

Fun Academy

Backend developer test

Intro

About the test

Welcome to the Fun Academy backend developer test. In this test we are going to evaluate the candidate skills for the position of backend developer.

The main objective of the test is to design an API for a digital shop. This digital shop will consist of two different parts, one part related to the shop management and another part related to the customer interaction with the shop.

The evaluator will take in consideration how documented is the project, the quality of the code and the justification of every design choice.

If there is any questions, please don't hesitate to send them to fabio@funacademy.fi

Initial considerations

Description of the API

We have a digital shop and we need to design all the elements to work with it.

The digital shop has:

Admin user:

The user that manage the customers and orders. It should be pre-created on the system(no need of registration) .

Customers:

The customers are user that can register on the shop and buy things. There will be three different type of customers: retailers, providers and website users.

Retailers:

The retailers are users that can have a related provider or not. If they buy having a provider, they will have some discounts associated to the provider and they will be able to buy only five products with each related provider. When they register, they will submit a form with the following fields:

```
email(required)
password(required)
name(required)
retailerResponsibleName(string,required)
phone
mobilePhone(required)
address(required)
```

IBAN(required)
provider
createdAt(required)

The retailer will be inactive until an admin activates it.

Providers:

A provider is an user that can buy items by itself too. The admin of the shop manually assigns discounts to the provider and the provider and all the retailers associated with it will get this discounts when they buy.

When they register they will submit a form with the following fields:

email(required)
password(required)
companyname(required)
phone(required)
IBAN(required)
companyDescription
createdAt(required)

The provider will be inactive until an admin activates it.

Website users:

The website users buy products normaly. When they register they will submit a form with the following fields:

email(required)
password(required)
name(required)
mobilePhone(required)
createdAt(required)

The website user will be inactive until an admin activates it.

Products:

This shop will only sell cherries at 0.77 € each one.

The providers and the retailers associated will have a discount depending of what quantity the shop admin set before.

Orders:

They will have a list of lines with the products, the quantity and the amount, and what is it related customer.

Order status:

It will be 4 different possible status: Pending, proccessing, complete, cancelled. If an order is cancelled, the admin will put some explanation on the observation field

of the order, and a notification email will be sent to the user with it.

If an order is completed, it will send a notification to the user too.

Part 1

Design the register/login API.

The objective of this part is design an API that receives all the registration submits and insert the users on the database. It will take on count the different validations of the information and return a response with success or fail and the reason in case of fail.

The API will check that users doesn't exists in the system before insert them.

This API will let the admin activate the customers and to retrieve all the information about the users and modify it. The admin will be able to modify the provider discounts and the assigned providers of a retailer.

The retailers will be able to add associated Providers and the providers will be able to accept this association request.

The users could be able to login in the API and the API will return a token in case of success.

This token will expire in a minute, then it will be an API route to refresh it.

All of this operations should return a json response if it was necessary reporting the success of the operation.

Part 2

Design the orders API

In this part, we are going to generate the API part that ables the users to send orders and let the admin change it status or delete them. There is a daily limit of the number of products each one could buy. The retailers without provider only can buy a max of 30 products, the providers a max of 50 and the website users, 10 products max.

When a retailer with providers buy there is a limit of 5 daily discounted products for each provider. The shop will distribute the providers discounts trying to maximize the total discount of the order with all the providers that weren't reached the limit for those retailer. If there are no available providers, a normal price will be set for the remaining products. The validated users will have a route to send the order and the admin would be able to change the status or delete it.

Bonus part

Extra functionalities

As a plus bonification, there are some extra functionalities that could be added:

- Documentate the API with an automated tool (like Swagger).
- Use any kind of API query language (like GraphQL) .
- Add a user icon field that receives a file and stores it(local folder or AWS, etc).
- Generate a password recovery system(with email notifications).

Final considerations

Final considerations and info

The purpose of this test is evaluate the candidature looking why and what solutions are taken, and how are they done.

The idea is complete the maximum of the points in the tests, prioritizing the correctness of them, having documented and well tested solutions, etc.

About technology, is recommended use Laravel with lumen.

At the end of the test, the candidate will send a file to fabio@funacademy.fi with all the source code and files required for the test, a document explaining how to install/ deploy it and a summary of information that justify the used design patterns, what kind of technology is used and why, and all the information he thinks it could be valuable for the comprehension of the solution. Try to complete all the parts you can for a better evaluation.

The end of the test will be the 31st of July.