Hello! Thank you for your interest in this position. Before the final interview, we would like to see an example of your work.

Instructions

1. This is the project you will use: https://drive.google.com/file/d/1-fcpet_CF8IpLdnpt4qPaH1F22yC4T3Q/view
2. Please, add this project to your GitHub account with the name "LAST_NAME_be_FIRST_NAME_exercise", and start developing from there.
3. Pay attention to the Main Tasks, must-have items and what we will be evaluating on your PR, as described below. The must-have items are **eliminatory**, and the other items will complement the challenge evaluation.
4. Once you are ready to submit, please create a **public PR** on your repository with all the changes done.
5. Return this PR by the time indicated in the email you received this challenge.
6. Although some items in the code challenge are open to interpretation by the candidate, you should feel free to contact us with any questions you may have.
7. Deliver the most you can with the time you have, everything will be analyzed.

Project Description

This project is a Java SpringBoot microservice that spins up a REST API, called Roles API, to improve upon the capabilities of the Users API and the Teams API to manage memberships between them. The project is covered by unit tests and API tests.

The microservice is exposed via a REST Api. For more info check-out the exposed OpenApi (checkout the README for more info) under
<host>:<port>/swagger-ui.html

Main tasks

The code you are receiving may contain flaws such as: code smells, bad practices, bad/wrong/missing tests, bad/missing validations, and so on.

One of the tasks is to find the problems in the code. Fix them as soon as you find them, and describe the error with the correction in the PR description. For issues you don't have time to fix, mention them in the PR description with the fix you would like to have implemented.

The other task is to implement a new API endpoint where the user can get a Role using userId and teamId as filters. **Remember to add Unit Tests and API Tests for it**. **Add a comment in your PR explaining what was your approach for the new API endpoint**.

Must have

- A **public PR** containing all the changes;
- Your **public PR** should contain a detailed description:
  - You should mention and justify every change you did on the code;
  - You should describe your approach to implement the new endpoint;
- You should implement **tests** for the new endpoint;
- The entire code, README, and any supporting documentation should be written in English.

We will look for experience and good practices with:

- **Tests!**
- Code review;
- Project patterns and clean code;
- Business validations;
- Implementing a new feature following the existing project standard;
- Creating a PR.