



Mercedes-Benz

Software Engineer Quiz

Thank you very much for filling out this quiz to the best of your competencies. The purpose of this quiz is to evaluate your areas of expertise.

If a question is not clear to you or using notations which you are not familiar with, feel free to rephrase the question or lay assumptions which help you in solving the problem.

Good luck and thank you again for your time!

1. Star Wars Web Application

The goal of this exercise is to develop a Web application to display data coming from a third-party API.

AS A Star Wars fan

I WANT to see the information about PEOPLE and STARSHIPS being able to sort the information and search by too **SO THAT** I can have fun searching Star Wars information.

Acceptance criteria

- Web application to display information from [SWAPI](https://swapi.co/) (<https://swapi.co/>).
- The Web application should read and display information in a web browser, from the API endpoints:
 - People
 - Starships
- The Web application should allow to sort the data by "name" and "created" fields, in descending and ascending order, both cases. The mechanism for sorting should be designed following the "Open-closed" principle.
- Visualize the data using table format.
- Develop this solution using Java.
- Provide a Dockerfile to be able to deploy the application as a Docker container on port 6969.
- Use of JavaScript for sorting is not allowed.

Definition of Done

- The application works as defined in the Acceptance criteria.
- Provide source code of the web application.
- Provide a clear README file about how to run the application, from scratch, as a Docker container.
- Integration tests green.

Extra points

The requirements below are not mandatory. The fact of not doing them will not discard you from the hiring process neither reduce points. In case you decide to implement these requirements, they must work as expected, otherwise the ones not working will not be considered.

- Use Gradle as building tool.
- Provide the Kubernetes resources to deploy the application as a service.

We will evaluate

- Follow Software Engineering principles.
- API performance.
- Clean code.
- Usage of Docker.

2. Message Broker

The goal of this exercise is to implement a CLI tool that sends a protobuf message (<https://developers.google.com/protocol-buffers>) to a message broker.

AS A Software Engineer

I WANT a tool that is able to send information in protobuf format to a message broker system **SO THAT** other distributed systems can read that information for their purposes.

Acceptance criteria

The CLI tool should read a JSON file, indicated as parameter, where the information to send is stored.

The information from that JSON file will be encapsulated in a protobuf message that will be sent to the message broker. Develop this tool using some of the languages below:

- Java
- Python

The message broker should be one of the list below:

- Kafka
- RabbitMQ

Definition of Done

- The message written into the message broker is readable from a consumer.
- Provide source code of the tool with clear instructions about how to run it.
- Provide exhaustive documentation about how to run the tool and read the messages from the message broker.

We will evaluate

- Follow Software Engineering principles.
- Clean code.

Appendix I

Protobuf message to be used in the exercise

```
message Person {
  required string name = 1;
  required int32 id = 2;
  optional string email = 3;
}
```