

## **DAY-3 API INTEGRATION REPORT –ThriftTreasures**

### API Integration Process:

#### **Overview**

The process of fetching data from an external API, storing it in Sanity, and using that data in a Next.js project. It covers environment setup, configuring Sanity, and uploading images.

#### **Environment Setup:**

1. Install dependencies : `npm install axios @sanity/client`
2. API END POINT: <https://template6-six.vercel.app/api/products>
3. API provides **schema** for the data, including fields like name, title, price, and image and description.

#### **Key Variables:**

`SANITY_PROJECT_ID=<your-project-id>`

`SANITY_DATASET=<your-dataset>`

`SANITY_API_TOKEN=<your-api-token>`

#### **Error Handling**

`try-catch` block to handle errors during data fetching.

#### **Schema Adjustments**

1. **Field Renaming:**
  - a. "productImage" was renamed to "image".
  - b. "tags" was removed, and "discountPercentage" and "isNew" were excluded.
2. **Field Adjustments:**
  - a. Renamed fields (e.g., "name" instead of "title").
  - b. Validation for price now includes a minimum value of 0.
3. **Image Field Enhancement:**
  - a. Added an **options** property with `hotspot: true` to enable image cropping.

#### **SCHEMA SCREENSHOT:**

src > sanity > schemaTypes > TS product.ts > product > fields > type

```
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: "name",
10       title: "Name",
11       type: "string",
12       validation: (Rule) => Rule.required(),
13     },
14     {
15       name: "title",
16       title: "Title",
17       type: "string",
18       validation: (Rule) => Rule.required(),
19     },
20     {
21       name: "description",
22       title: "Description",
23       type: "text",
24       validation: (Rule) => Rule.required(),
25     },
26     {
27       name: "image", // Change field name to productImage
28       title: "Product Image",
29       type: "image", // Sanity image asset field
30       options: {
31         hotspot: true, // Enables image cropping
32       },
33     },
34     {
35       name: "price",
36       title: "Price",
37       type: "number",
38       validation: (Rule) => Rule.required().min(0),
39     },
40   ],
41 });
42
```

## Migration Steps and Tools Used

### 1. Installed Libraries:

- Installed @sanity/client and axios for API integration and data handling.

### 2. Created env.local File:

- Stored Sanity keys and sensitive information securely in .env.local.

### 3. Retrieved Product Data:

- Fetches product data from an external API using axios.

### 4. Downloaded Images:

- Used axios to download images and save them to Sanity as assets.

## API CALL OUTPUT:

```
name: 'DANVOUY Womens T Shirt Casual Cotton Short',
price: 12.99,
title: 'T-shirt',
description: '95%Cotton,5%Spandex, Features: Casual, Short Sleeve, Letter Print,V-Neck,Fashion Tees, The fabric is soft and has some stretch., Occasion: Casual/Office/Beach/School/Home/Street. Season: Spring,Summer,Autumn,Winter.',
imageUrl: 'https://cdn.sanity.io/images/25suzqcl/production/833db4d824cd2772361abfe16f4d739dcaae42ca-679x761.jpg'
},
{
  title: 'Cotton Jacket',
  description: 'great outerwear jackets for Spring/Autumn/Winter, suitable for many occasions, such as working, hiking, camping, mountain/rock climbing, cycling, traveling or other outdoors. Good gift choice for you or your family member. A warm hearted love to Father, husband or son in this thanksgiving or Christmas Day.',
  imageUrl: 'https://cdn.sanity.io/images/25suzqcl/production/5619a5fa8fee4d3f36f7c3e3b9eac81e601c8c61-679x766.jpg',
  name: 'Mens Cotton Jacket',
  price: 55.99
},
{
  imageUrl: 'https://cdn.sanity.io/images/25suzqcl/production/48385f45f0abc12b5a0b24c39fef6f3f8baf4838-692x879.jpg',
  name: 'Mens Casual Slim Fit',
  price: 15.99,
  title: 'Mens Casual',
  description: 'The color could be slightly different between on the screen and in practice. / Please note that body builds vary by person, therefore, detailed size information should be reviewed below on the product description.'
},
{
  description: 'From our Legends Collection, the Naga was inspired by the mythical water dragon that protects the ocean's pearl. Wear facing inward to be bestowed with love and abundance, or outward for protection.',
  imageUrl: 'https://cdn.sanity.io/images/25suzqcl/production/918a44e944bf99fcd7c73c85c89d8bd5d609104-370x640.jpg',
  name: 'John Hardy Women's Legends Naga Gold & Silver Dragon Station Chain Bracelet',
  price: 695,
  title: 'Chain Bracelet'
},
{
  imageUrl: 'https://cdn.sanity.io/images/25suzqcl/production/74256ec87298d5e8303a2e92358c893738977af7-640x333.jpg',
  name: 'Solid Gold Petite Micropave ',
  price: 108,
  title: 'Micropave ',
  description: 'Satisfaction Guaranteed. Return or exchange any order within 30 days.Designed and sold by Hafeez Center in the United States. Satisfaction Guaranteed. Return or exchange any order within 30 days.'
},
{
  name: 'White Gold Plated Princess',
  price: 9.99,
  title: 'Plated Princess',
  description: 'Classic Created Wedding Engagement Solitaire Diamond Promise Ring for Her. Gifts to spoil your love more for Engagement, Wedding, Anniversary, Valentine's Day...',
  imageUrl: 'https://cdn.sanity.io/images/25suzqcl/production/d47de256189cf2cf23d2093674a881889b36a704-540x640.jpg'
},
{
  price: 10.99,
  title: 'Pierced Owl',
  description: 'Rose Gold Plated Double Flared Tunnel Plug Earrings. Made of 316L Stainless Steel',
  imageUrl: 'https://cdn.sanity.io/images/25suzqcl/production/e085a7e29cb4f47ee44b2b8ce660e9c028e100-640x369.jpg',
  name: 'Pierced Owl Rose Gold Plated Stainless Steel Double'
},
{
  name: 'WD 2TB Elements Portable External Hard Drive - USB 3.0 ',
  price: 64,
  title: 'hard drive',
  description: 'USB 3.0 and USB 2.0 Compatibility Fast data transfers Improve PC Performance High Capacity; Compatibility Formatted NTFS for Windows 10, Windows 8.1, Windows 7; Reformatting may be required for other operating systems; Compatibility may vary depending on user's hardware configuration and operating system',
  imageUrl: 'https://cdn.sanity.io/images/25suzqcl/production/44c029d778597d3fa5f8bb61ee816a0cf1b1624-653x879.jpg'
}
]
GET / 200 in 433ms
```

