manipulates payloads.
>
> **Database -** Returns data from the database, only interfaced via the controller.

The tests are only occurring at the controller level. However, the tests at the controller level hit the entire stack.

Since the backend is just a REST API, the unit test is setup to interface with the stack using http requests. That way the entire chain (Request -> Server -> Controller -> Database -> Controller -> Server -> Response) can be tested sequentially with a single test function per endpoint (multiple tests are performed in a single test function for the entire chain). Only moving to the next package when the previous package tests are exhausted. This works because, the tests are written per endpoint basis and each endpoint has functions written specifically just for it.

For each endpoint, we always test the same sequence. Starting with the server package. Since each endpoint has a different configuration (method, payload, query params, expected response payload and status), it needs an exhaustive test.

We then test the controller. Such as checking payload contents, responding with appropriate http status.

Each controller calls specific database functions written just for it; Therefore, testing database functions on their own would be unnecessary code duplication. Database package is simply tested by passing optional query parameters and payloads to see if SQL statements are behaving as expected. Since we create test data at the beginning of each test, we know what is inside the database and what is supposed to be returned.

Once data is returned from the database package, the controller checks if anything is found (if necessary) and returns appropriate response back to the server.

Because each endpoint has an expected response payload and status configured, if the wrong payload is returned from the controller that is supposed to handle that endpoint, it will panic.

After the entire chain is cleared, the response is returned.

Testing library used: https://github.com/smartystreets/goconvey

When is TDD Not Occurring?

No unit tests will be ran on JavaScript or front end code.

What tests are occurring?

Controller:

- **TestMain –** Tests the connection to the database and migration scripts, sets up database and the server and passes them to further tests.
- **TestGetMedicalData –** Tests query parameter validity. Filtering by description and code, minimum and maximum prices, proximity, longitude and latitude, pagination, as well as combinations of said parameters.

How will the tests be monitored?

Tests will be run after every feature is added, to continually check correctness of the program. Tests will be satisfied when there is a 100% success rate of the tests. The tests are running every time there is a pull request to master. There is an automatic Github action setup just for that.