

# DAY 3 - API INTEGRATION AND DATA MIGRATION - [General E-commerce]

Date: 17/01/2025

## Report Documentation:

### API Integration Process:

- API call was made to fetch product details dynamically.
- API token was included in request headers for security.
- Data was displayed on the frontend using React hooks (useState & useEffect).

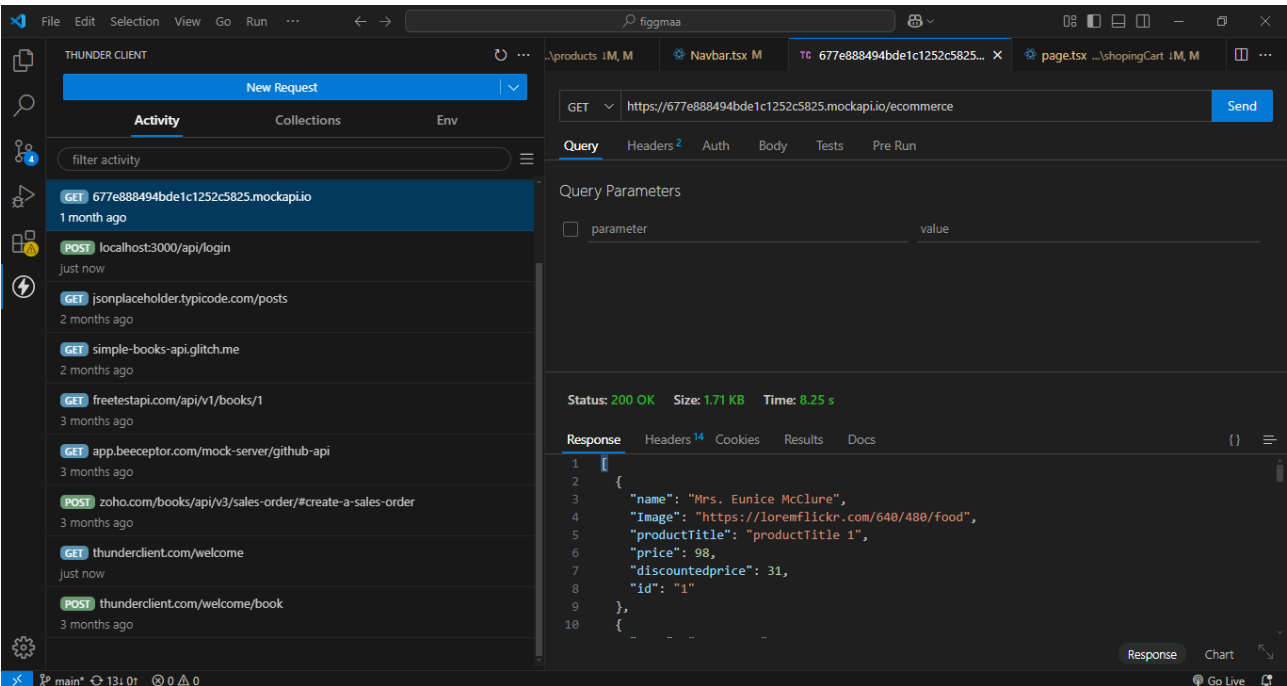
### Adjustments Made to Schemas:

- Added static ID field for easy data retrieval.
- Updated schema to align with the API structure.


### Migration Steps and Tools Used:

- Used CLI tools to migrate product data to new schema.
- Ensured compatibility before migration.


### API GET Request:




### API Response Handling:




**Country Burger**  
Classic country-style burger served with fries.  
Category: Sandwich  
**\$45.00 \$50.00**  
Available  
Recommended



**Pizza**  
Delicious vegetarian pizza topped with fresh vegetables and cheese.  
Category: Main Course  
**\$43.00 \$50.00**  
Available  
Cheesy Vegetarian



**Chicken Chup**  
Crispy fried chicken bites served with dipping sauce.  
Category: Appetizer  
**\$12.00 \$15.00**  
Available  
Sell Crispy




**Fresh Lime**  
Refreshing fresh lime drink made with natural ingredients.  
Category: Drink  
**\$38.00 \$45.00**  
Available  
Healthy Popular

Database Schema Update:

Foodtuck

[Home](#)[Menu](#)[Blog](#)[Pages](#)[About](#)[Shop](#)[Chef](#)[Foods](#)[Contact](#)

Product Detail



### Chicken Chup

Crispy fried chicken bites served with dipping sauce.

**\$12.00 \$15.00**

Add to Cart

Data Migration Script:

File Edit Selection View Go Run ...

