Java/Kotlin task

Endpoints monitoring service

The task is to create a REST API JSON Java microservice which allows you to monitor particular http/https URLs.

The service should allow you to

- create, edit and delete monitored URLs and list them for a particular user (CRUD),
- monitor URLs in the background and log status codes + the returned payload,
- and to list the last 10 monitored results for each particular monitored URL.

Data model (*just a suggestion for the data model*)**:**

MonitoredEndpoint:

- **id**: (*any type you want*)
- **name**: String
- **url**: String
- **date of creation**: DateTime
- **date of last check**: DateTime
- **monitored interval**: Integer (*in seconds)*
- **owner**: User

MonitoringResult:

- **id**: (*any type you want)*
- **date of check**: DateTime
- **returned http status code**: Integer
- **returned payload**: String
- **monitoredEndpointId**: MonitoredEndpoint

User (*CRUD isn't necessary, it's perfectly fine to have it seeded in the database*):

- **id**: (*any type you want*)
- **username**: String
- **email**: String
- **access token** : UUID like String

applifting.cz
info@applifting.cz

Filip Kirschner | COO
+420 739 627 797

What's expected:

```
{
  name: "Applifting",
  email: "info@applifting.cz",
  accessToken: "93f39e2f-80de-4033-99ee-249d92736a25"
},
{
  name: "Batman",
  email: "batman@example.com",
  accessToken: "dcb20f8a-5657-4f1b-9f7f-ce65739b359e"
}
```

In detail:

- design REST endpoints for the management of **MonitoredEndpoints** (*if you don't know how, don't hesitate to ask us!*)
- monitor endpoints in the background and create **MonitoringResult**
- implement an endpoint for getting **MonitoringResult**
- implement a microservice created in *Java,* ideally written in **Spring Boot**. Use **MySQL** for the database. Use **Spring MVC** as a REST framework
- authentication: do it in the HTTP header according to your choice, you will get the accessToken in it
- authorization: a **User** can see only **MonitoredEndpoints** and **Result** for him/herself only (*according to **accessToken***)
- don't forget model validations (*you* decide what's necessary to validate)
- write basic tests in **JUnit** or **TestNG**
- push everything into a public repo on **GitHub**
- create a readme file where you explain how to start and use the service
- send us a link to the **GitHub**
- **bonus points**: create a **Dockerfile**, add docker-compose and describe how to start and run it in **Docker**

Alright, that's it. Good luck and, most importantly, **have fun!**

applifting.cz
info@applifting.cz
Filip Kirschner | COO
+420 739 627 797