Remote Coding Test: Sales Orders System

1. Introduction

This document describes the system design of the ***Sales Orders System*** as of the requirement from ***Vue.js + Node.js Remote Coding Test*** document.

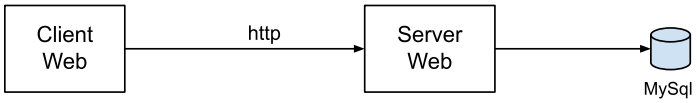
2. Overview

This is a simple system where it provides a single page: ***Sales Orders Listing*** for the ***Sales Administrator*** to filter the sales records.

Assumption made for this system:

- No login page is required to access this page.
- The Category filters: Electronics and Furniture Id is fixed as 1 and 2.

3. System Architecture



3.1. Client Web

The client Web provides the UI for users to view the sales records.

3.2. Server Web

The Server Web provides api for Client Web to retrieve the data from the database.

3.3. MySql

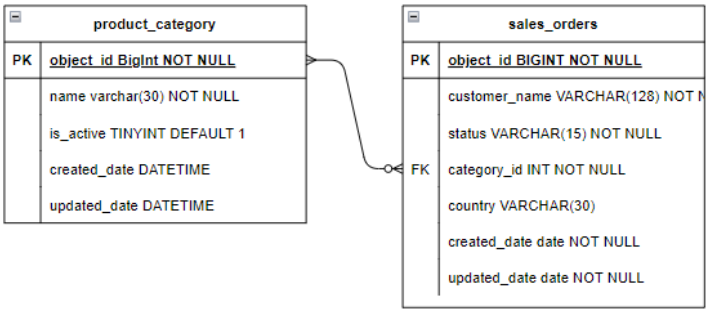
The MySql database is used to store the sales orders data.

4. Data Model

The database design is as provided. Assuming there are only 2 tables:

4.1. Product Category table

4.2. Sales Order table



5. Component Design

5.1. Server Web - The Backend Web Application

This Web application mainly is used to provide api services for the Client Web.

5.1.1. API

The provided API are as following:

5.1.1.1. Sales Orders List API

Method	POST
Path	api/v1/orders/filter
Header	Provide the Authorization in the header
Content-Type	application/json

Sample Request	<pre>{"startDate": "2018-01-01",
	"endDate": "2022-01-01",
	"customerFilter": {"list": ["Jones"],"op": "IN"},
	"statusFilter": {"list": ["Open","Rejected"],"op": "IN"},
	"categoryFilter": {"list": ["Open"],"op": "NOT_IN"},
	"countryFilter": {"list": ["German"],"op": "IN"}
	}</pre>

Sample Response	<pre>{"totalItems": 1,
	"data": [
	{"orderId": "Kivell",
	{"customerName": "Kivell",
	{"status": "Open"},
	{"category": "Furniture"},
	{"country": "Taiwan"},
	{"createdDate": "2019-06-07T16:00:00.000Z"}
	]}</pre>

5.1.1.2. Customer List API

Method	GET
Path	api/v1/orders/customers
Content-Type	application/json

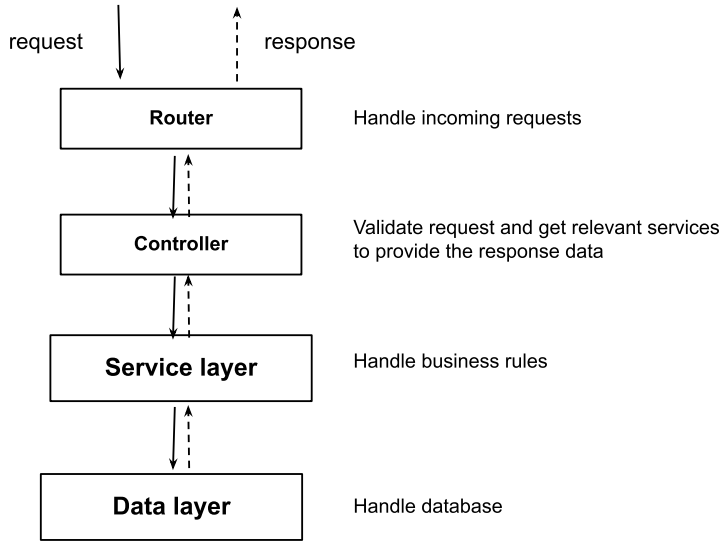
Sample Response	<pre>{"totalItems": 9,
	"data": [
	{"customerName": "Kivell"},
	{"customerName": "Jardine"},
	{"customerName": "Gill"}
	]}</pre>

