

Back-end developer exercise

Please see the exercise. You have 48 hours to do this test so please return it by the time indicated on the email.

Once you are ready to submit, please post your work to a private GitHub repository and send me the link or just send me the files in a zip directly.

Please follow the instructions, they are vague to allow for your interpretation. However, be sure to add tests and validation to your coding exercise.

Please do not hesitate to be in contact if you want any clarification or ask questions.

Introduction

Take a look at the following REST endpoints:

<https://cgjresszgg.execute-api.eu-west-1.amazonaws.com/users/>

<https://cgjresszgg.execute-api.eu-west-1.amazonaws.com/teams/>

Append the IDs of users and teams to get further information about each entity, e.g.

<https://cgjresszgg.execute-api.eu-west-1.amazonaws.com/users/02095fb1-0e3d-469e-922c-0b2f9f58635d>

<https://cgjresszgg.execute-api.eu-west-1.amazonaws.com/teams/bdad9afe-63c3-4ff2-ae6b-f1143d92ca15>

This is the Teams REST service and it has a list of users, each user can be part of zero or more teams. Each team has one user as a team lead.

Main task

Enrich this service, by giving us Team Roles and the ability to associate team members with them.

At minimum three roles should be pre-defined: "Developer", "Product Owner", and "Tester".

"Developer" should be the default role.

Must have

- README.md :
- Describe how you approach the problem and/or solution.
- Please provide information on how to run the code

- Code!! github url/tarball/zip/etc
- Assume this is an API you can not change, but if you could improve the Team or User API what would you ask the developers for to aid you with your task.

Recommended

- docker and docker-compose, you should be able to run this up with no manual set up
- **Write tests (unit tests and rest api tests)**
- **Add validation and think about edge cases**
- Avoid logic where it does not belong (think about architecture as if the application was big)
- What happens if data you are using gets deleted
- Java (e.g. Spring Boot)

Backend points for:

- The user of this Roles service should be able to do the following through REST:
- Look up a role for a membership, where membership is defined by a user id and a team id.
- Look up memberships for a role
- Create a new role
- Assign a role to a member