Golang test task - adjoe

You have done an awesome job in your 1st interview :-)
We designed this case study to get to know you and your way of working a little better.

There are some requirements to install on your machine for the task:
- Docker
- Docker compose

make
- go (1.11.7 or newer, see golang.org)

Checkout the git repo to prepare your dev environment:
https://github.com/adjoeio/go-test-task

Please read the README to know how to get started.

After you started the dev environment with docker compose, you have the following infrastructure:
- MySQL Server v8.0 binded to 127.0.0.1:3306: user, pass and db name = adjoe (mysql -h 127.0.0.1 -P 3306 -u adjoe -padjoe adjoe)
- running localstack with mocked SQS available under http://localhost:4566
- a SQS queue with the url http://localhost:4566/queue/my-queue
- you can execute aws sqs commands on your localstack with:

```
make aws-cli sqs list-queues
```

Go backend test task:
1. Build an application which stores every 3 seconds a new entity named TodoItem with the properties id, description and dueDate into the MySQL adjoe database. Use the SQL or ORM library of your choice.
2. Add an HTTP server to your application which serves the last written TodoItem under the path /todo. The response body should be encoded in json. Again, use the http library of your choice.

3. In addition to the MySQL persistence, write all TodoItems to the SQS queue my-queue, too. This queue should be created with your dev environment. Use the official aws go sdk to accomplish this.
4. Implement a second application, which consumes the TodoItems from the queue and logs the items to the console. If you have enough time, you could also use a go-routine which is more fun!

Proper logging, configuration management, abstraction layers, dependency injection and project layout is a plus.

Please do not fork our push your results to a public github repository. We would love not to find any results online, to provide every applicant the same conditions. A zip via mail is totally fine.