

# Day 3 - API Integration and Data Migration Assignment

## Title:

Day 3 - API Integration Report – Furniture MarketPlace  
“Hekto”

## Introduction

The objective of this assignment is to integrate APIs and migrate data into Sanity CMS, ensuring compatibility with the chosen template. This report will document the process of schema validation, data migration, and API integration using Next.js, as well as the challenges faced and solutions implemented.

## 1. API Documentation

### API Overview

- **Selected Template:** Template 4 (Sir Anas and Sir Ubaid Ur Rehman)

- **API Endpoint:** <https://next-ecommerce-template-4.vercel.app/api/product>

## API Response

```
"products": [  
  {  
    "id": "1",  
    "name": "Tribù Elio Chair",  
    "imagePath": "https://next-ecommerce-template-4.vercel.app/product/Chair (5).png",  
    "price": "1200",  
    "description": "A sleek outdoor chair with natural wooden elements and a modern look.",  
    "discountPercentage": 10,  
    "isFeaturedProduct": true,  
    "stockLevel": 25,  
    "category": "Chair"  
  },  
  {  
    "id": "2",  
    "name": "Armchair Tortuga",  
    "imagePath": "https://next-ecommerce-template-4.vercel.app/product/Chair (35).png",  
    "price": "850",  
    "description": "An elegant armchair with plush cushions and a curved design for comfort.",  
    "discountPercentage": 0,  
    "isFeaturedProduct": false,  
    "stockLevel": 10,  
    "category": "Chair"  
  },  
  {  
    "id": "3",  
    "name": "Uchiwa Quilted Lounge Chair",  
    "imagePath": "https://next-ecommerce-template-4.vercel.app/product/Chair (6).png",  
    "price": "1600",  
    "description": "A spacious lounge chair with a quilted back and soft cushioning.",  
    "discountPercentage": 20,  
    "isFeaturedProduct": true,  
    "stockLevel": 5,  
    "category": "Chair"  
  },  
  {  
    "id": "4",  
    "name": "Stylish Golden Metal Legs Mint Blue Fabric Velvet Sofa Leisure Armchair",  
    "imagePath": "https://next-ecommerce-template-4.vercel.app/product/Chair (3).png",  
    "price": "780",  
  }  
]
```

## 2. API Integration Process

### Steps I Followed:

#### 1. Running the Command to Import Data:

a. I opened my terminal and ran the command:

