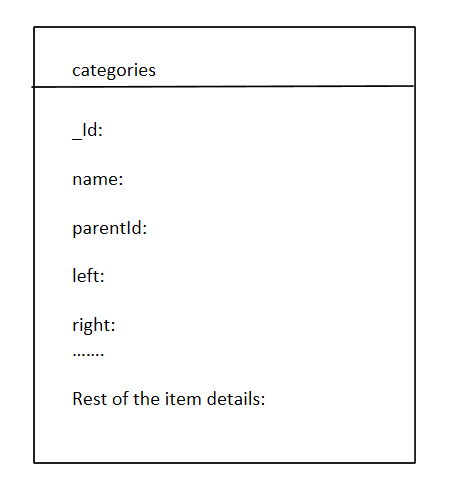
Link to my Git repository

<https://github.com/Maneesha-Keloth/backendtest-cynaxlabs.git>

Q1.

Database structure for categories (Here I am using mongodb for data modelling)



\_id: id of the category document

Name: Name of the category

Left: left index for round trip traversal of the tree from the root

Right: right index for round trip traversal of the tree back to root

Write a query to fetch all the first-level children (categories+items) for a category.

And: databse.categories.find({parentId:”1234”})

Write a query to fetch all the children (categories+items) at all category levels for a category

Ans: var categoryParent= databse. categories .findOne({\_id:1234});

databse. categories.find({ left: { $gt: databaseCategory.left }, right: { $lt: databaseCategory.right } });

Note: This is efficient if the tree is static in nature

Otherwise we can go for a path based approach instead left and right indices.

**Q2.** Product Rest API end points

Create:

Post

https://host:port/products/1234

{

“name”: “Milk”

“price” : 100

}

Update

PATCH

https:// host:port /products/1234

{

“price” : 150

}

Read

GET

https:// host:port/products/1234

1. DELETE

https://host:port/products/1234

From a versioning side we can introduce one more set of API end points without changing the existing ones.

Post

https://host:port/v2/products

{

“id”: 1234

“name”: “Milk”

“price” : 100,

“currency”: “$”

}

1. When we introduce a new set of APIs to accommodate currency field, new methods for CRUD will get added in the new controller(new methods get added w/o touching existing API end points)
2. We will new entity to accommodate Product’s currency code
3. New service layer as well as DAO (repository) also get added.

**Q4.**

Please refer my git hub repository

<https://github.com/Maneesha-Keloth/backendtest-cynaxlabs.git>