Specifications for the introduction day

Welcome!

On the following pages you will find the specifications for the introduction day.
The application is a blog rest API.

Please note the following:

- Only the Go standard libs may be used
- No MVC frameworks are permitted
- Dependencies can be drawn via packet management software
- The total scope of the application deliberately exceeds the available time
- Quality is more important to us than quantity

If you have any questions or problems, please do not hesitate to contact us. We wish you every success and look forward to your results!

Table of contents

Introduction

You are no doubt familiar with the basic functions of a blog. We will build on this basic understanding. Please write everything that has to do with your code in English, regardless of whether it has to do with program instructions or inline documentation. The database tables and fields should also be named in English.

Backend

Database and data model

All data must be stored in a MySQL database. You must create the relational data model for this based on the requirements. Pay attention to structured naming.

The database can be accessed using the following access data:

IP: 172.29.24.51
User: applicant
PW: check24.de

API

A rest API is to be provided via a web service which can be used to maintain the blog. The web service should be built in such a way that it can process approx. 100 requests in parallel.

API methods

The interface should have the following API methods to maintain data records:

- Creating or inserting blog entries
- Creating or inserting comments on blog entries
- Read previously saved data records (blog entries)
    - The following filters are possible
        - Sorting
        - Number
        - Text length
    - The following information is displayed in the entry overview
        - Date, author, URL to the image, text (see filter above) and number of comments
- Output of a complete entry with
    - Date, text, author, comment(s), URL to the image
- Updating existing data records
- Individual deletion of blog entries

The interfaces must be defined according to the use case. Input and output must be described.

JSON is to be used as the data format.

Data record

Each blog entry consists of at least the following values:

- Title
- Creation or editing date
- author
- text
- Link to image (optional)

Each comment contains at least the following values:

- Name
- E-mail address (optional)
- URL (optional)
- Comment text

Security

Requests or calls to the web service are only possible via authentication.

Furthermore, all security-relevant aspects for the web service must be observed.

Output

Corresponding responses and status codes are returned for the respective requests to the web service.

Unit tests

At least 2 unit tests are created to check the functions created.

Further notes

- If you have to program something "unclean", but know how to implement it more cleanly, then simply write a comment in the relevant place
- If you have any questions, please ask!

Bonus task

If you have already completed all the tasks and the code is complete,
then there is the following bonus task:

Please test your web service with Jmeter or another load testing tool to make sure that the 1000 Rq/s are possible.