

Make sure you read the whole document carefully and follow the guidelines in it.

Preface:

Build a RESTful API that can `/create/read/update/delete` **Product** and **Category** data from a persistence database.

Product Model:

```
{
    productId : xxx,           // Product ID
    productName : xxx,         // Product Name
    qtyPerUnit : xxx,          // Quantity of the Product
    unitPrice : xxx,           // Unit Price of the Product
    unitInStock : xxx,         // Unit in Stock
    discontinued : xxx,        // Boolean (yes/no)
    categoryId : xxx,          // Category ID
}
```

Category Model:

```
{
    categoryId : xxx,          // Category ID
    categoryName : xxx,        // Category Name
}
```

Functionality:

- The API should follow typical RESTful API design patterns.
- The data should be saved in the DB.
- Category ID in product table should be referenced in the category table.
- Provide proper unit tests.
- Provide proper API documents.
- `/create` should create the product and category.
- `/read` should read particular record from the product table (if product has any category then category should be fetched in the response)
- `/readAll` should read all the records from the product table (if product has any category then category should be fetched in the response)
- `/update` should update one particular record of the product
- `/delete` should delete one particular record of the product.

Requirements:

- Write clear documentation on how it's designed and how to run the code.
- Write good in-code comments.
- Write good commit messages.
- An online demo is always welcome.
- Provide proper readme which includes steps to setup the code in any system, API documentation (Postman documentation link is preferred).
- Candidate needs to provide the github link and the candidate has to make his repository private.

Tech stack:

- Use Node.js and any framework.
- Use any DB. NoSQL DB is preferred.

Bonus Points:

- If you are familiar with ES6 standards then it will be a bonus point for you.
- If you can use aggregation for /read query for fetching the data from multiple tables then it would be considered as a bonus point.
- If you follow the good practices of the Node.js for coding standard/folder structure then it would be considered as a bonus point.

What We Care About

Feel free to use any open-source library as you see fit, but remember that we are evaluating your coding skills and problem solving skills.

Here's what you should aim for:

- Good use of current Node.js & API design best practices.
- Good testing approach.
- Extensible code.

NOTE: Candidate should not be able for further rounds if we found plagiarism.