

# **Day 3 - API Integration And Data Migration- [Nike Ecommerce]**

## **1. API Integration Process**

### **Objective:**

The goal was to integrate product data from an external API into Sanity CMS and display the data on the frontend of the marketplace application.

### **API Source:**

The product data was fetched from an external API:

URL: <https://template-03-api.vercel.app/api/products>

## **Integration Process:**

### **Data Fetching:**

The product data was fetched from the external API using Axios for HTTP requests. The data included various details such as product name, category, price, inventory, etc.

### **Sanity Client Configuration:**

I created a Sanity client using the provided credentials from the .env.local file to interact with the Sanity CMS.

### **Schema Adjustments:**

The schema in Sanity was adjusted to match the structure of the incoming data from the API, allowing fields like productName, price, category, etc., to be stored in Sanity.

### **Data Migration Script:**

A custom Node.js migration script was written to automate the process of uploading data from the API to Sanity CMS, including image uploads.

### **API Integration Steps:**

Fetches product data from the API.

Uploaded the data to Sanity using the Sanity client.

Used the Sanity API to upload images by downloading them and referencing them within the schema.

Rendered the data on the frontend using the Next.js framework.

## **2. Adjustments Made to Schemas**

The Sanity product schema was adjusted to ensure compatibility with the product data from the external API. The following fields were added:

productName (string)

category (string)

price (number)

inventory (number)

colors (array of strings)

status (string)

image (image type)

description (text)

**Updated schema in Sanity:**

```

export const productSchema = {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'productName',
      title: 'Product Name',
      type: 'string',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'inventory',
      title: 'Inventory',
      type: 'number',
    },
    {
      name: 'colors',
      title: 'Colors',
      type: 'array',
      of: [{ type: 'string' }],
    },
    {
      name: 'status',
      title: 'Status',
      type: 'string',
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image',
      options: {
        hotspot: true,
      },
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
  ],
};

```

### 3. Migration Steps and Tools Used

#### Migration Process:

**Data Fetching:**

-Used Axios to fetch the product data from the external API.

**Image Uploading:**

-For products with images, images were uploaded to Sanity using the asset upload feature, where images were fetched and uploaded as buffer data.

**Sanity Client:**

-Used Sanity client to create new documents within the CMS using the data received from the API.

**Migration Script:**

-The script (importSanityData.mjs) was written to automate the migration of product data and images.

**Tools Used:**

**Axios:** For making HTTP requests to the external API.

**Sanity Client:** For pushing the data to Sanity CMS.

**dotenv:** For managing environment variables securely.

## **Migration Script**

```

import axios from 'axios';
import { createClient } from '@sanity/client';
import dotenv from 'dotenv';

dotenv.config(); // Load environment variables

const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  apiVersion: '2021-08-31',
  useCdn: false,
  token: process.env.SANITY_API_TOKEN,
});

async function fetchProducts() {
  const response = await axios.get('https://template-03-api.vercel.app/api/products');
  return response.data.data;
}

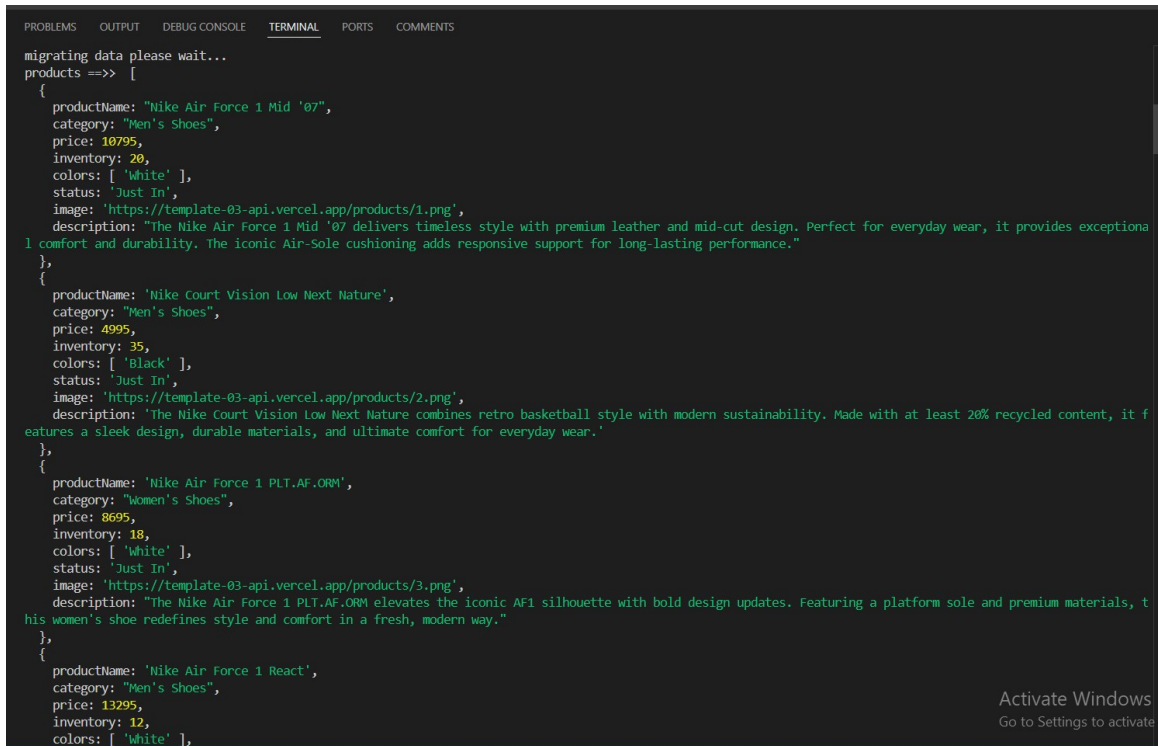
async function importData() {
  const products = await fetchProducts();
  for (const product of products) {
    await client.create({
      _type: 'product',
      productName: product.productName,
      category: product.category,
      price: product.price,
      inventory: product.inventory,
      colors: product.colors,
      status: product.status,
      description: product.description,
      image: product.image,
    });
  }
}

importData();

