

Furniture Bazaar: Day 3 - API Integration and Data Migration

Project Overview

Project Name: Furniture Bazaar

Goal: Create a marketplace for furniture with dynamic data display by integrating APIs and managing data migration.

API Integration

API Details

- **API URL:** <https://template-0-beta.vercel.app/api/product>
- **Endpoint Used:** /products

Fetches Data Fields

1. name
2. description
3. price
4. discountPercentage
5. isFeaturedProduct
6. stockLevel
7. category
8. imageURL

Security Measures

- **Sensitive Information:** API keys are securely stored using .env files to avoid exposure.
-

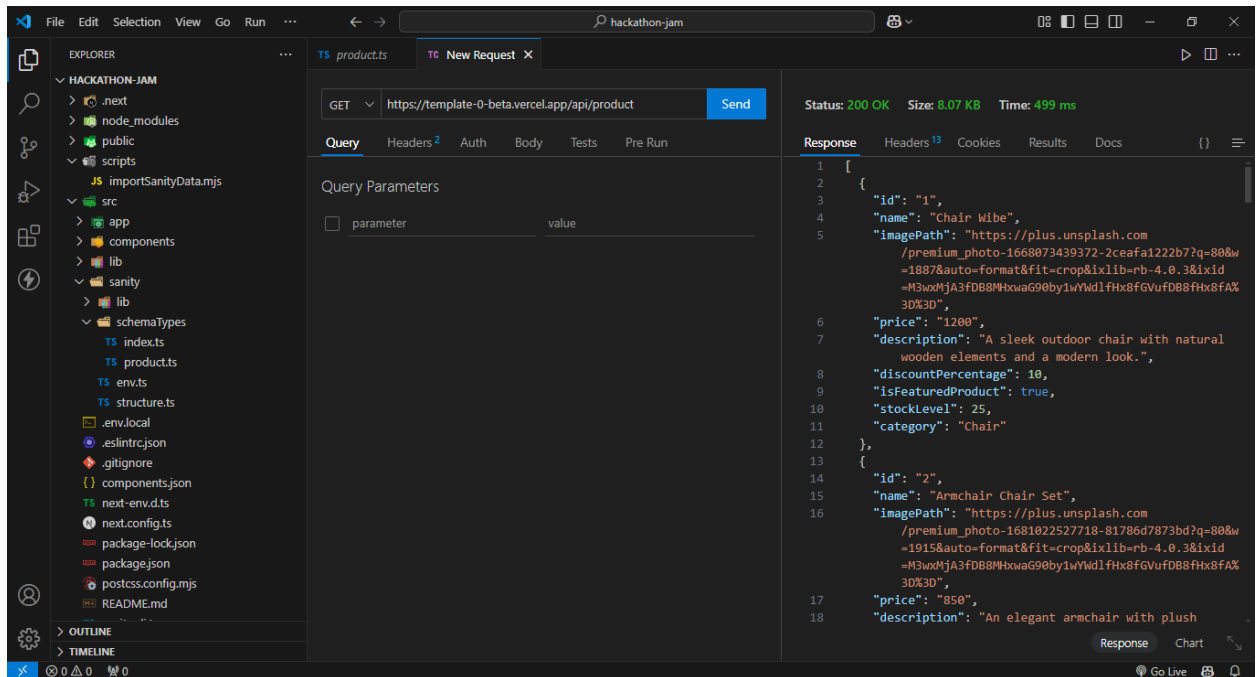
Data Migration

Existing Dataset

- **Source:** .JSON files.

Validation

- **Method:** Schema validation ensures the integrity and correctness of data.



Migration Method

- **Approach:** Data will be migrated using API fetching and scripting to automate the process and avoid manual errors while making importSanityData.mjs file in scripts folder and implementing code to fetch data to sanity.
- Install axios to fetch data **npm install @sanity/client axios**

```

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 // Create Sanity client
12 const client = createClient({
13   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
14   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
15   useCdn: false,
16   token: process.env.SANITY_API_TOKEN,
17   apiVersion: '2021-08-31'
18 })
19 async function uploadImageToSanity(imageUrl) {
20   try {
21     console.log(`Uploading image: ${imageUrl}`)
22     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' })
23     const buffer = Buffer.from(response.data)
24     const asset = await client.assets.upload('image', buffer, {
25       filename: imageUrl.split('/').pop()
26     })
27     console.log(`Image uploaded successfully: ${asset._id}`)
28     return asset._id
29   } catch (error) {
30     console.error('Failed to upload image:', imageUrl, error)
31     return null
32   }
33 }
34 async function importData() {
35   try {
36     console.log('Fetching products from API...')
37     const response = await axios.get('https://template-0-beta.vercel.app/api/product')
38     const products = response.data
39     console.log(`Fetched ${products.length} products`)
40     for (const product of products) {
41       console.log(`Processing product: ${product.title}`)
42       let imageRef = null
43       if (product.imagePath) {
44         imageRef = await uploadImageToSanity(product.imagePath)
45       }
46       const sanityProduct = {
47         _type: 'product',
48         name: product.name,
49         description: product.description,
50         price: parseInt(product.price),
51         discountPercentage: product.discountPercentage,
52         isFeaturedProduct: product.isFeaturedProduct,
53         stockLevel: product.stockLevel,
54         category: product.category,
55         image: imageRef ? {
56           _type: 'image',
57           asset: {
58             _type: 'reference',
59             _ref: imageRef,
60           },
61         } : undefined,
62       }
63     }
64   }
65 }

```

- After writing the code we run the code to transfer data to sanity fields `npm run importSanityData.mjs` .

