

Day 03 Data Migration And API Integration (Hekto Furniture)

1. Introduction:

The documents the integration of migration external data from a REST API into the sanity CMS for “HEKTO” furniture e-commerce platform. The goal is to automate content management for product such as chair, sofa, bed etc.

Key objectives include:

- Fetch product category and data from an external API.
- Structure and migration the data to sanity CMS.
- Display the migration data dynamically on the frontend.

2. API Integration:

- **Data Fetched:**
 - ♦ **Products:** Detail such as product name, price, description, image, stock, image.
 - ♦ **Category:** Furniture categories like Sofa, Chair etc
- **API Name:**

The external API, <https://next-ecommerce-template-4.vercel.app/api/product>.

3. Schema Structure:

The schema was enhance to better represent furniture Product detail. Key update include:

- Name (string)
- Price (number)
- Description (string)
- DiscountPercentage (number)
- StockLevel (number)
- IsFeaturedProduct (Boolean)
- Category (reference)
- Image (image)

Snippet of Product Schema:

TS product.ts X

next-e-commerce-template-4 > src > sanity > schemaTypes > TS product.ts > [⌕] default > fields

```
1  export default {
2    name: 'product',
3    type: 'document',
4    title: 'Product',
5    fields: [
6      {
7        name: 'name',
8        type: 'string',
9        title: 'Name',
10       validation: (Rule: any) => Rule.required().error('Name is required'),
11     },
12     {
13       name: 'image',
14       type: 'image',
15       title: 'Image',
16       options: {
17         hotspot: true,
18       },
19       description: 'Upload an image of the product.',
20     },
21     {
22       name: 'price',
23       type: 'string',
24       title: 'Price',
25       validation: (Rule: any) => Rule.required().error('Price is required'),
26     },
27     {
28       name: 'description',
29       type: 'text',
30       title: 'Description',
31       validation: (Rule: any) =>
32         Rule.max(150).warning('Keep the description under 150 characters.'),
33     },
34     {
35       name: 'discountPercentage',
36       type: 'number',
37       title: 'Discount Percentage',
38       validation: (Rule: any) =>
39         Rule.min(0).max(100).warning('Discount must be between 0 and 100.'),
40     },
41     {
42       name: 'isFeaturedProduct',
43       type: 'boolean',
44       title: 'Is Featured Product',
45     },
46   ]
47 }
```

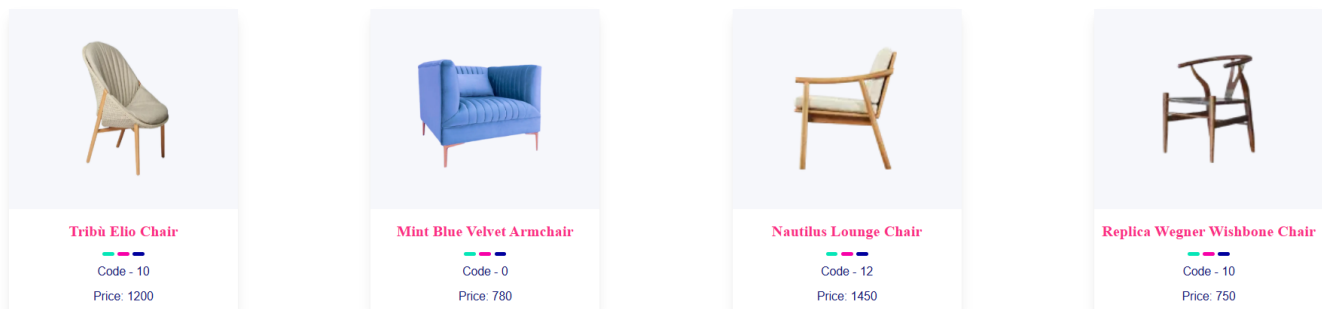
4. Migration Steps:

- **Setup Environment :**
Configured .env .local for secure storage for API
- **Sanity Client Setup:**
Create a Sanity client instance using project ID, Dataset and API token

```
TS product.ts  TS client.ts M X
next-e-commerce-template-4 > src > sanity > lib > TS client.ts > ...
1  import { createClient } from 'next-sanity'
2
3  import { apiVersion, dataset, projectId } from '../env'
4
5
6
7  export const client = createClient({
8    projectId,
9    dataset,
10   apiVersion,
11   useCdn: true, // Set to false if statically generating pages, using ISR or tag-based revalidation
12 })
13
14
15
```

