

Day 3 - API Integration Report: Nike

API Integration Steps

Step 1: API Setup

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

Data Fields: Includes productName, category, price, inventory, colors, status, description, and image.

Tool Used: Axios was used to fetch data.

Step 2: Fetching Data

Code Example:

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

The API returns a list of products with all the necessary details.

Schema Updates

1. Fields Added:

Colors: Now supports multiple color options as an array of strings.

Image: Configured for better cropping and scaling using a Sanity image field.

2. Final Schema

```
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', // Using Sanity's image type for image field
    options: {
      hotspot: true,
    },
  },
  {
    name: 'description',
    title: 'Description',
    type: 'text',
  },
],
}
```

Steps to Migrate Data into Sanity

1. Set Up Environment

- a. Add API keys and tokens in `.env.local`.
- b. Import libraries: `@sanity/client`, `axios`, `dotenv`.

2. Create Sanity Client

- a. Use `createClient()` with project ID, dataset, token, and API version.

3. Fetch Data from API

- a. Use Axios to get product data from the source endpoint.

4. Upload Images

- a. Convert image URLs into buffers.
- b. Upload to Sanity using `client.assets.upload()`.

5. Prepare Data

- a. Map product fields (e.g., `productName`, `price`, `image`) to match Sanity schema.
- b. Include optional fields like `colors`.

6. Save Data in Sanity

- a. Use `client.create()` to add each product to Sanity.

Screenshots

API call:




```
Body Cookies Headers (15) Test Results 200 OK • 1.42 s • 7.11 KB
{} JSON Preview Visualize
17     "productName": "Nike Court Vision Low Next Nature",
18     "category": "Men's Shoes",
19     "price": 4995,
20     "inventory": 35,
21     "colors": [
22       "Black"
23     ],
24     "status": "Just In",
25     "image": "https://template-03-api.vercel.app/products/2.png",
26     "description": "The Nike Court Vision Low Next Nature combines retro basketball style with modern
27                   sustainability. Made with at least 20% recycled content, it features a sleek design, durable
28                   materials, and ultimate comfort for everyday wear."
29   },
30   {
31     "productName": "Nike Air Force 1 PLT.AF.ORM",
32     "category": "Women's Shoes",
33     "price": 8695,
```

Activate Windows
Go to Settings to activate Windows


Display Data in Frontend:

Discover Our Premium Collection


Browse through a variety of exclusive, high-quality products designed for ultimate comfort and style.




Nike Court Vision Low Next Nature
Men's Shoes
\$4995
Just In
[View Product](#)



Nike Air Force 1 PLT.AF.FORM
Women's Shoes
\$8695
Just In
[View Product](#)



