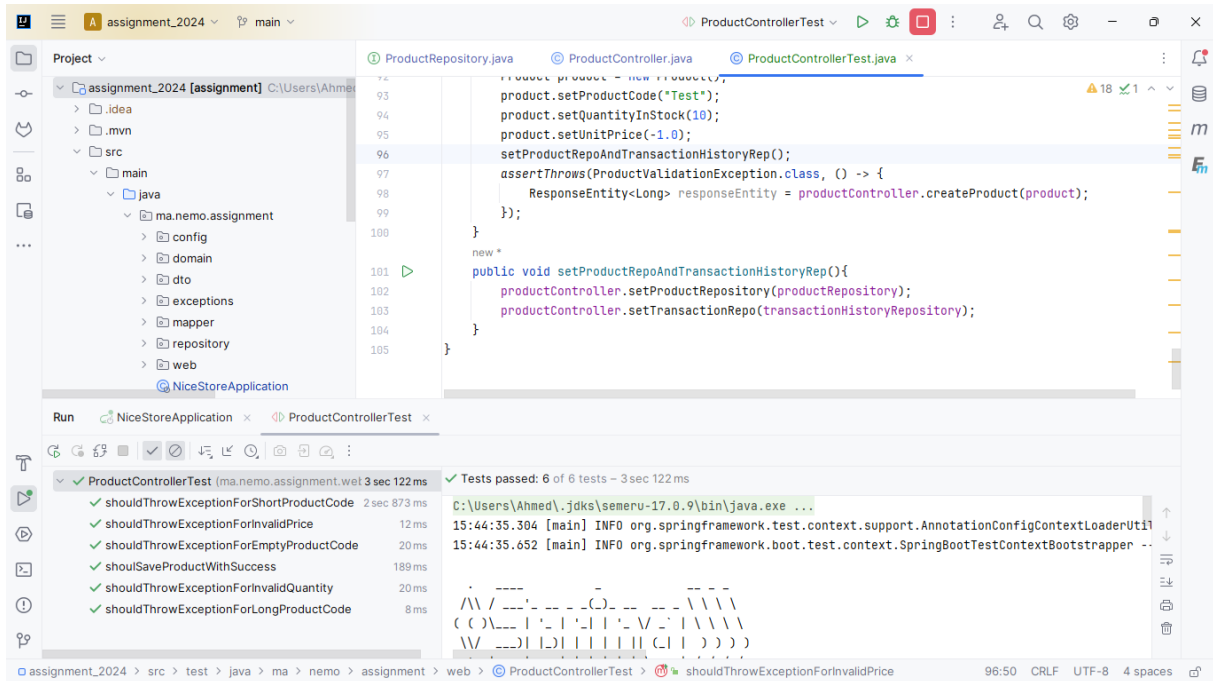


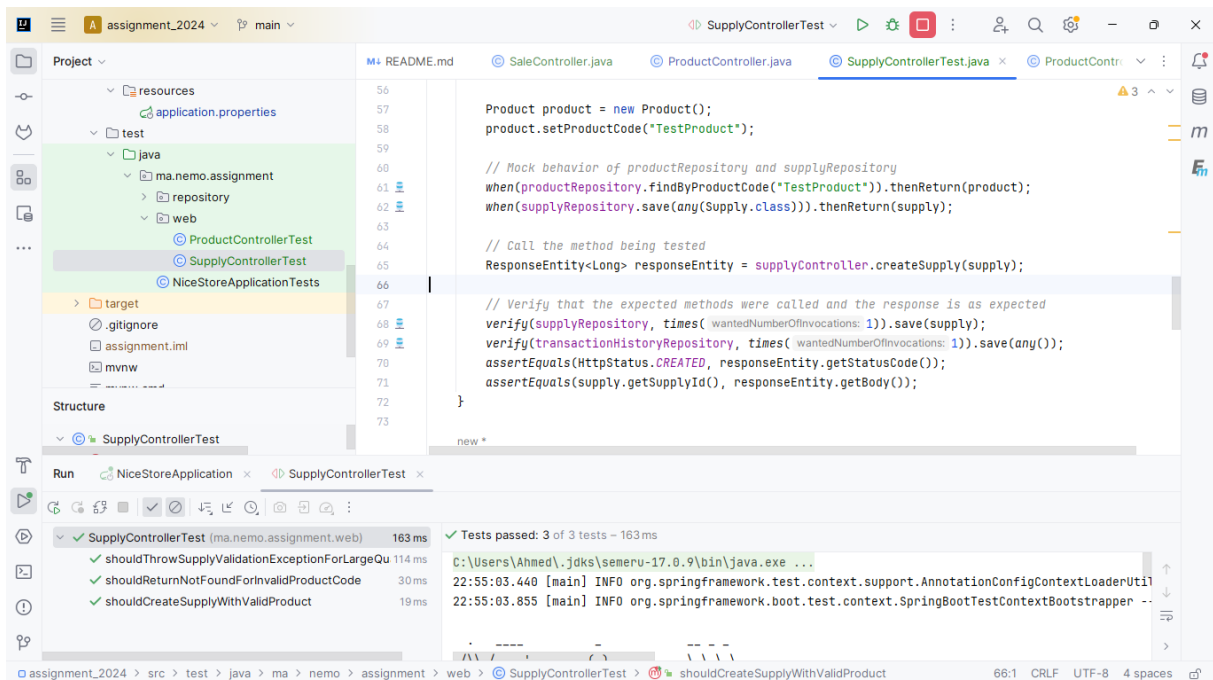
1. Les tests unitaires

a- ProductControllerTest

Le Screenshot suivant montre la phase de test dans laquelle je crée des tests unitaires pour chaque petite exigence de création de produit afin de couvrir un seul scénario dans chaque méthode de test et le faire de manière indépendante.