5.1.1.3. Country List API

Method	GET
Path	api/v1/orders/countries
Content-Type	application/json

Sample Response	<pre>{"totalItems": 9,
	"data": [
	{"country": "United Kingdom"},
	{"country": "Russia"},
	{"country": "German"}
	]}</pre>

5.1.2. Architecture



5.2. Client Web - The Frontend Web Application

This Web application mainly is used to provide web pages for the end user.

The Url : ***[domain-name]/sales-orders/***

5.2.1. The GUI - Sales Order Listing

This page is shown as a tabular format as shown in *Figure 5.1 Sales Order Listing Page.*

This page provides a ***Filter*** button for users to filter the display data. Clicks on the ***Filter*** will prompt a ***Filter Form*** as shown in ***Figure 5.2 Sales Order Listing Filters Form***.

Figure 5.1 Sales Order Listing Page

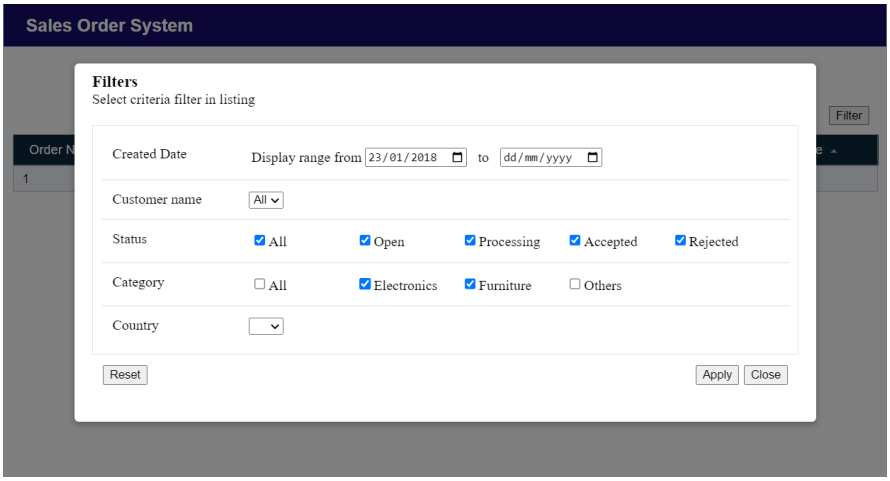


Figure 5.2 Sales Order Listing Filters Form

5.2.2. View Sales Orders Use Case

This use case allows users to view the sales orders. The system applies the previous filters if it is still in the browser storage.

5.2.2.1. User accesses the domain url to view the ***Sales Orders Listing Page***.

5.2.2.2. The system gets the last applied filters stored in the browser session.

5.2.2.3. The system requests the sales orders data with the previous filters from the ***Server Web*** and displays the data on the ***Sales Order Listing Page.***

5.2.3. Sorting Orders Use Case

This use case allows users to sort the sales orders by different fields.

5.2.3.1. User clicks on a target sorting column header.

5.2.3.2. The system sorts the data according to the clicked field.

5.2.4. Filter Sales Orders Use Case

This use case allows users to specify the filters criterias. Users can have the options to reset the filters, apply the filters or discard the filters.

