JUCR

Backend Challenge

Hi, welcome to JUCR!🚀

If you read this text, you are one of the lucky ones who have made it to the final stageof our hiring process - Congrats!

You are only a small step away from being part of the most amazing team ever🎇

The Case

- Each service is deployed to a Kubernetes cluster (so we need a **docker image**)
- As JUCR aims to have the highest possible uptime, each service needs to be **scalable by design** (so think about timeouts and other stuff which could go wrong)
- To serve data to several clients, we use a federated GraphQL gateway (so eachservice needs to provide its /graphql endpoint which serves a **sub-graph**)
- We use **MongoDB** as our persistence layer
  - It’s required to use UUIDs (v4) instead of the default ObjectIds internally

The Task

- Create a service which **pulls current charging station data** from Open ChargeMap
  - The service pulls the data and update the own database (when therearechanges!)
- API-Key: *ff82541f-c8d1-4507-be67-bd07e3259c4e*
- This service needs to provide a **/graphql endpoint** which can be queried to list all imported charging stations
- It should be possible to paginate through the list in GraphQL using **relay-style pagination**
- We need **separate docker images** for pulling the data and the service itself
- We need the following fields (from Open Charge Map) to be imported:
  - operatorInfo
  - statusType
  - addressInfo
  - connections
- It should be possible to **run the whole service and the import locally** using a single command (so please provide instructions and a docker-compose file for the additional local infrastructure)
- **Bonus**: Minimum 50% unit test coverage and minimum one E2E test

Information

- Document and show your solution well, don’t show only the result, **show the way** you’ve taken to implement your solution
- Create a **private repository on GitHub** and commit as often as you think it makes sense
- If you’re unsure about what we really need, make an assumption anddocument that assumption
- Try to follow all best practices you know to provide a **clean solution**

What we’re looking for

- Smartness of the solution
- Well structured documentation
- The way you’ve taken to come to your solution

We’re really looking forward to discussing the results with you!