```

## 4. Screenshots

## API Calls:



The screenshot shows a VS Code terminal window with the 'TERMINAL' tab selected. The terminal output displays a JSON array of product data for Nike shoes. The data includes product names, categories, prices, inventory levels, colors, status, image URLs, and descriptions. The products listed are Nike Air Force 1 Mid '07, Nike Court Vision Low Next Nature, Nike Air Force 1 PLT.AF.ORM, and Nike Air Force 1 React. The terminal also shows a message 'migrating data please wait...' and 'products ==>'. An 'Activate Windows' watermark is visible in the bottom right corner of the terminal window.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
migrating data please wait...
products ==> [
  {
    productName: "Nike Air Force 1 Mid '07",
    category: "Men's Shoes",
    price: 10795,
    inventory: 20,
    colors: [ 'White' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/1.png',
    description: "The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and mid-cut design. Perfect for everyday wear, it provides exceptional comfort and durability. The iconic Air-Sole cushioning adds responsive support for long-lasting performance."
  },
  {
    productName: 'Nike Court Vision Low Next Nature',
    category: "Men's Shoes",
    price: 4995,
    inventory: 35,
    colors: [ 'Black' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/2.png',
    description: 'The Nike Court Vision Low Next Nature combines retro basketball style with modern sustainability. Made with at least 20% recycled content, it features a sleek design, durable materials, and ultimate comfort for everyday wear.'
  },
  {
    productName: 'Nike Air Force 1 PLT.AF.ORM',
    category: "Women's Shoes",
    price: 8695,
    inventory: 18,
    colors: [ 'White' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/3.png',
    description: "The Nike Air Force 1 PLT.AF.ORM elevates the iconic AF1 silhouette with bold design updates. Featuring a platform sole and premium materials, this women's shoe redefines style and comfort in a fresh, modern way."
  },
  {
    productName: 'Nike Air Force 1 React',
    category: "Men's Shoes",
    price: 13295,
    inventory: 12,
    colors: [ 'White' ],
    status: 'Just In',
    image: 'https://template-03-api.vercel.app/products/4.png',
    description: "The Nike Air Force 1 React features a full-length React foam sole for a soft, responsive feel. The classic AF1 design is updated with a mesh upper and a rubber outsole for durability and traction."
  }
]
```

## Data Successfully Displayed in Frontend:

## Products



**Nike Renew Run 3**  
Category: Men's Running Shoes  
Price: \$7295  
Inventory: 40  
Status: Trending  
The Nike Renew Run 3 offers plush cushioning and a breathable design for smooth, comfortable runs. Perfect for beginners or seasoned athletes, it delivers support and style for every step.



**Nike Metcon 8**  
Category: Men's Training Shoes  
Price: \$10295  
Inventory: 25  
Status: Best Seller  
The Nike Metcon 8 is built for intense training sessions. Featuring a stable base, durable materials, and excellent traction, it's the perfect companion for weightlifting, CrossFit, or HIIT workouts.



**Nike Air Force 1 Mid '07**  
Category: Men's Shoes  
Price: \$10795  
Inventory: 20  
Status: Just In  
The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and mid-cut design. Perfect for everyday wear, it provides exceptional comfort and durability. The iconic Air Sole cushioning adds responsive support for long-lasting performance.



**Nike Court Vision Low Next Nature**  
Category: Men's Shoes  
Price: \$4995  
Inventory: 35  
Status: Just In  
The Nike Court Vision Low Next Nature combines retro basketball style with modern sustainability. Made with at least 20% recycled content, it features a sleek design, durable materials, and ultimate comfort for everyday wear.



**Nike Air Force 1 PLTAFORM**  
Category: Women's Shoes  
Price: \$8695  
Inventory: 18  
Status: Just In  
The Nike Air Force 1 PLTAFORM elevates the iconic AF1 silhouette with bold design updates. Featuring a platform sole and premium materials, this women's shoe redefines style and comfort in a fresh, modern way.



**Nike Air Force 1 React**  
Category: Men's Shoes  
Price: \$13295  
Inventory: 12  
Status: Just In  
The Nike Air Force 1 React takes the classic AF1 to the next level with React technology. Experience unparalleled cushioning and modern design details for all-day comfort and a bold statement of style.



**Nike Standard Issue Basketball Jersey**  
Category: Women's Basketball Jersey  
Price: \$2895  
Inventory: 40  
Status: Just In  
The Nike Standard Issue Basketball Jersey delivers a lightweight, breathable fit for top performance. Designed for athletes, it combines style and functionality to keep you cool and comfortable on and off the court.



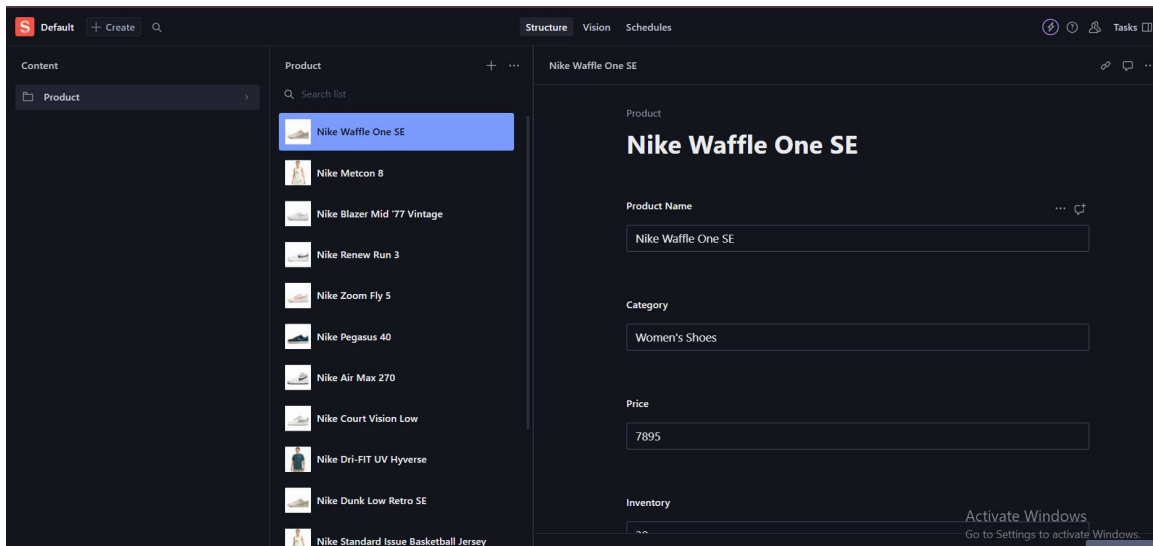
**Nike Dunk Low Retro SE**  
Category: Men's Shoes  
Price: \$9695  
Inventory: 15  
Status: Promo Exclusion  
The Nike Dunk Low Retro SE brings vintage vibes with a modern twist. Featuring premium materials and a neutral color palette, this classic sneaker is versatile, stylish, and perfect for any casual occasion.



**Nike Air Max 270**  
Category: Men's Shoes  
Price: \$13295  
Inventory: 20  
Status: Trending  
The Nike Air Max 270 features the largest Air unit yet for exceptional cushioning. Its bold design, breathable mesh upper, and vibrant colors make it a standout choice for style and comfort.

## Populated Sanity CMS Fields:





## 5. Code Snippets for API Integration and Migration Scripts:

### Sanity Client Setup (client.ts):