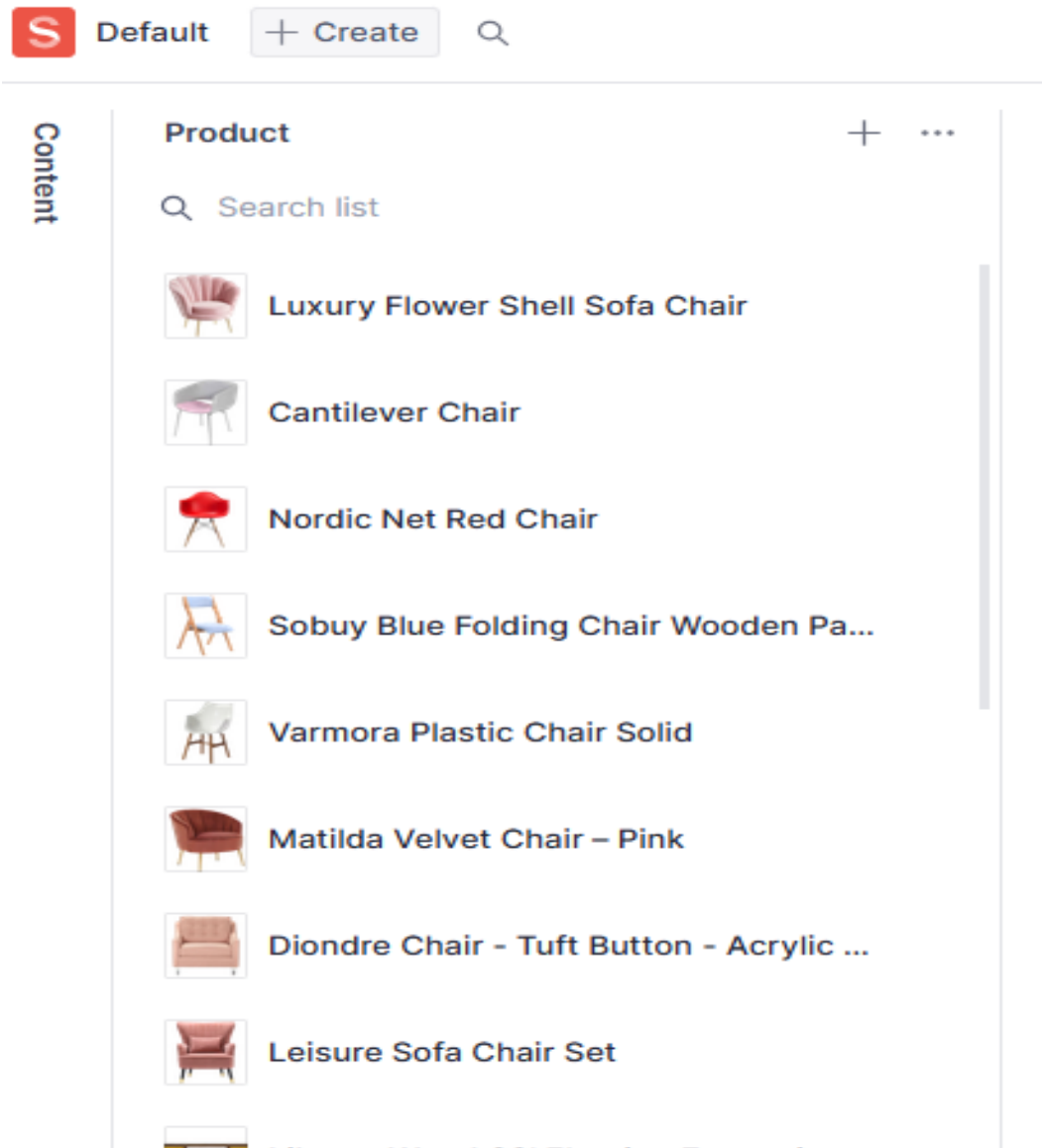
```
npm run import-data
```

b. This command fetched data from the provided API and transferred it to Sanity CMS.

#### 2. Verifying Data in Sanity CMS:

a. After running the command, I navigated to my Sanity Studio.

b. I confirmed that the data was successfully imported by checking the relevant document types (e.g., product schema).



### 3. Adding the Token from Sanity:

- a. To enable secure API operations, I generated and added a token from Sanity:
  - i. I accessed my Sanity project dashboard and navigated to **Settings > API**.

- ii. I clicked **Add Token**, assigned it the necessary permissions ( "Read" or "Write"), and generated the token.
- iii. I copied the token and added it to my project's `.env` file.
- iv. Finally, I restarted my development server to apply the new environment variables.

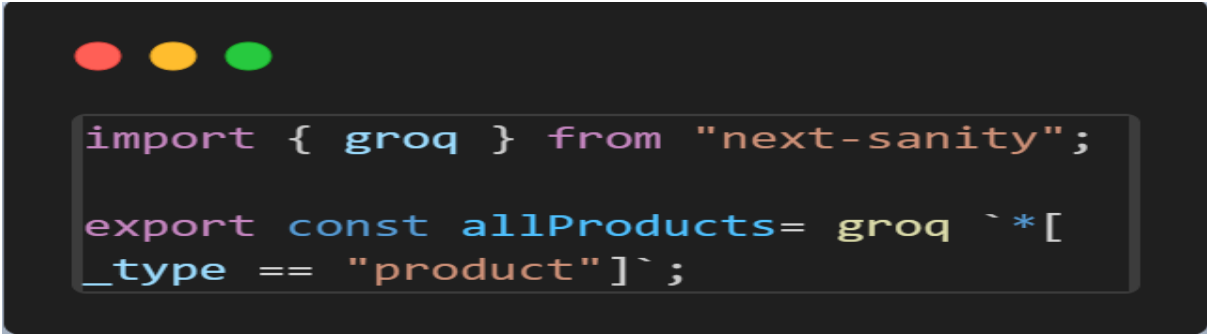
### 3. Fetching Data from Sanity

#### Steps I Followed:

##### 1. Created a `queries.ts` File:

- a. I created a `queries.ts` file in the `sanity/lib` folder to define the GROQ query for fetching products from Sanity CMS.

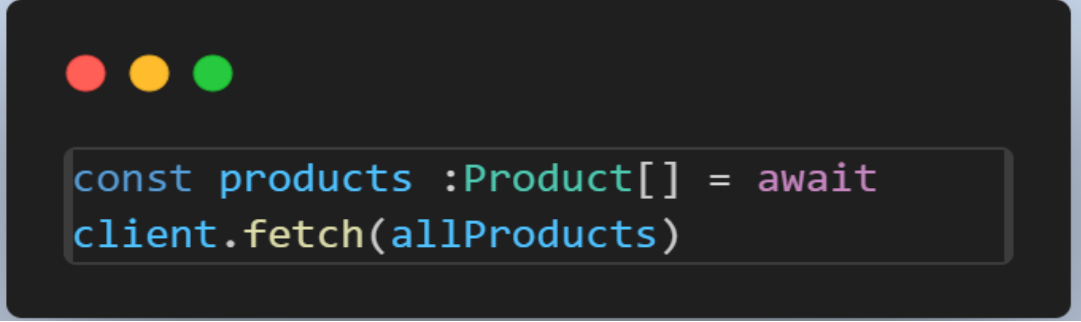
- b. GROQ Query:



```
import { groq } from "next-sanity";  
  
export const allProducts = groq `*[_type == "product"]`;
```


