Service Provider Ratings and Notifications

Welcome to Homerun's backend assignment!

Please read the instructions below and don't hesitate to ask questions if you need any clarification on the specs.

Good luck, and have fun!

Introduction

We need to create a service marketplace that enables Customers to rate Service Providers and notifies Service Providers when new ratings are added.

As a Backend Developer, your task is to create and implement REST APIs to be consumed by frontend applications. For this assignment, **no authentication mechanism is required**.

Service specifications

Two Backend API services are required for this project:

Rating Service:

- Provide an endpoint to Customers to submit ratings for Service Providers.
- Provide an endpoint to Customers and Service Providers to get the average rating for a specific Service Provider.
- Data must be persisted on a database.
- Data about new ratings must be sent to the notification service.

Notification Service:

- Provide an endpoint to Service Providers to get the newest notifications about ratings created for them. Assume that this endpoint is polled by Frontend clients to get the latest notifications. The notifications must be returned just once, subsequent calls should not return the same notifications and only return the new ones received after the previous call.
- It’s ok to lose data in case the process crashes or gets restarted.

Definition of Done

- The system works as specified.

- .NET and/or Go are used to implement the services. It's up to you to decide which service should be built using which platform.
- Your main decisions, the maintainability, reliability, and scalability of the solution and possible future improvements are documented.
- The two services must use a communication mechanism such as REST, AMQP, gRPC or Streams to communicate. Data storage mechanisms such as databases or file systems are not allowed as a communication system.
- Include build scripts, Dockerfile, and docker-compose.yml to build and run the application.
- A reasonable number of automatic tests are required.
- All API endpoints must comply with REST principles.
- (Optional) Include OpenAPI which outlines the available endpoints and their functionality.
- (Optional) There is no actual need for a notification service to send notifications. However, for simulation purposes, these notifications can be logged somewhere for debugging and testing.
- (Optional) Use a structured logging solution of your choice.
- (Optional) Using Testcontainers for integration tests is a plus.
- (Optional) Define CI/CD pipelines definition (e.g. using Github Actions.) to check code quality and automate releases.

Implementation hints

- Keep the solution as simple as possible.
- Design the solution to be reliable, scalable, and maintainable. If you decide to trade off quality for simplicity (we don't want you to waste too much time), motivate your choices and propose possible improvements by documenting them.
- The apps that consume the APIs are developed by the frontend team, so you should not worry about it.
- Start describing the list of interactions between the rating service and notification service.
- Feel free to use any library that might help you do the job faster or more conveniently.
- Consider the network to be not reliable.
- Consider that other services in the system rely on the notification service and generate a high volume of requests.
- Ensure compliance with the community coding standards of the preferred development language.

Assignment evaluation

Here's what we'll look into when evaluating your submission (items are sorted randomly and not by importance):

- Requirements understanding.
- System functionality.
- Code quality (simplicity, maintainability, scalability and reliability).
- Tests quality.
- Coding style consistency.
- Documentation (e.g., notes explaining decisions or possible improvements, quality of code comments, setup instructions).
- Git commit messages & content.
- Knowledge of languages/libraries (compared to previous personal experience).

Assignment delivery

Work on your private repository and send us access to it once the work is completed; you can add as collaborators on Github the following users:

https://github.com/ezgipeker
https://github.com/marconota
https://github.com/lorenzoranucci
https://github.com/Blind-Striker

If you want to use a different platform, let us know how we can access it.

The assignment does not have any time constraints (you can return it after a day or in two weeks); in any case, we will consider the time you need to complete the assignment.