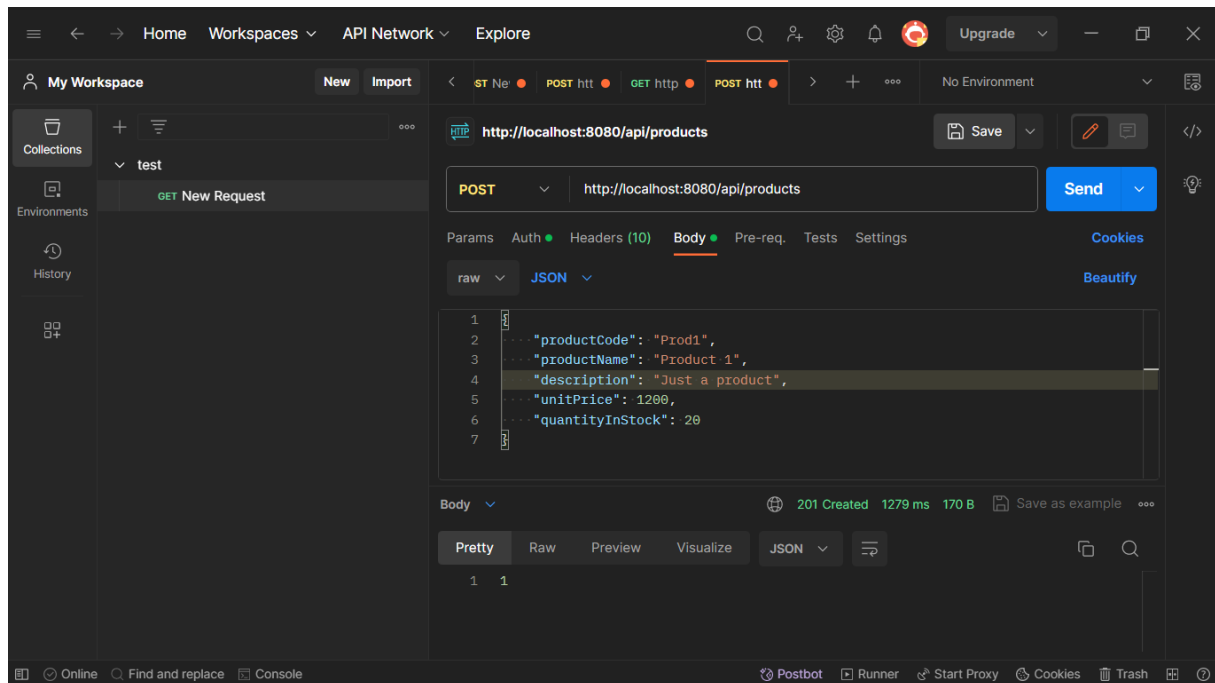


b- SupplyControllerTest

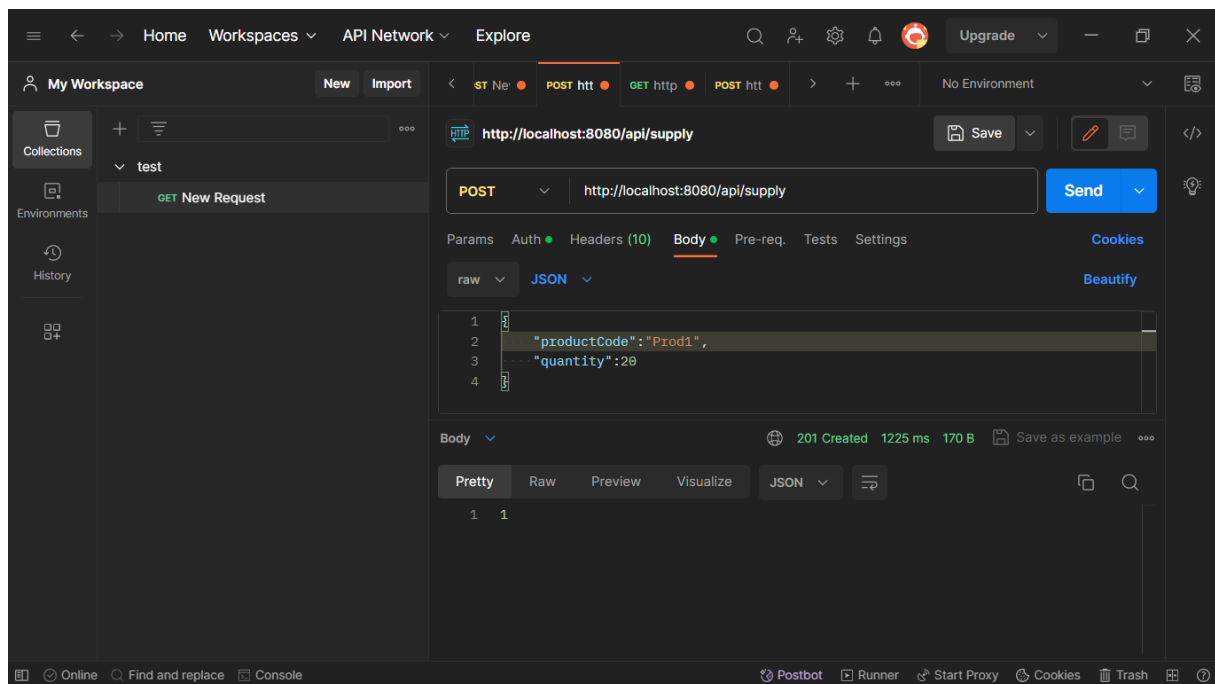


2. Les tests des endPoints sur Postman

a- Création des produit

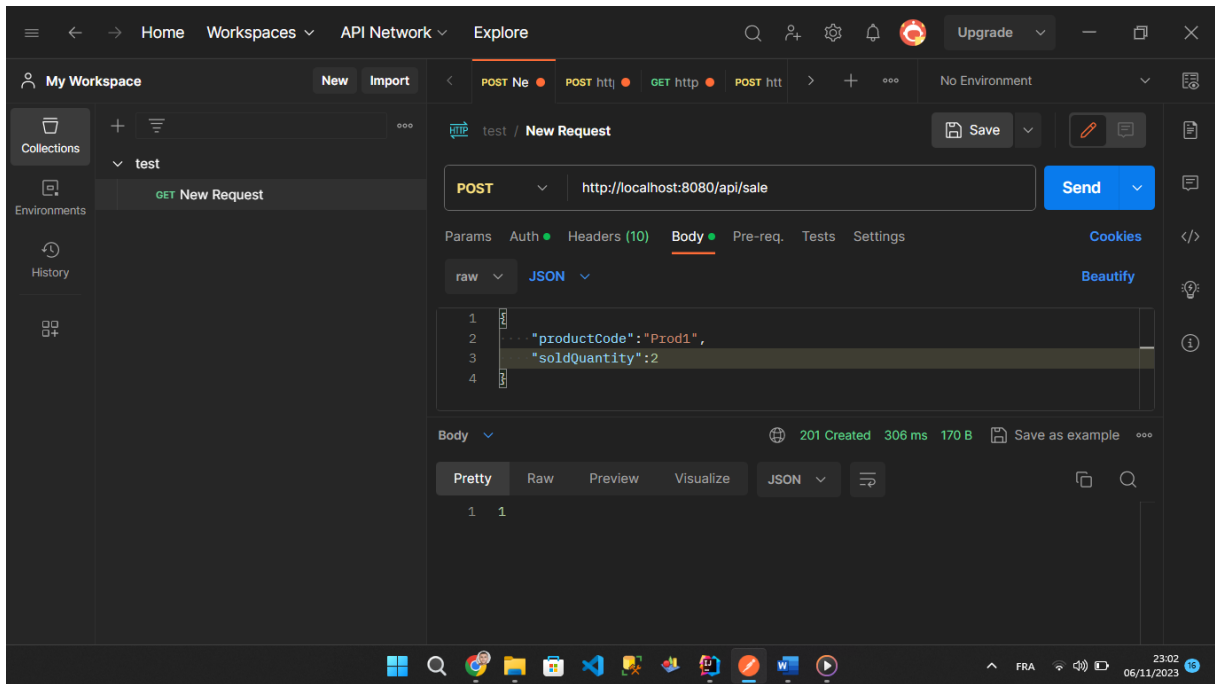


b- Supply

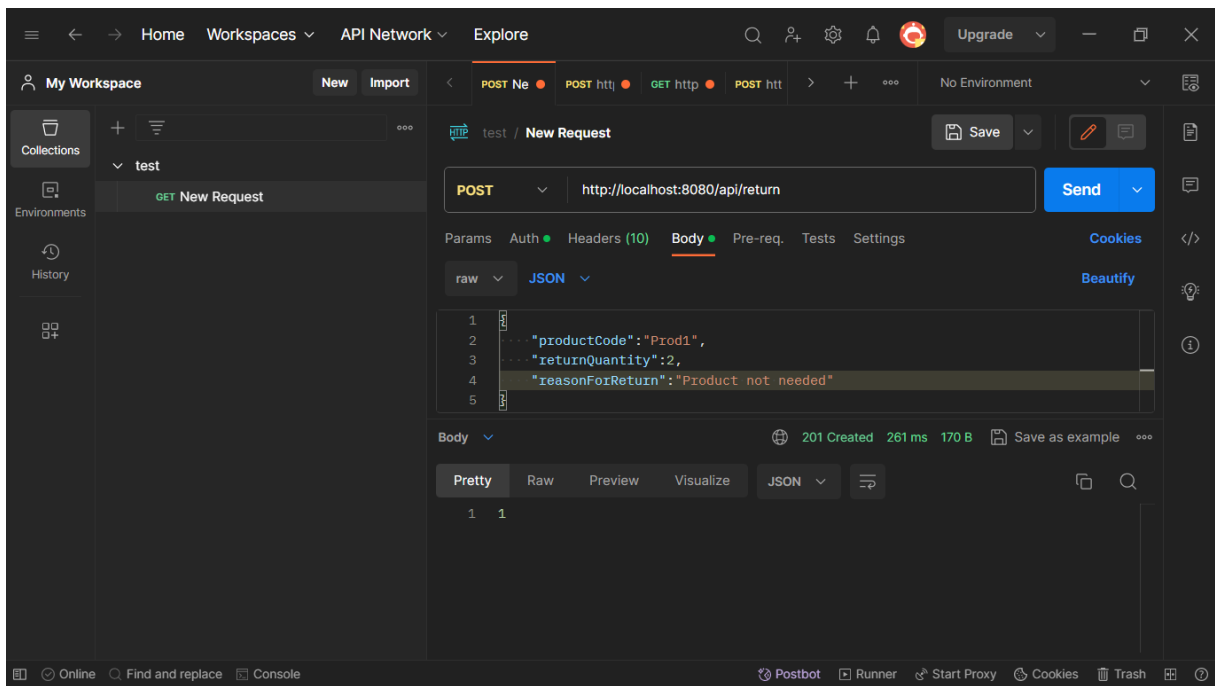


c- Sale

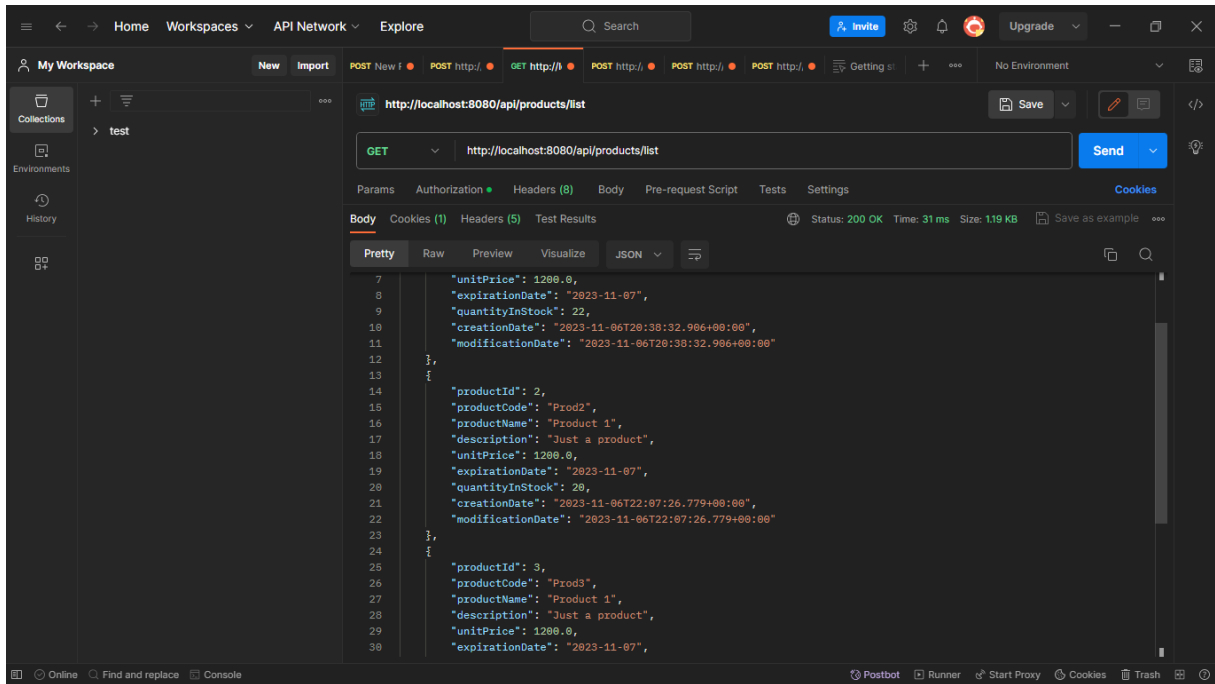
Assigment_2024 Documentation



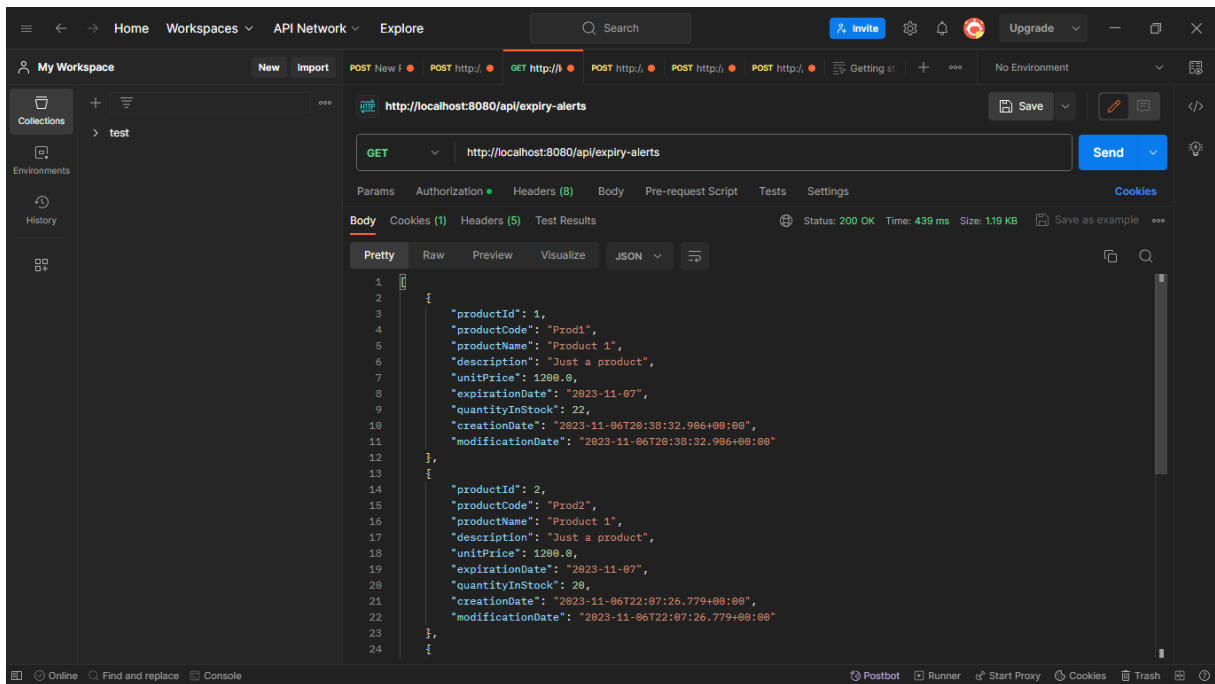
d- Return



e- List products



f- Product nearing expiration date



3. H2-Console

a- Products

The screenshot shows the H2 Console interface. On the left, a sidebar lists database objects: jdbc:h2:mem:dbformation, PRODUCTS, RETURNS, SALES, SUPPLIES, TRANSACTIONHISTORY, USERS, INFORMATION_SCHEMA, and Users. The main area displays the SQL statement `SELECT * FROM PRODUCTS` and its results in a table format. The table has 9 columns: EXPIRATIONDATE, QUANTITYINSTOCK, UNITPRICE, CREATIONDATE, MODIFICATIONDATE, PRODUCTID, DESCRIPTION, PRODUCTCODE, and PRODUCTNAME. There are 4 rows of data, all with a unit price of 1200.0 and a description of 'Just a product'.

| EXPIRATIONDATE | QUANTITYINSTOCK | UNITPRICE | CREATIONDATE | MODIFICATIONDATE | PRODUCTID | DESCRIPTION | PRODUCTCODE | PRODUCTNAME |
|----------------|-----------------|-----------|-------------------------|-------------------------|-----------|----------------|-------------|-------------|
| 2023-11-07 | 22 | 1200.0 | 2023-11-06 21:38:32.906 | 2023-11-06 21:38:32.906 | 1 | Just a product | Prod1 | Product 1 |
| 2023-11-07 | 20 | 1200.0 | 2023-11-06 23:07:26.779 | 2023-11-06 23:07:26.779 | 2 | Just a product | Prod2 | Product 1 |
| 2023-11-07 | 20 | 1200.0 | 2023-11-06 23:07:32.069 | 2023-11-06 23:07:32.069 | 3 | Just a product | Prod3 | Product 1 |
| 2023-11-07 | 20 | 1200.0 | 2023-11-06 23:07:37.505 | 2023-11-06 23:07:37.505 | 4 | Just a product | Prod4 | Product 1 |

