

Day 3 - API Integration Report-Dukandaro

● API INTEGRATION REPORT

1. API Integration Process

- Reviewed the provided documentation to understand the requirements for Template 4.
- Used the Thunder Client tool to test and verify API responses.
- Developed a migration script (`migrate.mjs`) to handle the data upload from the API to Sanity:
- Fetches data from the external API.
- Processes and uploads product data, including images, to Sanity.

2. Adjustments Made to Schemas

- Compared the existing schema with the provided schema.
- Identified differences and implemented necessary changes to align with the new requirements.

3. Migration Steps and Tools Used

- Developed a migration script (`migrate.mjs`) to transfer API data into Sanity, ensuring smooth data integration.
- Verified the successful migration of data, including images, by running the migration script.
- Created a helper function to handle image uploads and streamline data insertion into Sanity.
- Successfully uploaded data to Sanity and used the helper function to insert the fetched data into the project's template.

● Screen shots

1- Api Response

GET

http://localhost:3000/api/product?category=featured

Send

Query

Headers²

Auth

Body

Tests

Pre Run

Query Parameters

☒ category

featured

☐ parameter

value

Status: 200 OK

Size: 1.89 KB

Time: 385 ms

Response

Headers⁵

Cookies


Results

Docs

```
1 {
2   "success": true,
3   "products": [
4     {
5       "_id": "AJVJVObFUPWewZSb1SKDd7",
6       "rating": 2,
7       "badge": null,
8       "image": "https://cdn.sanity.io/images/ga0r9g3q/production/fe502c91678776be52471704ed6aa26cd1d28dd8-586x426.png",
9       "shipment": {
10        "weight": {
11          "value": 0.5,
12          "unit": "ounce"
13        },
14        "dimensions": {
15          "height": 0.5,
16          "width": 3,
17          "length": 6,
18          "unit": "inch"
19        }
20      },
21      "name": "Cantilever Chair - Durable",
22      "description": "A durable cantilever chair with a classic design, ideal for home or office use.",
23      "price": 42,
24      "prevPrice": null,
25      "code": "YS2301",
26      "category": "Featured"
27    },
28    {
29      "_id": "aWt125Cyehv2YfnCtyGS8i",
30      "name": "Cantilever Chair - Stylish",
31      "description": "A stylish cantilever chair with a sturdy frame and comfortable",
32      "price": 42,
33      "rating": 3,
34      "code": "YS2301",
35      "image": "https://cdn.sanity.io/images/ga0r9g3q/production/ac53427cec2feb66129667973939b2ea5191c695-500x500.png",
36      "prevPrice": null
37    }
38  ]
39 }
```