figgmaa

EXPLORER

FIGGMAA

.next

vercel

project.json

README.txt

node\_modules

public

scripts

**JS import-data.mjs**

src

app

404Error

about

blog

Cart

page.tsx

checkout

checkoutt

Coffee

components

constant

contact

context

FAQ

foodcatagery

HomePage

ffcode-workspace

OUTLINE

TIMELINE

scripts > JS import-data.mjs > ...

```
1 import { createClient } from '@sanity/client';
2 import axios from 'axios';
3 import dotenv from 'dotenv';
4 import { fileURLToPath } from 'url';
5 import path from 'path';
6
7 // Load environment variables from .env.local
8 const __filename = fileURLToPath(import.meta.url);
9 const __dirname = path.dirname(__filename);
10 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12 // Create Sanity client
13 const client = createClient({
14   projectId: "ettb5rql",
15   dataset: "production",
16   useCdn: false,
17   token: "sk09UqL0nvKTNp6LvaQp6cvcroiYE3VTuxbo6CPnHafEFUSjn31dxemhxQFmvBVxnHe9tP3Dq5j7PYcFNcpsPJVorux1p8jYoMXyOj9vkaEpmh7Ljp",
18   ,
19   apiVersion: '2021-08-31',
20 });
21
22 async function uploadImageToSanity(imageUrl) {
23   try {
24     console.log(`Uploading image: ${imageUrl}`);
25     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
26     const buffer = Buffer.from(response.data);
27     const asset = await client.assets.upload('image', buffer, {
28       filename: imageUrl.split('/').pop(),
29     });
30     console.log(`Image uploaded successfully: ${asset._id}`);
31     return asset._id;
32   } catch (error) {
33   }
```

Bushratürk (1 week ago)

Ln 6, Col 1

Spaces: 2

UTF-8

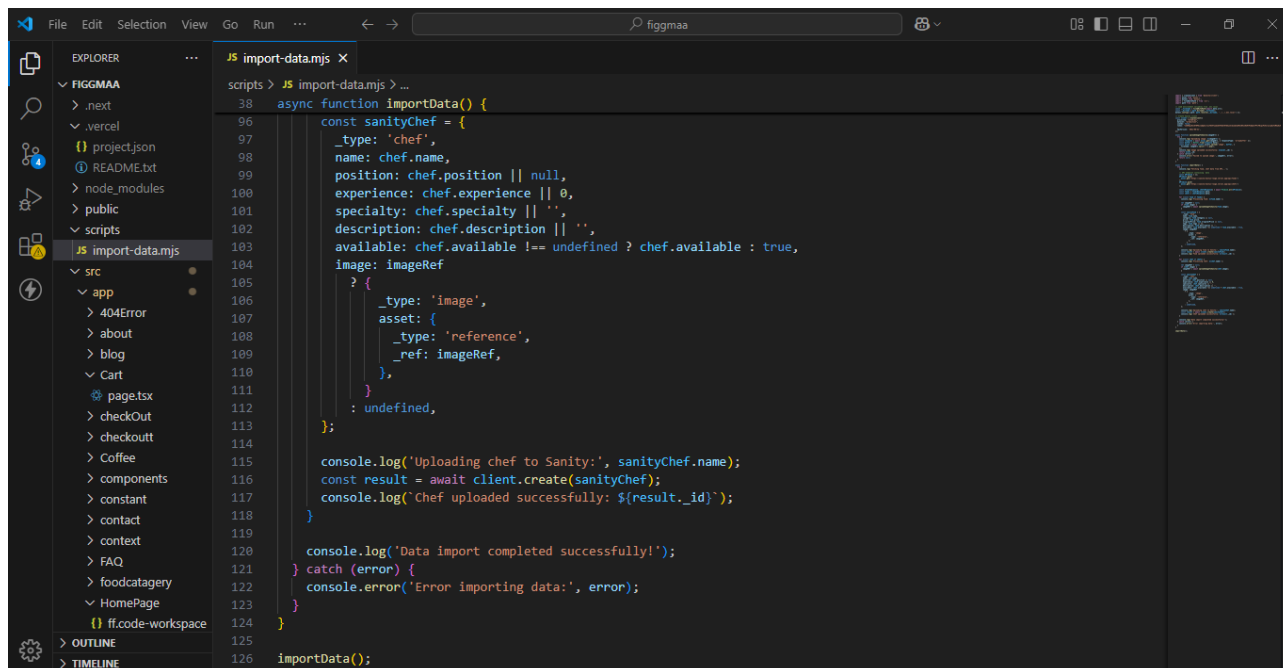
CRLF

JavaScript

Go Live

Prettier

## Final Output Display:



The screenshot shows a VS Code editor window with the file explorer on the left and the code editor in the center. The file explorer shows a project structure for 'FIGGMAA' with a 'scripts' folder containing 'import-data.mjs'. The code editor displays the content of 'import-data.mjs', which is a JavaScript file. The code defines an async function 'importData()' that creates a 'sanityChef' object and uploads it to a Sanity API. The code includes console logs for successful uploads and error handling for failed imports. The code is as follows:

```
38 async function importData() {
96   const sanityChef = {
97     _type: 'chef',
98     name: chef.name,
99     position: chef.position || null,
100    experience: chef.experience || 0,
101    specialty: chef.specialty || '',
102    description: chef.description || '',
103    available: chef.available !== undefined ? chef.available : true,
104    image: imageRef
105    ? {
106      _type: 'image',
107      asset: {
108        _type: 'reference',
109        _ref: imageRef,
110      },
111    }
112    : undefined,
113  };
114
115  console.log('Uploading chef to Sanity:', sanityChef.name);
116  const result = await client.create(sanityChef);
117  console.log('Chef uploaded successfully: ${result._id}');
118  }
119
120  console.log('Data import completed successfully!');
121  } catch (error) {
122    console.error('Error importing data:', error);
123  }
124  }
125
126  importData();
```

## Day 3 Checklist:

- API Understanding (Completed)
- Schema Validation (Completed)
- Data Migration (Completed)
- API Integration in Next.js (Completed)
- Submission Preparation (Completed)