

Code Challenge Documentation

Setup

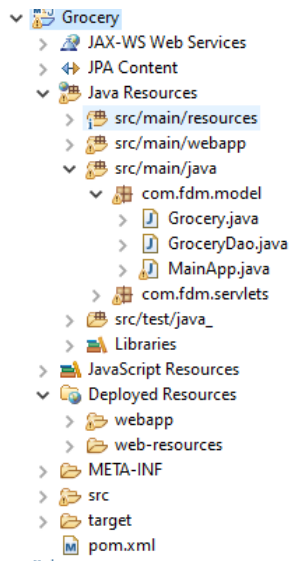


Figure 1. Backend set up

For my implementation, I used JPA to connect Java and Oracle database. The database is used to store the groceries information. I created Grocery entity which used ID as its primary key and it is auto-generated. It has other attributes which are name and category. I also did a DAO to do the SQL query for Grocery including persist, getAll, delete, update and search.

For the backend testing, I tested all the queries in the DAO using Mockito and achieved 100% of the testing coverage.

| | | | | |
|-----------------------------|---------|-----|----|-----|
| NabProject | 95.8 % | 554 | 24 | 578 |
| src/main/java | 87.4 % | 167 | 24 | 191 |
| com.fdm.model | 87.4 % | 167 | 24 | 191 |
| Grocery.java | 46.7 % | 21 | 24 | 45 |
| GroceryDao.java | 100.0 % | 146 | 0 | 146 |
| GroceryDao | 100.0 % | 146 | 0 | 146 |
| GroceryDao(EntityManagerFac | 100.0 % | 6 | 0 | 6 |
| findByCategory(String) | 100.0 % | 25 | 0 | 25 |
| getAll() | 100.0 % | 16 | 0 | 16 |
| persist(Grocery) | 100.0 % | 28 | 0 | 28 |
| remove(int) | 100.0 % | 35 | 0 | 35 |
| update(Grocery) | 100.0 % | 36 | 0 | 36 |
| src/test/java | 100.0 % | 387 | 0 | 387 |

Figure 2. Test coverage

For the front end, I used Java servlets to construct the UI. I used Tomcat as web containers to test the front end.

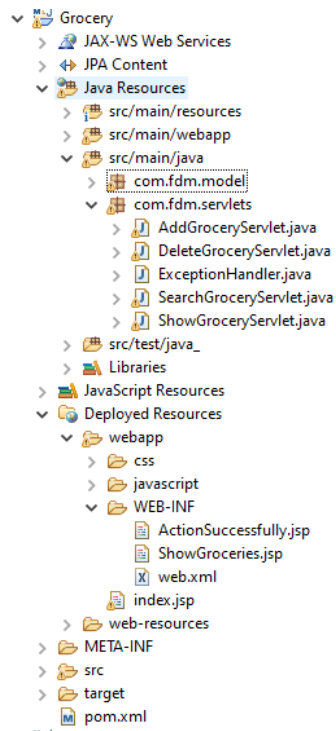


Figure 3. Frontend setup

I used servlets to do some backend processing to the requests inputted from the user, and JSP files to display. The index.jsp is the welcome page, which is shown in the Figure 4.

View existing grocery
[View existing grocery](#)

Add new grocery
 Please enter the grocery detail

Name of the grocery

Category

[Submit](#)

Delete grocery
 Please enter the id of the grocery you want to delete

The id of the grocery you want to delete:

[DeleteGrocery](#)

Search
 The category you want to search:

[SearchCategory](#)

Figure 4. Main UI

The UI has four basic functions which are showing the existing groceries, adding new grocery, deleting grocery and searching.

When clicking the top button which views the existing grocery, it will show all the groceries in the database. The entities shown are all for testing. (Miss Id 1 as I tested delete 1 before)

Show Groceries JSP

Showing all groceries

| Id | Name | Category |
|----|-------------|-------------|
| 2 | sada1 | sadsad |
| 3 | sada1 | sadsad22 |
| 6 | sad | retpgk |
| 4 | sada1 | sadsad22213 |
| 5 | sada2131231 | sadsad22213 |

Click here to return [home](#)

Figure 5.

The next sets of screenshots show the adding query, which new grocery named apple with category fruit is added to the database.

Add new grocery

Please enter the grocery detail

Name of the grocery
Category

Add successfully!

Click here to return [home](#)

| | GROCERYID | CATEGORY | GROCERYNAME |
|---|-----------|-------------|-------------|
| 1 | 1 | fruit | apple |
| 2 | 2 | sadsad | sada1 |
| 3 | 3 | sadsad22 | sada1 |
| 4 | 6 | retpgk | sad |
| 5 | 4 | sadsad22213 | sada1 |
| 6 | 5 | sadsad22213 | sada2131231 |

The next set of screenshots show the delete query, which the grocery with id 4 has been successfully removed from the database.

Delete grocery

Please enter the id of the grocery you want to delete

The Id of the grocery you want to delete:

| | GROCERYID | CATEGORY | GROCERYNAME |
|---|-----------|----------|-------------|
| 1 | 1 | fruit | apple |
| 2 | 2 | sadsad | sadal |
| 3 | 3 | sadsad22 | sadal |
| 4 | 6 | retpgk | sad |

The next set of screenshots show the search function.

Searching Result

Search

The category you want to search:

| Id | Name | Category |
|----|-------|----------|
| 1 | apple | fruit |

Click here to return [home](#)