5.2.4.1. User clicks on the ***Filter*** button at the top right of the table.

5.2.4.2. The system prompts the filter form as shown in ***Filter Form***(*Figure 5.2 Sales Order Listing Filters Form).*

5.2.4.3. User selects the criteria and clicks the ***Apply*** button on this form.

5.2.4.4. The system closes the ***Filter Form*** and submits the filter criteria to the ***Server Web***. The ***Server Web*** returns the filtered data to the system. The system refreshes the table with the filtered data.

5.2.4.5. The system saves the filtered criteria to the browser session storage.

Alternative Case: User sets the filters criterias and clicks ***Reset***

5.2.4.6. The system reset all the filter criterias to select “All”.

5.2.4.7. User clicks on the ***Apply*** button on the ***Filter Page***.

5.2.4.8. The system continues to 5.2.4.4 above.

Alternative Case: User sets the filters criterias and clicks ***Close***.

5.2.4.9. The system checks if the new filters are different from the previous filters. If there are no changes, the system closes the form.

5.2.4.10. If there are changes, the system prompts the ***Confirmation Page***(*Figure 5.3 Confirmation Page*) to confirm with the user whether to apply the changes.

5.2.4.11. User clicks on ***Yes*** on the ***Confirmation Page***.

5.2.4.12. The system continues to 5.2.4.4 above.

Alternative Case: User clicks on ***No*** on the ***Confirmation Page***.

5.2.4.13. The system closes the ***Filter Form***.

5.2.4.14. The system returns to the ***Sales Orders Listing Page.***

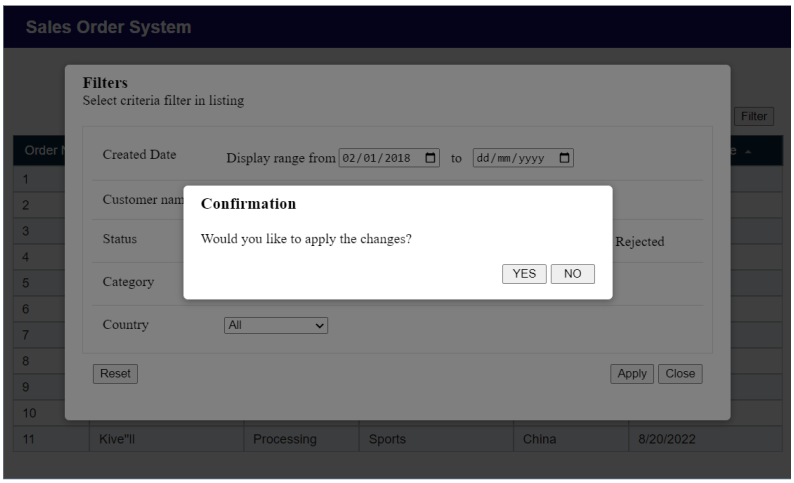


Figure 5.3 Confirmation Page

5.3. Environment Handling

Both ***Client Web*** and ***Server Web*** use the ***dotenv package*** to manage builds for different environments.
2 sample environment files are create:***.env.production*** and ***.env.development***.
Each for production build and development build respectively. The database and Url is configured in these environment files.

For Server Web environment files uses to configure the following:

API_KEY ~ the api Authorization key
PORT ~ the http port to use for this Web application
DB_USER ~ database user
DB_PASSWORD ~ database password
DB_NAME ~ database name

For Client Web environment files uses to configure the following:

PORT ~ the client web application port
VUE_APP_SERVER_URL~ The Server Web url
VUE_APP_API_KEY ~ the api Authorization key

NOTE: In Practical, the .env.production and .env.development should not be placed into the git repository.

6. Suggestion

6.1. *Sales Order Listing Filters*

In my opinion, I would prefer to place the filters on top of the Table Listing. This is because:

- It provides a clearer view of the filter options.
- It is more mobile friendly. As popups modal does not look good in mobile view.

7. Conclusion

The records of the time taken for this project:

Task	Duration
Analysis & Design	0.5 days
Development & Testing	2 days
Documentation	0.5 days

The source code is available in **Github**:
https://github.com/ghwoan/assessment.git.