5. Data Successfully Display In Frontend Featured Product

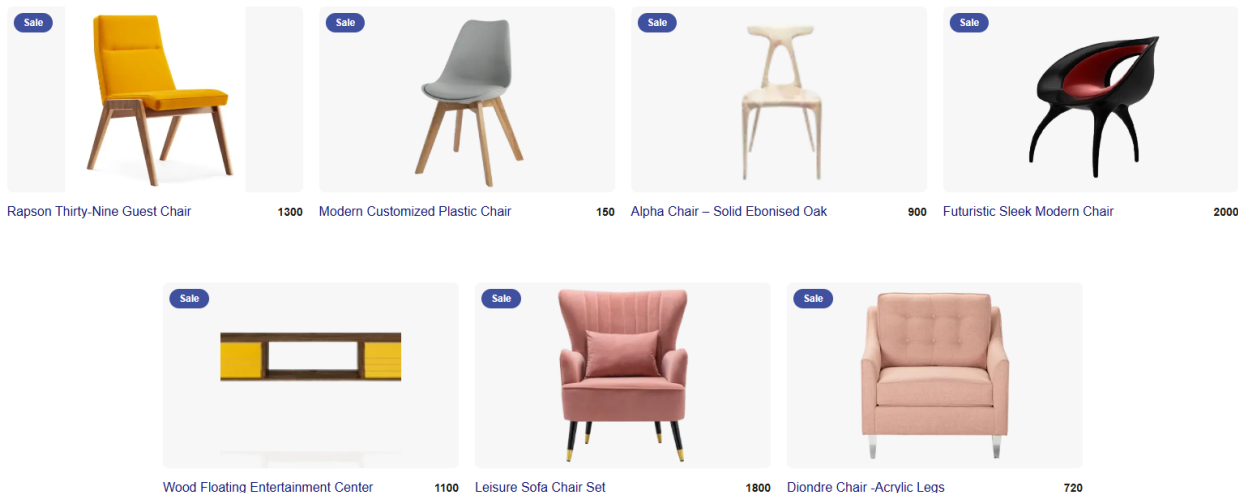
Featured Products



Latest Product

Latest Products

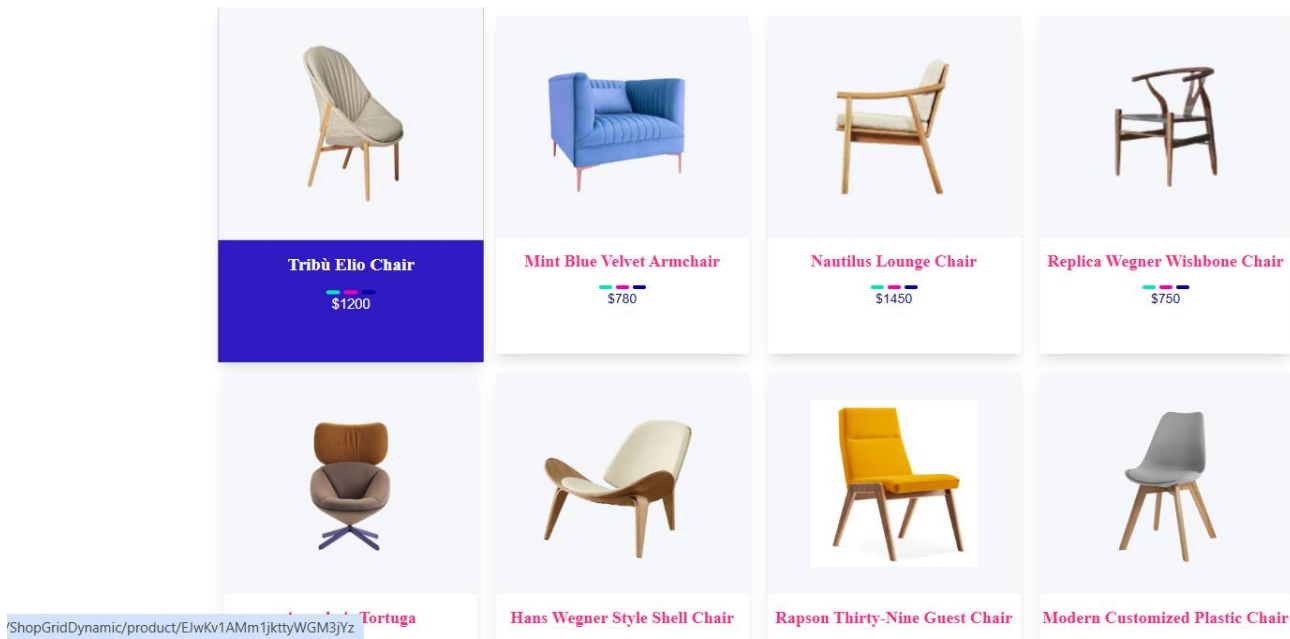
New Arrival Best Seller Featured Special Offers



6. API Integration and Migration Snipped:

```
TS product.ts   TS client.ts M   JS import-data.mjs X
next-ecommerce-template-4 > src > scripts > JS import-data.mjs > importData > [e] sanityItem

Tabnine | Edit | Test | Explain | Document
36  async function importData() {
37    try {
38      console.log('Fetching Product Data From API ...');
39
40      const response = await axios.get("https://next-ecommerce-template-4.vercel.app/api/product")
41      const products = response.data.products;
42
43      for (const item of products) {
44        console.log(`Processing Item: ${item.name}`);
45
46        let imageRef = null;
47        if (item.imagePath) {
48          imageRef = await uploadImageToSanity(item.imagePath);
49        }
50
51        const sanityItem = {
52          _type: 'product',
53          name: item.name,
54          category: item.category || null,
55          price: item.price,
56          description: item.description || '',
57          discountPercentage: item.discountPercentage || 0,
58          stockLevel: item.stockLevel || 0,
59          isFeaturedProduct: item.isFeaturedProduct,
60          image: imageRef
61            ? {
62              _type: 'image',
63              asset: {
64                _type: 'reference',
65                _ref: imageRef,
66              },
67            }
68            : undefined,
69        };
70
71        console.log(`Uploading ${sanityItem.category} - ${sanityItem.name} to Sanity !`);
72        const result = await client.create(sanityItem);
73        console.log(`Uploaded Successfully: ${result._id}`);
74        console.log("-----")
75        console.log("\n\n")
76      }
77
78      console.log('Data Import Completed Successfully !');
```



7. Final Checklist

| API Understanding | Schema Validation | Data Migration | API Intergration In Nextjs | Submission Prepartion |
|-------------------|-------------------|----------------|----------------------------|-----------------------|
| ✓ | ✓ | ✓ | ✓ | ✓ |