

Testing

Testing is done with Jest and Supertest

Setup

Ensure you have the connection parameters for "test": filled out in

`./server/app/config/config.json`

It should look something like:

```
"test": {
  "username": "dbadmin",
  "password": "abc123",
  "database": "sayyara_test",
  "port": 12345,
  "host": "mysql-host.com",
  "dialect": "mysql",
  "dialectOptions": {
    "ssl": {
      "require": true,
      "rejectUnauthorized": false
    }
  }
}
```

Check out the database setup PDF for more details.

Running Tests

Run `npm run test` in the root directory to run all the backend tests. There are currently no tests for the frontend.

If successful, the following should be visible in the terminal:

```
> sayyara_server@1.0.0 test
> npx cross-env NODE_ENV=test jest --detectOpenHandles --no-cache --silent

PASS tests/unit/shop.test.js
PASS tests/unit/vehicle.test.js
PASS tests/unit/rating.test.js
PASS tests/unit/service.test.js

Test Suites: 4 passed, 4 total
Tests:      8 passed, 8 total
Snapshots:  0 total
Time:       7.531 s
```

Additional Tests

New tests for the backend can simply be placed in `./server/tests` and Jest will recursively go through all file/subdirectories during testing.

Frontend does not have a testing infrastructure setup, but a similar strategy can be applied as well. For ease of use, you can append the frontend test runner scripts to `"test":` in `./package.json` to something like

```
"test": "cd server && npm run test:setup && npm run test && cd .. && cd client && npm run test"
```

GitHub Actions

These test are automatically run after a push/pull. The GitHub Actions configs can be found in `./.github/workflows/testing.yml`.