

BACKEND TEST

1. CONTEXT

The idea behind this test is to create two services in order to test your knowledge with Go, REST API, security and optimization.

- One service (S1) will be connected to a mongo database and will provide an API to store a record.
- The second service (S2) can read multiple csv files and push it to S1.

2. DATA

3 CSV files will be provided. Those CSV contain raw data. We want to save the exact same data to the database except that we will round the float value with two decimals.

3. SERVICE S1

Server address: <http://localhost:5555>

- Should connect to a local mongo database with a collection “records”
- Should have one endpoint that take the raw data (look at csv structure) through this endpoint <https://localhost:5555/api/records>
- Should check that the float value and the date given is valid or return an error
- Should provide an endpoint to get a record by uuid
- Should provide an endpoint to update a record by incrementing the value with an int given in parameter
- Should provide an endpoint to delete a record Only S2 can talk with this service

4. SERVICE S2

Server address: <http://localhost:4444>

- Should read all CSV files from a path given in a config file and for each csv entry, push the data to S1
- Should be able to scale to thousands of files if needed

5. DEVOPS

Use docker for those two services

6. TESTS

Add unit tests where you think it's necessary

7. DELIVERABLE

Give a repo with the code and a readme to start the services

8. NOTE

If you have any suggestion to improve this test, feel free to write it. If you have enough time, you can even start improving the services with new features and security. Impress us!