(4 rows, 6 ms)

b- Transaction History

The screenshot shows the H2 Console interface. On the left, a sidebar lists database objects: jdbc:h2:mem:dbformation, PRODUCTS, RETURNS, SALES, SUPPLIES, TRANSACTIONHISTORY, USERS, INFORMATION_SCHEMA, and Users. The main area displays the SQL statement `SELECT * FROM TRANSACTIONHISTORY` and its results in a table format. The table has 6 columns: QUANTITY, PRODUCTID, TRANSACTIONDATE, TRANSACTIONID, USERID, and TRANSACTIONTYPE. There are 7 rows of data, showing various transactions like 'Adding product', 'Supply', and 'Add sale'.

| QUANTITY | PRODUCTID | TRANSACTIONDATE | TRANSACTIONID | USERID | TRANSACTIONTYPE |
|----------|-----------|-------------------------|---------------|--------|-----------------|
| 1 | 1 | 2023-11-06 21:38:33.106 | 1 | null | Adding product |
| 20 | 1 | 2023-11-06 23:00:55.838 | 2 | null | Supply |
| 2 | 1 | 2023-11-06 23:02:29.823 | 3 | null | Add sale |
| 2 | 1 | 2023-11-06 23:05:27.221 | 4 | null | Product Return |
| 1 | 2 | 2023-11-06 23:07:26.785 | 5 | null | Adding product |
| 1 | 3 | 2023-11-06 23:07:32.07 | 6 | null | Adding product |
| 1 | 4 | 2023-11-06 23:07:37.513 | 7 | null | Adding product |

(7 rows, 11 ms)