

Zefir hiring - backend code test

Explanation

Hi, and welcome to Zefir's backend code test ! The objective of this test is to assess your ability to understand and develop a solid architecture for a nestjs server. You will find in this project a home module with 3 entities inside it :

- a `home` entity, which describes the basic properties of a home (surface in m2 etc.)
- a `businessData` entity, which describes the pricing of a home (initial offer price, final offer price, target sale price etc.), and which has a one to one relation with the home entity
- a `user` entity, which is used to save our internal employees (it's not a proper authentication system, it's just used to save our employees on our database)

The module is divided between `entities` (they represent the table schemas in the sqlite database), `models` (they are used by graphql for our endpoints), `repositories` (they maintain the connection and perform the operations to the sqlite database), `resolvers` (they are our graphql endpoints), and `services` (they perform and hold the business logic).

You will also find helper functions in the shared folder, especially for the tests setup (see <src/shared/test.helper.ts>).

Tasks

1. You must create the following functions for the business data logic (see <src/home/services/businessData.service.ts>, function `generateBusinessDataForHome`) :
 - a function to compute the negotiation margin `computeNegotiationMargin` - it takes 3 parameters : `finalOfferPrice`, `targetSalePrice` and `maxNegociationMargin` (percentage with a default value of 7%) and returns the value of the margin (computed as the minimum between $[\text{targetSalePrice} / \text{finalOfferPrice} - 1]$ and $[\text{maxNegociationMargin}]$)
 - a function to compute the service fees of an offer called `computeServiceFees` - it takes 2 parameters : the `finalOfferPrice` of the home, and the `zipCode` where the home is located (for example '75016', see Annexes for the computation method)
2. You must create the test functions for the new critical business functions you just added.
3. You must refactor the code architecture to make it more coherent. The idea is that the current architecture is not satisfactory - we would like to have something which looks more like a micro-service architecture, with at least 2 modules (you are free to create more if you feel like it makes sense).

4. You must add tests for our resolvers to make sure the functions they provide work properly (see <src/home/resolvers/home.resolver.test.ts> for an example). You must add at least one test for every resolver, but you can add more if you feel the need.

Evaluation

We will evaluate your work based on the quality of your code, the quality of your refactor, and the precision of your tests. Therefore, don't forget that your tests must run :

```
yarn install
yarn test
```

Annexes

Service fees by City:

Lille

- price < 100 000€ : 15 000 €
- 100 000€ <= price < 145 000€ : 19 000€
- 145 000€ <= price < 200 000€ : 20 000€
- 200 000€ <= price < 400 000€ : 10% of the price
- 400 000€ <= price < 650 000€ : 8% of the price
- price >= 600 000€ : 30% of the price

Paris region (75, 92, 93, 94)

- price < 100 000€ : 20 000 €
- 100 000€ <= price < 145 000€ : 22 000€
- 145 000€ <= price < 200 000€ : 23 000€
- 200 000€ <= price < 400 000€ : 11% of the price
- 400 000€ <= price < 650 000€ : 8% of the price
- price >= 600 000€ : 10% of the price

Nantes (44) and Lyon (69)

- price < 100 000€ : 20 000 €
- 100 000€ <= price < 145 000€ : 22 000€
- 145 000€ <= price < 200 000€ : 23 000€
- 200 000€ <= price < 400 000€ : 11% of the price
- 400 000€ <= price < 650 000€ : 8% of the price

- price \geq 600 000€ : 99.9999% of the price