Response


2- Frontend



Classic Sturdy Chair

\$26.00

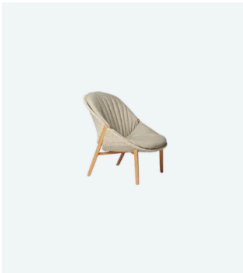
\$42.00



Elegant Simplicity Chair

\$26.00


\$42.00



Velvet Elegance Chair

\$26.00

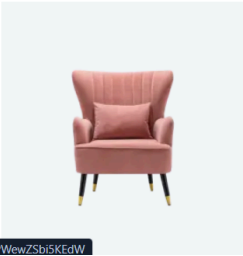
\$42.00




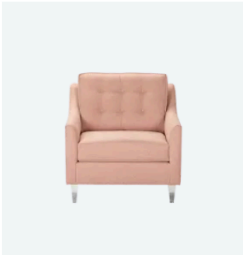
Timeless Design Chair


\$26.00

\$42.00



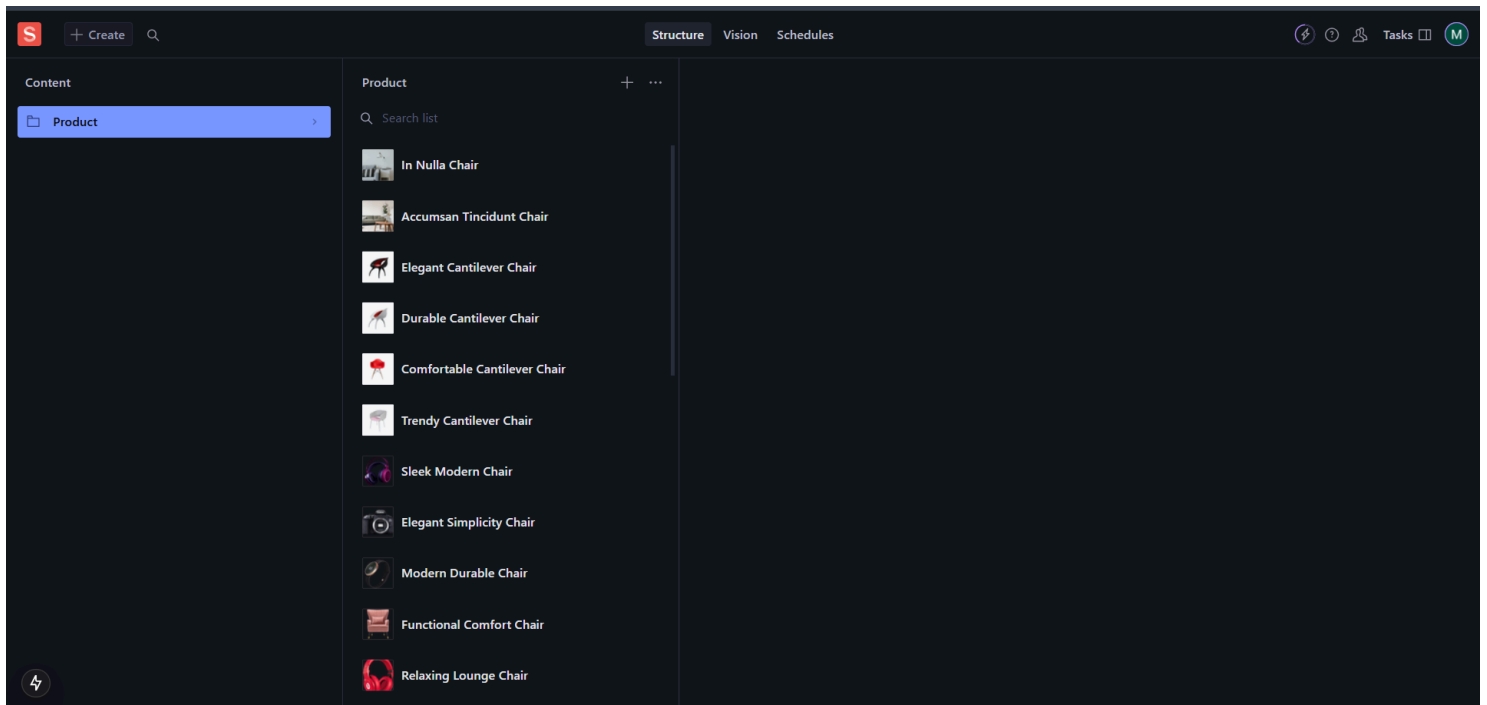






localhost:3000/product/AJVJVObFUPWewZSb1SKEdW

3- Sanity Data display



• Code snippets

1- code of Migration

```
1  const uploadImage = async (imageUrl) => {
2    const res = await fetch(imageUrl);
3    const bufferImage = await res.buffer();
4    const image = await client.assets.upload("image", bufferImage);
5    return image;
6  };
7
8  // Function to upload a single product to Sanity
9  const uploadProduct = async (productData) => {
10   // Upload the product image
11   const image = await uploadImage(productData.image);
12
13   // Create the product document
14   const product = {
15     _type: "product",
16     name: productData.name,
17     description: productData.description,
18     image: {
19       _type: "image",
20       asset: {
21         _type: "reference",
22         _ref: image._id,
23       },
24     },
25     price: productData.price,
26     badge: productData.badge,
27     code: productData.code,
28     category: productData.category,
29     rating: productData.rating,
30     prevPrice: productData.prevPrice,
31     shipment: productData.shipment,
32   };
33
34   // Upload the product to Sanity
35   const newProduct = await client.create(product);
36   console.log("Uploaded Product:", newProduct);
37   return newProduct;
38 };
39
40 // Function to upload all products
41 const uploadAllProducts = async () => {
42   for (const product of allProducts) {
43     const uploadedProduct = await uploadProduct(product);
44     console.log(uploadedProduct);
45   }
46 };
47
48 await uploadAllProducts();
49
```

2- Fetching Api Code

```
1 export const GET = async (req: NextRequest) => {
2   try {
3     const category = req.nextUrl.searchParams.get("category");
4     const products =
5       await client.fetch(`*[_type = "product" && category = "${category}"]{
6         _id,
7         name,
8         description,
9         price,
10        rating,
11        prevPrice,
12        badge,
13        code,
14        category,
15        "image": image.asset->url,
16        shipment {
17          weight {
18            value,
19            unit
20          },
21          dimensions {
22            height,
23            width,
24            length,
25            unit
26          }
27        }
28      }`);
29     console.log(products);
30     return NextResponse.json({ success: true, products });
31   } catch (error) {
32     console.log(error);
33     return NextResponse.json(
34       { success: false, error: error.message },
35       { status: 500 }
36     );
37   }
38 };
```