

Day 3 - API Integration Report: Furniro E-Commerce

Overview:

This report documents the API integration and data management setup for the Furniro website using Sanity CMS. The goal is to design a seamless system to retrieve product data from the backend and display it accurately on the frontend. The report includes the integration process, schema customization, migration steps, and screenshots.

API Integration Process:

1: Data Source

Data was fetched from the given API endpoint: <https://template6-six.vercel.app/api/products>

The product data included the following fields: Title, Description, Price, Image URL, Discount Percentage, Tags, isNew

2: Integration Steps

- **Schema Design:** A custom product schema was created in Sanity CMS to align with the structure of the imported data.
- **Import Script:** A migration script was written to fetch data from the API and upload it to Sanity CMS.
- **Image Handling:** The script was enhanced to handle images as arrays and generate unique references.
- **Data Validation:** Validation rules for fields like slug, price, and reviews were implemented.
- **API Endpoints:** Endpoints were created to retrieve data from Sanity CMS for use in the frontend.

3: Schema Adjustments

Product Schema Custom fields were added:

Category

Slug

Summary

Key validation rules and slug uniqueness checks were implemented.

```

1  import { defineType } from "sanity"
2
3  export const product = defineType({
4    name: "product",
5    title: "Product",
6    type: "document",
7    fields: [
8      {
9        name: "title",
10       title: "Title",
11       validation: (rule) => rule.required(),
12       type: "string"
13     },
14     {
15       name: "description",
16       type: "text",
17       validation: (rule) => rule.required(),
18       title: "Description",
19     },
20     {
21       name: "productImage",
22       type: "image",
23       validation: (rule) => rule.required(),
24       title: "Product Image"
25     },
26     {
27       name: "price",
28       type: "number",
29       validation: (rule) => rule.required(),
30       title: "Price",
31     },
32     {
33       name: 'slug',
34       title: 'Slug',
35       type: 'slug',
36       options: {
37         source: 'title',
38         maxLength: 200,
39       },
40     },
41     {
42       name: "category",
43       type: "string",
44       title: "Category",
45       validation: (rule) => rule.required(),
46       options: {
47         list: [
48           { title: "Stylish Cafe Chair", value: "Stylish Cafe Chair" },
49           { title: "Luxury Big Sofa", value: "Luxury Big Sofa" },
50           { title: "Outdoor bar table and stool", value: "Outdoor bar table and stool" },
51           { title: "Night lamp", value: "Night lamp" },
52           { title: "Small mug", value: "Small mug" },
53           { title: "Cute bed set", value: "Cute bed set" },
54           { title: "Minimalist flower pot", value: "Minimalist flower pot" }
55         ]
56       },
57     },
58     {
59       name: 'summary',
60       title: 'Summary',
61       type: 'text',
62       validation: (rule) => rule.required(),
63       description: 'Summary of the product'
64     },
65     {
66       name: "tags",
67       type: "array",
68       title: "Tags",
69       of: [{ type: "string" }]
70     },
71     {
72       name: "dicountPercentage",
73       type: "number",
74       title: "Discount Percentage",
75     },
76     {
77       name: "isNew",
78       type: "boolean",
79       title: "New Badge",
80     }
81   ]
82 })

```

Migration Steps:

1. Tools Used

- Sanity Client: For uploading data.
- Fetch: For API calls.
- .env: For managing environment variables.

2. Migration Script

The script fetches API data and imports it into Sanity CMS while handling images.

```
1 import { createClient } from '@sanity/client';
2
3 const client = createClient({
4   projectId: 'ysdjtcex',
5   dataset: 'production',
6   useCdn: true,
7   apiVersion: '2025-01-18',
8   token: 'skS29k11lpX78h2Qs1FFRn6f4KYBEF9JmbFdQuUxd5rztVW82c4WheuK5S154KTQYbCHTupVrv2b5Maua1KkMk09hF9m0FrsXXCanBFZo9wKU1517sR8MHHDz2FcVQ05Myg1bw21D97x01KD8Y4u1LLZ0yhgWtYOF9sPor5FKXLVrp016',
9 });
10
11 async function uploadImageToSanity(imageUrl) {
12   try {
13     console.log('Uploading image: ${imageUrl}');
14
15     const response = await fetch(imageUrl);
16     if (!response.ok) {
17       throw new Error('Failed to fetch image: ${imageUrl}');
18     }
19
20     const buffer = await response.arrayBuffer();
21     const bufferImage = Buffer.from(buffer);
22
23     const asset = await client.assets.upload('image', bufferImage, {
24       filename: imageUrl.split('/').pop(),
25     });
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
35 async function uploadProduct(product) {
36   try {
37     const imageId = await uploadImageToSanity(product.imageUrl);
38
39     if (imageId) {
40       const document = {
41         _type: 'product',
42         title: product.title,
43         price: product.price,
44         productImage: {
45           _type: 'image',
46           asset: {
47             _ref: imageId,
48           },
49         },
50         tags: product.tags,
51         discountPercentage: product.discountPercentage, // Type in field name: discountPercentage -> discountPercentage
52         description: product.description,
53         isNew: product.isNew,
54       };
55
56       const createdProduct = await client.create(document);
57       console.log('Product ${product.title} uploaded successfully:', createdProduct);
58     } else {
59       console.log('Product ${product.title} skipped due to image upload failure. ');
60     }
61   } catch (error) {
62     console.error('Error uploading product:', error);
63   }
64 }
65
66 async function importProducts() {
67   try {
68     const response = await fetch('https://template6-six.vercel.app/api/products');
69
70     if (!response.ok) {
71       throw new Error('HTTP error! Status: ${response.status}');
72     }
73
74     const products = await response.json();
75
76     for (const product of products) {
77       await uploadProduct(product);
78     }
79   } catch (error) {
80     console.error('Error fetching products:', error);
81   }
82 }
83
84 importProducts();
```

API Calls:

Fetch All Products

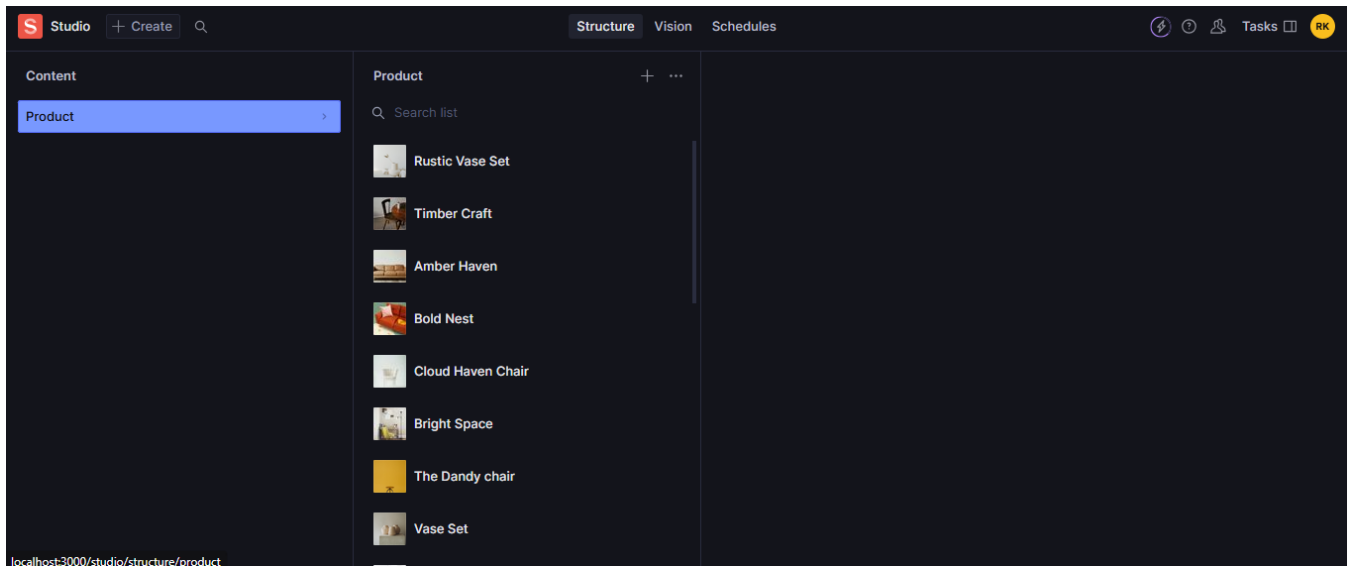
```
1  import { groq } from "next-sanity";
2  import sanityClient from "../sanity.client";
3
4  export async function getProducts() {
5    return sanityClient.fetch(
6      groq `
7        *[_type == "product"]{
8          _id,
9          title,
10         description,
11         "slug": slug.current,
12         "productImage": productImage.asset->url,
13         price,
14         category,
15         summary,
16         tags,
17         isNew
18       }
19     `
20   )
21 }
22
23
24
25 export async function getHomeProducts() {
26   return sanityClient.fetch(
27     groq `
28       *[_type == "product"][0...8]{
29         _id,
30         title,
31         description,
32         "slug": slug.current,
33         "productImage": productImage.asset->url,
34         price,
35         category,
36         summary,
37         tags,
38         isNew
39       }
40     `
41   )
42 }
```

