

Coding assignment for back-end engineer

Dear candidate,

Thank you in advance for taking your time to complete our coding assignment.

You will have up to 5 days to complete the test. However, we would appreciate if you could complete the test as soon as possible, since we will have to review the result and only thereafter move forward in our recruitment process.



Purpose

Main purpose of this assignment is to test practical skills of designing, coding, testing and building java applications.

Prerequisites

Basic java knowledge, along with experience in OOP, API design, testing frameworks, and understanding of basic algorithms and data structures is a necessary prerequisite for taking this test.

Assignment

We ask you to implement an application which represents a store for car charging session entities. It will hold all records in memory and provide REST API.

Each entity of the store represents unique charging session that can be in progress or finished. Entity can have the following fields:

```
UUID id;  
String stationId;  
LocalDateTime startedAt;  
LocalDateTime stoppedAt;  
StatusEnum status;
```

Endpoints to implement

| Endpoint | Description | Computational complexity (upper bound) | Request body | Response body |
|----------------------------|---|--|---|---|
| POST /chargingSessions | Submit a new charging session for the station | $O(\log(n))$ | <pre>{ "stationId": "ABC-12345" }</pre> | <pre>{ "id": "d9bb7458-d5d9-4de7-87f7-7f39edd51d18", "stationId": "ABC-12345", "startedAt": "2019-05-06T19:00:20.529", "status": "IN_PROGRESS" }</pre> |
| PUT /chargingSessions/{id} | Stop charging session | $O(\log(n))$ | | <pre>{ "id": "d9bb7458-d5d9-4de7-87f7-7f39edd51d18", "stationId": "ABC-12345", "startedAt": "2019-05-06T21:15:01.198", "stoppedAt": "2019-05-06T21:17:01.198", "status": "FINISHED" }</pre> |

| | | | | |
|-------------------------------|--|-----------|--|---|
| GET /chargingSessions | Retrieve all charging sessions | O(n) | | <pre>[{ "id": "d9bb7458-d5d9-4de7-87f7-7f39edd51d18", "stationId": "ABC-12345", "startedAt": "2019-05-06T19:00:20.529", "stoppedAt": "2019-05-06T21:17:01.198", "status": "IN_PROGRESS" }, { "id": "69acaf1b-743f-47df-9339-abe50998b201", "stationId": "ABC-12346", "startedAt": "2019-05-06T19:01:35.245" , "stoppedAt": "2019-05-06T21:17:01.198", "status": "FINISHED" }]</pre> |
| GET /chargingSessions/summary | Retrieve a summary of submitted charging sessions including: <ul style="list-style-type: none"> ▪ <i>totalCount</i> – total number of charging session updates for the last minute ▪ <i>startedCount</i> – total number of started charging sessions for the last minute ▪ <i>stoppedCount</i> – total number of stopped charging sessions for the last minute | O(log(n)) | | <pre>{ "totalCount": 5, "startedCount": 1 "stoppedCount": 4 }</pre> |

Requirements

1. Implementation should be done in Java 8, but feel free to use any libraries or frameworks you want
2. Application is thread-safe
3. Application is covered with tests (classes and endpoints)
4. Application is using in-memory data structures (not to be confused with in-memory databases)
5. Computational complexity meets our requirements (see the table). Limits are only applicable to the data structure which holds charging session objects (serialization, object mappings and other parts of application logic are out of consideration and may have arbitrary complexity)
6. Documentation of the implemented functionality and instructions how to run are present (consider adding javadocs and README file). We expect it to be run with a single command

It is important that your application meets all the requirements for successfully passing the assignment.

Out of scope

1. Space complexity
2. Data store limit considerations (we make an assumption that total amount of charging sessions will never exceed 2^{30})

Please send your solution in ZIP archive via email.

We hope you will enjoy this assignment and should you have any questions – please reach out to us 😊

Thanks again for your interest!