##### 4. Fetching Data in the `/products` API Route:

- I fetched the data using the GROQ query in the /products API route.
- Code:



```
const products :Product[] = await  
client.fetch(allProducts)
```

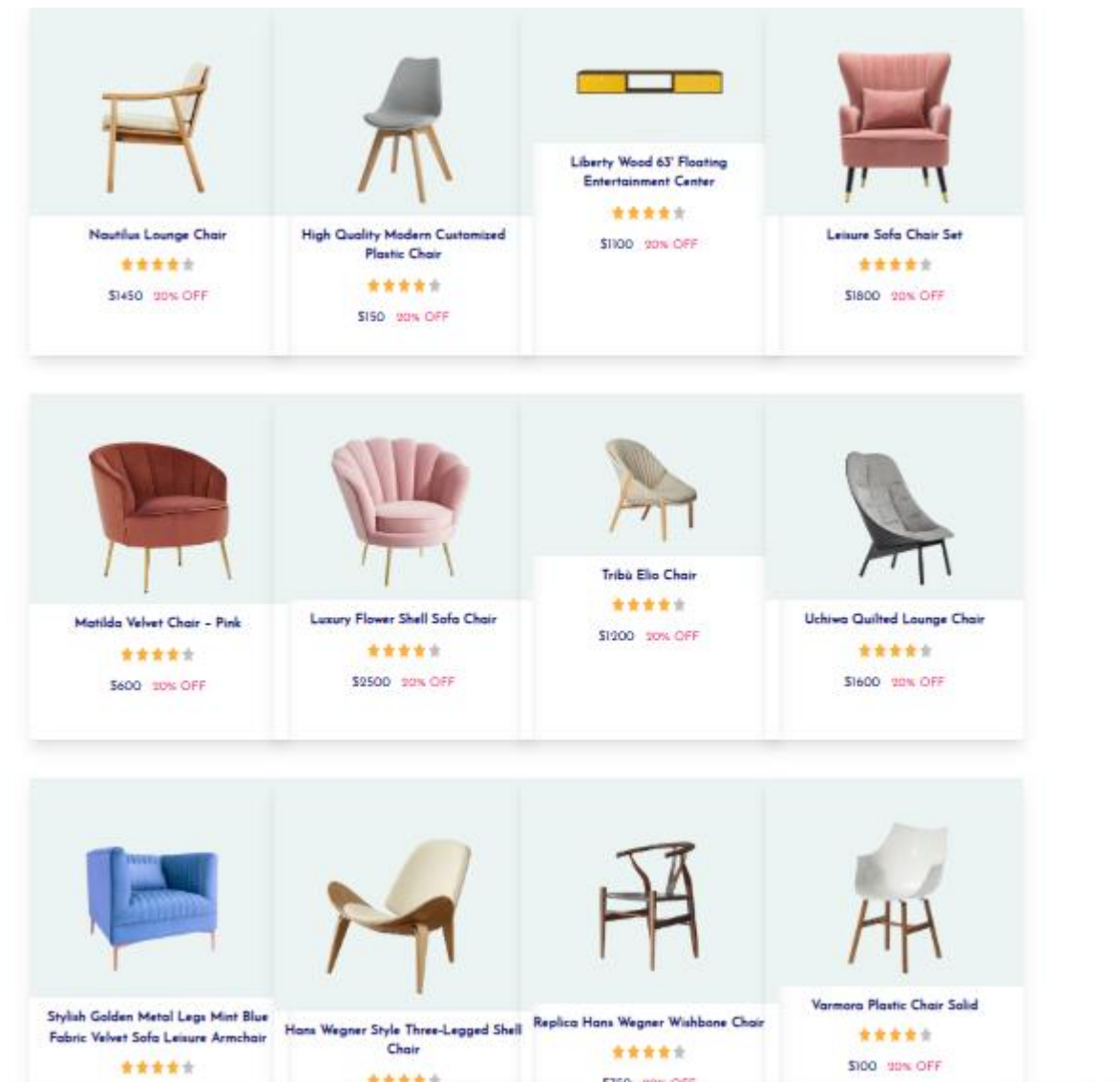
- **Created a Product Interface:**
- I defined the structure of a Product in the types folder to ensure type safety.
- Interface Code:



```
1 export interface Product{
2     _id:string;
3     name:string;
4     _type:"product";
5     image?:{asset:{
6         _ref:string;
7         _type:"image";
8     }}
9     price:number;
10    description?:string;
11    category:string;
12    isFeaturedProduct?:boolean
13    createdAt:string
14    stock?:number
15    review?: Array<Review>;
16 }
```

## Verified the Functionality:

- I tested the /products route after starting the server to ensure it fetched all product data correctly.





## Checklist

- API Understanding: ✓
- Schema Validation: ✓
- Data Migration: ✓
- API Integration in Next.js: ✓

## Conclusion

This assignment enhanced my understanding of API integration, data migration, and schema validation. By documenting each step and overcoming challenges, I successfully populated Sanity CMS with API data and displayed it using Next.js.