

# Day 3-API Integration Report for Nike

## I.Introduction

This report outlines the integration process of product and category data from an external API into the backend system of Nike, a sportswear and equipment e-commerce platform. The integration leverages Sanity CMS for content management and Next.js for frontend rendering.

The main objectives of this integration were:

- Fetching data from an external API.
- Storing and managing the data in Sanity CMS.
- Displaying the fetched data dynamically on the frontend.

```
"success": true,
"data": [
  {
    "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. Provides 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. 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. Features a modern way."
  },
  {
    "productName": "Nike Air Force 1 React",
    "category": "Men's Shoes",
    "price": 13295,
    "inventory": 12
```

## 2 API Integration Process

Step 1: Fetching Data from API

The API provided the following endpoints:

- **Products Endpoint:** includes details such as product names, pricing, descriptions, categories, stock availability, and images.

Base API URL:

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

```

try {
  console.log(`Uploading image: ${imageUrl}`);
  const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
  const buffer = Buffer.from(response.data);
  const filename = imageUrl.split('/').pop() || `image_${Date.now()}.jpg`;
  const asset = await client.assets.upload('image', buffer, { filename });

  console.log(`Image uploaded successfully: ${asset._id}`);
  return asset._id;
} catch (error) {
  console.error('Failed to upload image:', imageUrl, error.message);
  return null;
}

```

*// Function to migrate data*

```

export async function importData() {
  try {
    console.log('Starting data migration...');

    // Fetch data from API
    const apiUrl = 'https://template-03-api.vercel.app/api/products';
    const response = await axios.get(apiUrl);
    const products = response.data.data;

    console.log("❤️")

    console.log('Products fetched successfully:', products);

    // Iterate over products and upload to Sanity
    for (const product of products) {
      try {
        console.log(`Processing product: ${product.productName}`);

        // Upload product image
        let imageRef = null;
        if (product.image) {

```

Chat (CTRL + I) / Edit

## Step 2: Migrating Data to Sanity CMS

The fetched data was structured and stored in Sanity CMS using its JavaScript client library.

The product schema was updated to include additional fields for more detailed data.

Key Schema Fields:

- **title:** Name of the product.
- **price:** Current price of the product.
- **category:** References the category associated with the product.
- **image:** Image URL for the product.
- **colors:** Available color options for the product.

🔍 Search list



Nike Waffle One SE



Nike Metcon 8



Nike Blazer Mid '77 Vintage



Nike Renew Run 3



Nike Zoom Fly 5



Nike Pegasus 40



Nike Air Max 270



Nike Court Vision Low



Nike Dri-FIT UV Hyverse



Nike Dunk Low Retro SE



Nike Standard Issue Basketball Jersey

## 3. Frontend Integration

### Step 3: Rendering Data on Frontend

The fetched data from the API was dynamically rendered on the frontend using Next.js. This involved creating components to display product details such as name, price, category, image, and available colors. The integration ensured a seamless user experience with real-time updates and responsive design.

# Product Display:

Below is a screenshot of the product listing page, showing products dynamically rendered on the frontend:

