

Node Js interview test

Let's assume we have a payment sub-system. This system is supposed to check whether the given validated invoice can get a discount or not.

In this test we need an endpoint as follows:

Input: product code or product name, the user's ID and the invoice final amount.

Response: If it includes the discount, the discount percentage that can be considered for the given invoice.

Product structure is a graph:

1. A product can have a category as a parent.
2. Every category can have a parent category.

Applying the discount:

If the product has a discount this discount has a priority and will be returned. Otherwise, we move to its parent category (if it exists) and check if it has a discount and so on until the product-category chain is done. If no discount was found the API should return -1.

For simplicity, you can only consider 2 layers of this structure and we have one product from one category, which is this category is another subset of categories. But you're more than welcome to show your talent :)

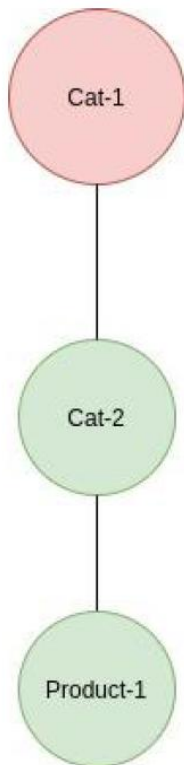


Figure 1: Example of the product category structure.

Notes:

1. Storing the structure is not a priority, you can even hard-code it. But it's preferable to use a database.
2. The data should be structured in a way that editing the discounts should be easy. Although there is no need to create an endpoint for editing the discounts.
3. The service knows what products are included in what categories so it's up to you whether you get the product id or the product name in the endpoint as an input.
4. It's preferable to use Mongo DB.
5. The users are authenticated. However, having good authorization in the system for the companies using the system earns an extra point.
6. The system should be created using nest JS.
7. Having a test unit earns an extra point.
8. Having a swagger earns an extra point.
9. A dockerized system earns an extra point.
10. Good documentation earns an extra point (explaining the system's structure, and future features that you think are important for this system but you didn't have the time to implement it).

Please create a public repository for the test when it's done and email it to us.

Good luck.