Sanity Schema Adjustments

- **Adjustments Needed:** The schema will be modified to align with API data fields and maintain compatibility with frontend requirements.
- **Critical Fields and Relationships:**
 - Product fields such as name, price, and category will be mapped for proper display in the frontend.

Frontend Integration

Framework

- **Technology Used:** Next.js

Display Strategy

- **Implementation:** The data will be dynamically displayed on the product listing page, fetched from Sanity CMS.

Testing and Validation

Tools

- **Testing Platform:** Thunder Client in VS Code is used to verify API endpoints and validate data migration scripts.

Challenges and Edge Cases

- Handling cases where data fields like `imageUrl` or `category` are missing.
- Ensuring smooth frontend updates when API data changes.

Expected Deliverables

1. Successfully imported API data into Sanity CMS.
2. Dynamic data rendering in the frontend using Next.js and Sanity.

Day 3 Checklist:

Self Validation Checklist:

Api Understanding:

- ✓

Schema Validation:

- ✓

Data Migration:

- ✓

Api Integration in Next.js:

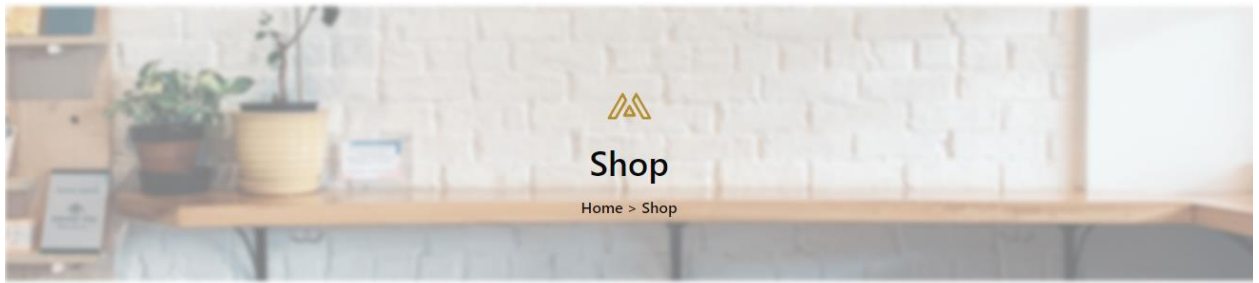
- ✓
-

Attachments

Sanity Schema Screenshot:

```
1 import { defineType,defineField } from "sanity";
2
3 export const product = defineType({
4   name: 'product',
5   type: 'document',
6   title: 'Product',
7   fields: [
8     defineField({
9       name: 'name',
10      type: 'string',
11      title: 'Product Name',
12    }),
13    defineField({
14      title: 'Slug',
15      name: 'slug',
16      type: 'slug',
17      options: {
18        source: 'name',
19        maxLength: 200, // will be ignored if slugify is set
20        slugify: input => input
21                      .toLowerCase()
22                      .replace(/\s+/g, '-')
23                      .slice(0, 200)
24      }
25    }),
26    defineField({
27      name: 'description',
28      type: 'string',
29      title: 'Description'
30    }),
31    defineField({
32      name: 'price',
33      type: 'number',
34      title: 'Product Price',
35    }),
36    defineField({
37      name: 'discountPercentage',
38      type: 'number',
39      title: 'Discount Percentage',
40    }),
41    defineField({
42      name: 'isFeaturedProduct',
43      type: 'boolean',
44      title: 'Is Featured Product',
45      description: 'Indicates if the product is featured',
46      initialValue: false, // Default value
47    }),
48    defineField({
49      name: 'stockLevel',
50      type: 'number'
```

1. **Fronted Output:**



[Filter](#) [Grid](#) [List](#) | Showing 1-21 of 21 results

Show [21](#) Sort by [Default](#)



Hans Wegner Style Three-Seat Sofa

Rs. 990



Leisure Sofa Chair Set

Rs. 1800



White Bed

Rs. 120



Chair Wibe

Rs. 1200



Armchair Chair Set

Rs. 850



Pink Lounge Chair

Rs. 1600



Stylish Armchair

Rs. 780



Rapson Thirty-Nine Sofa

Rs. 1300



High Quality Modern Sofa

Rs. 150



Alpha Table

Rs. 900



Sleek Modern Table

Rs. 2000



Liberty Center

Rs. 1100



Diondre Chair

Rs. 720



Red Bed

Rs. 320



Blue Bed

Rs. 780



Luxury Flower Bed

Rs. 2500



Nautilus Lounge Chair

Rs. 1450



Cozy Sofa

Rs. 520



Matilda Velvet Bed

Rs. 600



Leisure Sofa Chair Set

RS.1800

★★★★★ | 5 Customer Review

A comfortable set of chairs with soft cushions for relaxation.

Size

L XL XS

Color



- 6 +

Add To Cart

SKU : SS01

Category : Sofa

Tags : Sofa, Chair, Home, Shop

Share : [Facebook](#) [LinkedIn](#) [Twitter](#)

Description

Additional Information

Review [5]

Embodying the raw, wayward spirit of rock n roll, the Kilburn portable active stereo speaker takes the unmistakable look and sound of Marshall, unplugs the chords, and takes the show on the road.

Weighing in under 7 pounds, the Kilburn is a lightweight piece of vintage styled engineering. Setting the bar as one of the loudest speakers in its class, the Kilburn is a compact, stout-hearted hero with a well-balanced audio which boasts a clear midrange and extended highs for a sound that is both articulate and pronounced. The analogue knobs allow you to fine tune the controls to your personal preferences while the guitar-influenced leather strap enables easy and stylish travel.



Related Product