## FRONTEND DISPLAY:

Featured Products

## BESTSELLER PRODUCTS

Problems trying to resolve the conflict between



**BIYLACLESEN Women's  
3-in-1 Snowboard Jacket**  
Winter Coats

Jacket Winter Coats

\$56.99



**Rain Jacket Women  
Windbreaker Striped  
Climbing Raincoats**

Rain Jacket Women

\$39.99



**MBJ Women's Solid  
Short Sleeve Boat Neck  
V**

Short Sleeve Boat Neck V

\$9.85



**Opna Women's Short  
Sleeve Moisture**












Women's Short Sleeve

Moisture

\$7.95



**SANITY CMS FIELDS:**

Content	Product <span>+</span>
<div> <div>Product</div> </div>	<div> <div> <div>Q</div> <div>Search list</div> </div> <div> <div> <div></div> <div>Pierced Owl</div> </div> <div> <div></div> <div>Faux Leather</div> </div> <div> <div></div> <div>Micropave</div> </div> <div> <div></div> <div>T-shirt</div> </div> <div> <div></div> <div>flash</div> </div> <div> <div></div> <div>hard drive</div> </div> <div> <div></div> <div>Chain Bracelet</div> </div> <div> <div></div> <div>T-Shirt</div> </div> <div> <div></div> <div>Backpack</div> </div> <div> <div></div> <div>Mens Casual</div> </div> <div> <div></div> <div>Women's Short Sleeve Moisture</div> </div> </div> </div>

**CODE SNIPPET:**

**API INTEGRATION:**

```
const response = await axios.get('https://template6-six.vercel.app/api/products');
const products = response.data;
```

## SNIPPETS OF MIGRATION SCRIPS:

```
import axios from 'axios';
import { client } from './sanityClient.js';
async function uploadImageToSanity(imageUrl: string): Promise<string> {
  try {
    // Fetch the image from the URL and convert it to a buffer
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);
    // Upload the image to Sanity
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop(), // Extract the filename from URL
    });
    // Debugging: Log the asset returned by Sanity
    console.log('Image uploaded successfully:', asset);

    return asset._id; // Return the uploaded image asset reference ID
  } catch (error) {
    console.error('❌ Failed to upload image:', imageUrl, error);
    throw error;
  }
}

async function importData() {
  try {
    // Fetch data from external API
    const response = await axios.get('https://template6-six.vercel.app/api/products');
    const products = response.data;

    // Iterate over the products
    for (const product of products) {
      let imageRef = '';

      // Upload image and get asset reference if it exists
      if (product.image) {
        imageRef = await uploadImageToSanity(product.image);
      }

      const sanityProduct = {
        _id: `product-${product.id}`, // Prefix the ID to ensure validity
        _type: 'product',
        name: product.title,
        price: product.price,
        discountPercentage: product.discountPercentage || 0,
        tags: product.category ? [product.category] : [],
        image: {
          _type: 'image',
          asset: {
            _type: 'reference',
            _ref: imageRef, // Set the correct asset reference ID
          },
        },
        description: product.description,
        rating: product.rating?.rate || 0,
        ratingCount: product.rating?.count || 0,
      };

      // Log the product before attempting to upload it to Sanity
      console.log('Uploading product:', sanityProduct);
      // Import data into Sanity
      await client.createOrReplace(sanityProduct);
      console.log('✅ Imported product: ${sanityProduct.name}');
    }
    console.log('✅ Data import completed!');
  } catch (error) {
    console.error('❌ Error importing data:', error);
  }
}

importData();
```

## FINAL CHECK LIST:

API UNDERSTANDING	SCHEMA VALIDATION	DATA MIGRATION	API INTEGRATION IN NEXTJS	SUBMISSION PREPARATION
-------------------	-------------------	----------------	---------------------------	------------------------

✓	✓	✓	✓	✓
---	---	---	---	---