```
import { createClient } from '@sanity/client';
import imageUrlBuilder from '@sanity/image-url';

const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  apiVersion: '2021-08-31',
  useCdn: true,
});

const builder = imageUrlBuilder(client);

export const urlFor = (source: any) => builder.image(source);

export default client;
```

### Frontend Component (Displaying Products):

```

import { useEffect, useState } from 'react';
import client from '@lib/client';

const ProductsPage = () => {
  const [products, setProducts] = useState([]);

  useEffect(() => {
    const fetchProducts = async () => {
      const query = '*[_type == "product"]';
      const fetchedProducts = await client.fetch(query);
      setProducts(fetchedProducts);
    };

    fetchProducts();
  }, []);

  return (
    <div>
      <h1>Our Products</h1>
      <div className="products-list">
        {products.map((product) => (
          <div key={product._id} className="product-card">
            <img src={urlFor(product.image).url()} alt={product.productName} />
            <h2>{product.productName}</h2>
            <p>{product.description}</p>
            <p>Price: ${product.price}</p>
          </div>
        ))}
      </div>
    </div>
  );
};

export default ProductsPage;

```

## Conclusion:

This report outlines the API integration and data migration process that was carried out for my marketplace application. The Sanity schema was adjusted to accommodate the incoming product data, and a Node.js migration script was developed to automate the migration process. The product data is now successfully displayed on the frontend, and the CMS fields are populated with the data.

## Day 3 - API Integration Report Checklist

**1. API Understanding:**

✓

**2. Schema Validation:**

✓

**3. Data Migration:**

✓

**4. API Integration in Next.js:**

✓

**5. Submission Preparation:**

✓