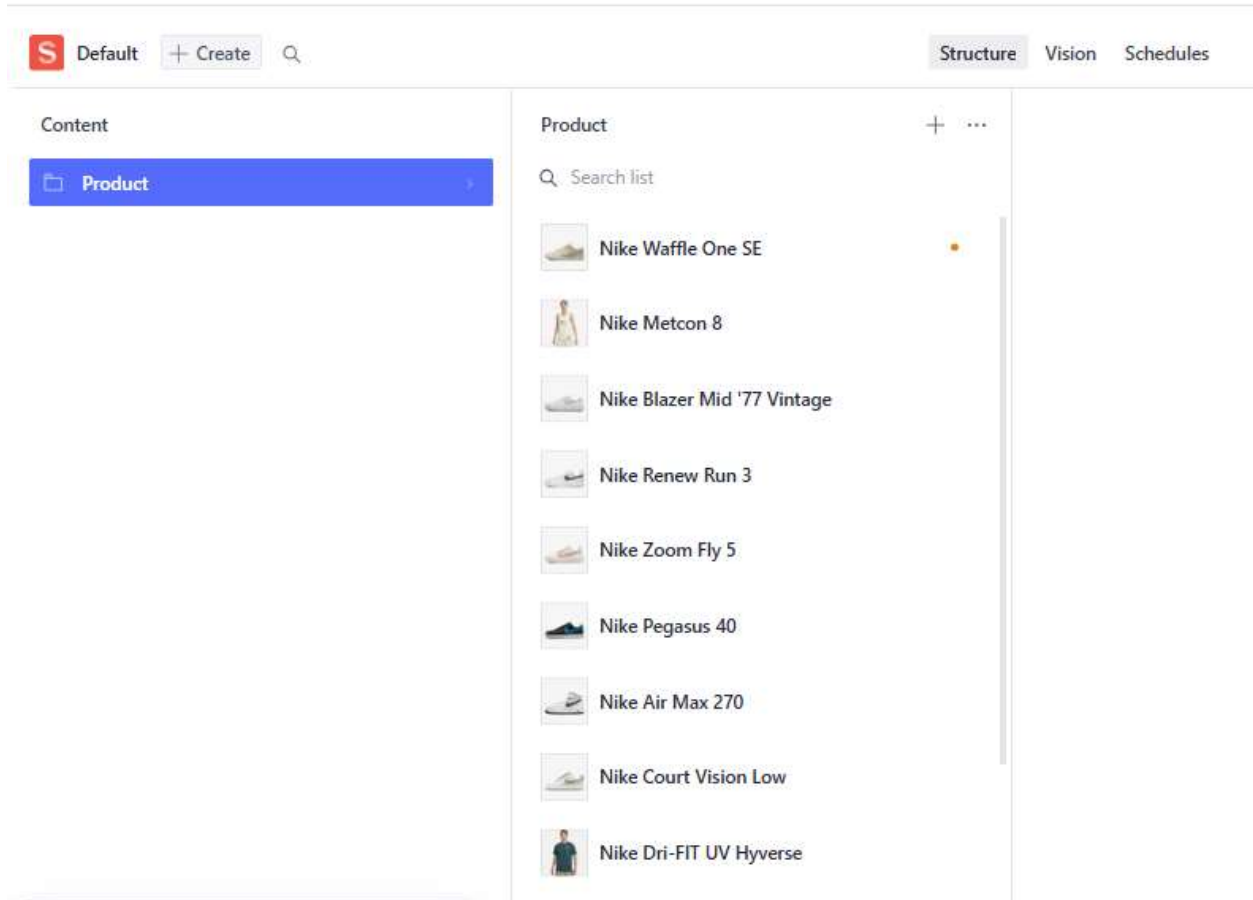
Air Jordan 1 Elevate Low
Women's Shoes
\$11895
Promo Exclusion
[View Product](#)



Nike Dri-FIT UV Hyverse
Men's Short-Sleeve Graphic Fitness Top
\$2495
Sustainable Materials
[View Product](#)

Activate Windows
Go to Settings to activate Windows.

Populated Sanity CMS Fields:



Code Snippets

Migration Scripts:

```
1 import { createClient } from '@sanity/client';
2 import axios from 'axios';
3 import dotenv from 'dotenv';
4 import { fileURLToPath } from 'url';
5 import path from 'path';
6
7 // Load environment variables from .env.local
8 const __filename = fileURLToPath(import.meta.url);
9 const __dirname = path.dirname(__filename);
10 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12 // Create Sanity client
13 const client = createClient({
14   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
15   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
16   useCdn: false,
17   token: process.env.SANITY_API_TOKEN,
18   apiVersion: '2021-08-31'
19 });
20
21
22 async function uploadImageToSanity(imageUrl) {
23   try {
24     console.log(`Uploading image: ${imageUrl}`);
25     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
26     const buffer = Buffer.from(response.data);
27     const asset = await client.assets.upload('image', buffer, {
28       filename: imageUrl.split('/').pop()
29     });
30     console.log(`Image uploaded successfully: ${asset._id}`);
31     return asset._id;
32   } catch (error) {
```

```

53
54     const sanityProduct = {
55       _type: 'product',
56       productName: product.productName,
57       category: product.category,
58       price: product.price,
59       inventory: product.inventory,
60       colors: product.colors || [], // Optional, as per your schema
61       status: product.status,
62       description: product.description,
63       image: imageRef ? {
64         _type: 'image',
65         asset: {
66           _type: 'reference',
67           _ref: imageRef,
68         },
69       } : undefined,
70     };
71
72     await client.create(sanityProduct);
73   }
74
75   console.log('Data migrated successfully!');
76 } catch (error) {
77   console.error('Error in migrating data ==>> ', error);
78 }
79 }
80
81 importData();

```

API Integration:

```

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

```

Final Checklist

API Understanding	Schema Validation	Data Migration	API Integration in Next.js	Submission Preparation
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>