Fetch Product by Slug

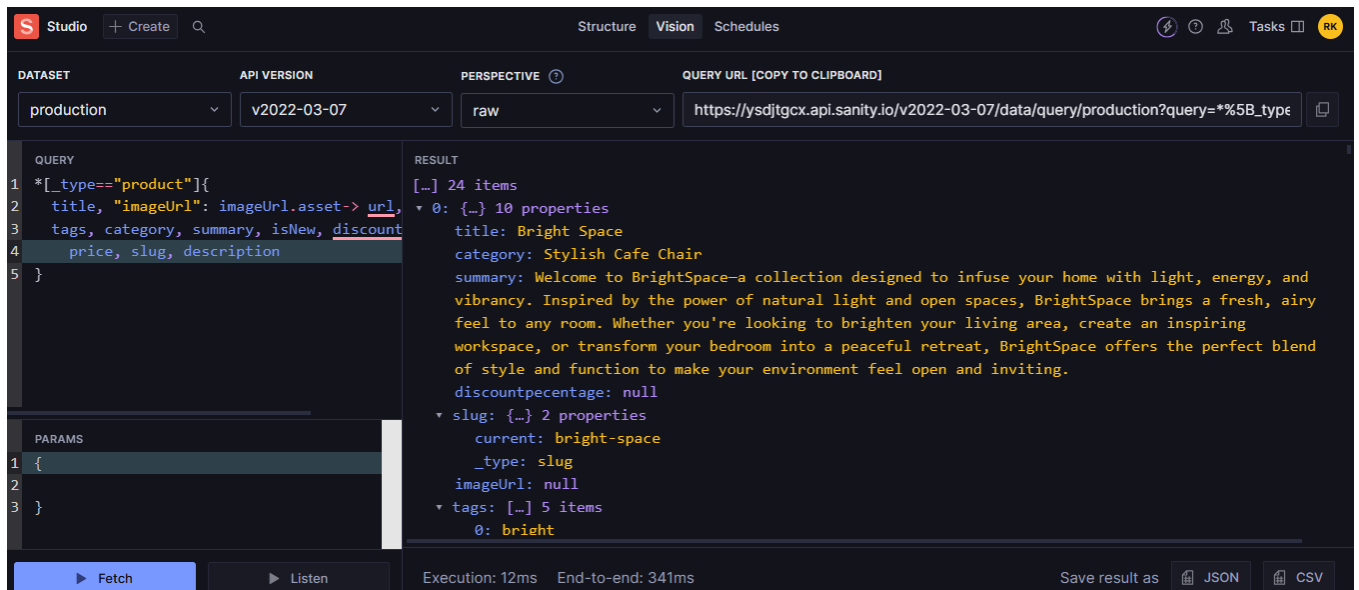
[illegible]

Screenshots

1. Data Import

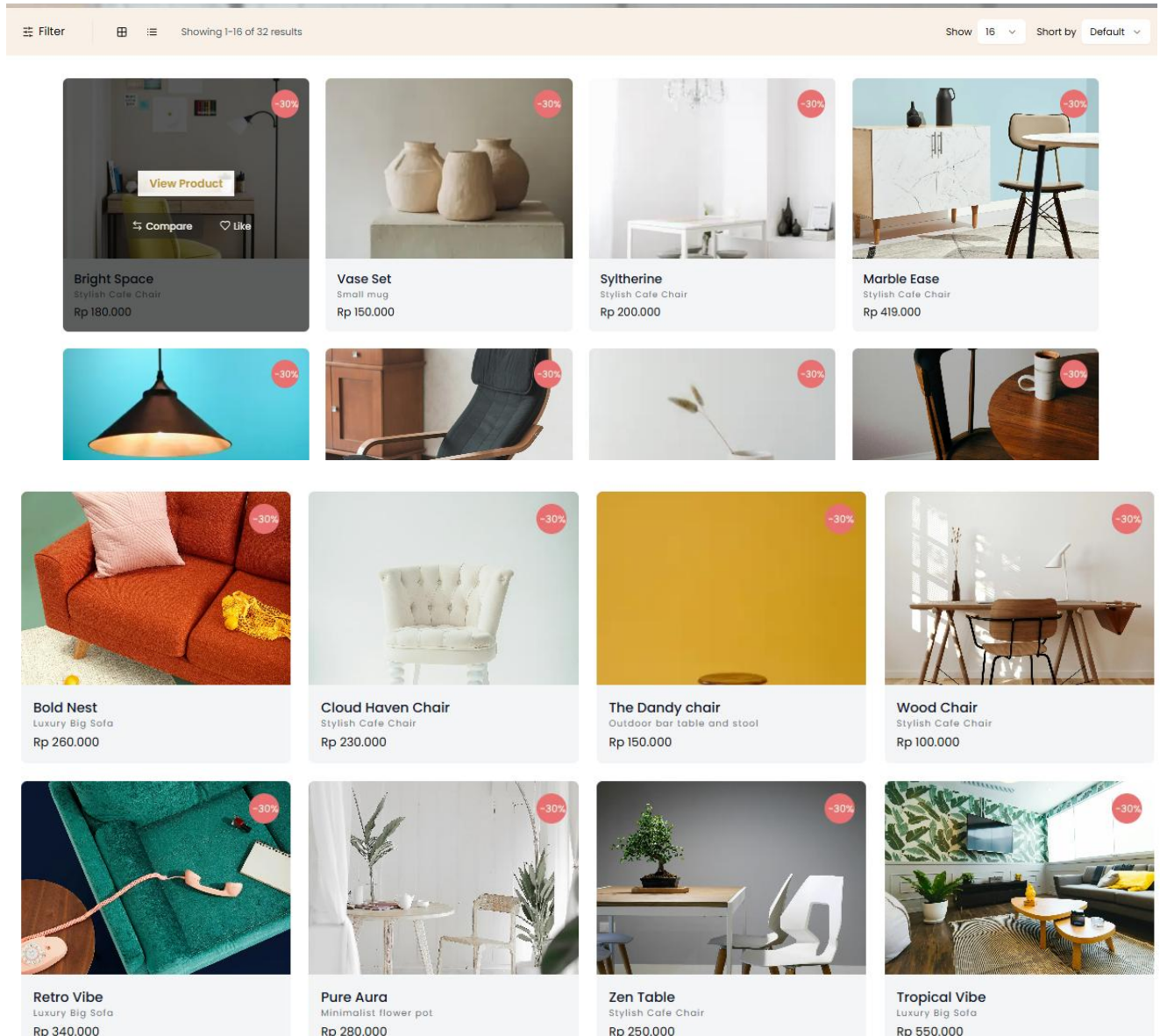


2. API Call Response



3. Frontend Display

Screenshot of Furniro products rendered dynamically on the website.



Conclusion:

The integration established a seamless process for importing, validating, and displaying Furniro product data from the API into Sanity CMS. Custom schemas and migration scripts ensured data consistency and project alignment. Future steps include enhancing API queries and implementing advanced features like search and filtering.