

Day 3 - API Integration Report – Nike

API Integration Process

Step 1: API Endpoint Setup

- API Endpoint: `https://template-03-api.vercel.app/api/products`
- Data Structure: The API provides product data, including fields like ``productName``, ``category``, ``price``, ``inventory``, ``colors``, ``status``, ``description``, and ``image``.
- Library Used: Axios was used to fetch data from the API.

Step 2: Fetch Data from API

- **Code Snippet:**

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

- The API call retrieves an array of product objects from the endpoint, ensuring proper data serialization.

Adjustments Made to Schemas

Product Schema Modifications

1. **Field Updates:**

- ``colors``: Added as an optional array of strings to accommodate multiple color options.
- ``image``: Configured as a Sanity image field with hotspot enabled for better image cropping and scaling.

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',
      options: {
        hotspot: true,
      },
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
  ],
}
```

Migration Steps and Tools Used

Migration Steps

1. **Environment Setup:**

- Installed required dependencies: `@sanity/client`, `axios`, `dotenv`.
- Created a `.env.local` file to securely store environment variables.

2. **Data Fetching:**

- Retrieved product data from the API endpoint using Axios.
- Parsed and logged the data to confirm its structure and integrity.

3. **Image Upload:**

- Downloaded images from the API's `image` field using Axios.
- Uploaded images to Sanity's Asset Manager using the Sanity client.
- Code to upload images:

```
const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
const buffer = Buffer.from(response.data);
const asset = await client.assets.upload('image', buffer, {
  filename: imageUrl.split('/').pop()
});
```

4. **Document Creation:**

- Created Sanity documents for each product by combining API data and uploaded image references.

- Code snippet:

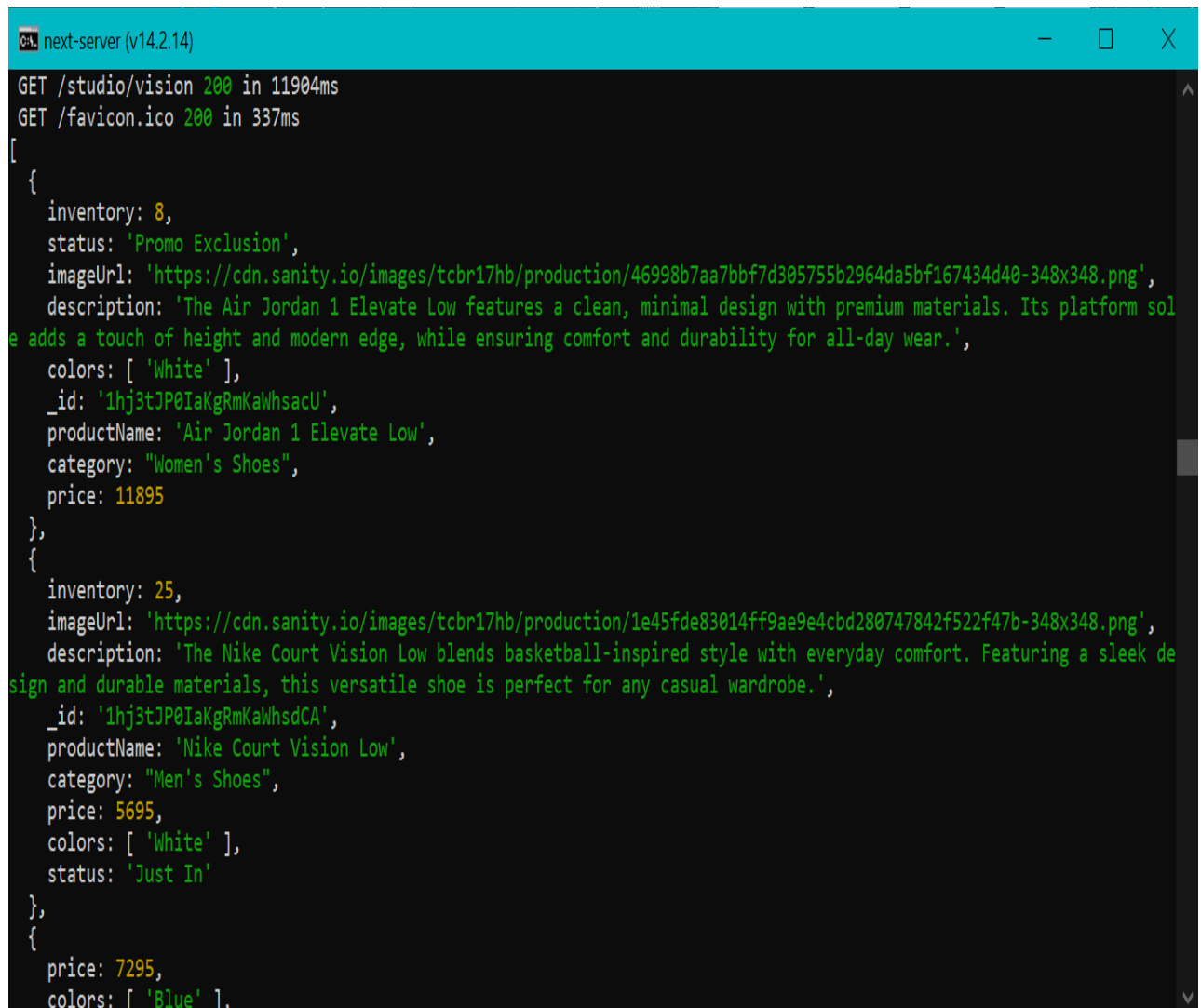
```
const sanityProduct = {
  _type: 'product',
  productName: product.productName,
  category: product.category,
  price: product.price,
  inventory: product.inventory,
  colors: product.colors || [],
  status: product.status,
  description: product.description,
  image: imageRef ? {
    _type: 'image',
    asset: {
      _type: 'reference',
      _ref: imageRef,
    },
  } : undefined,
};
await client.create(sanityProduct);
```

Tools Used

- ****Sanity Client:**** For interacting with the Sanity CMS.
- ****Axios:**** For making HTTP requests to fetch API data and images.
- ****Dotenv:**** To load environment variables from ``.env.local``.

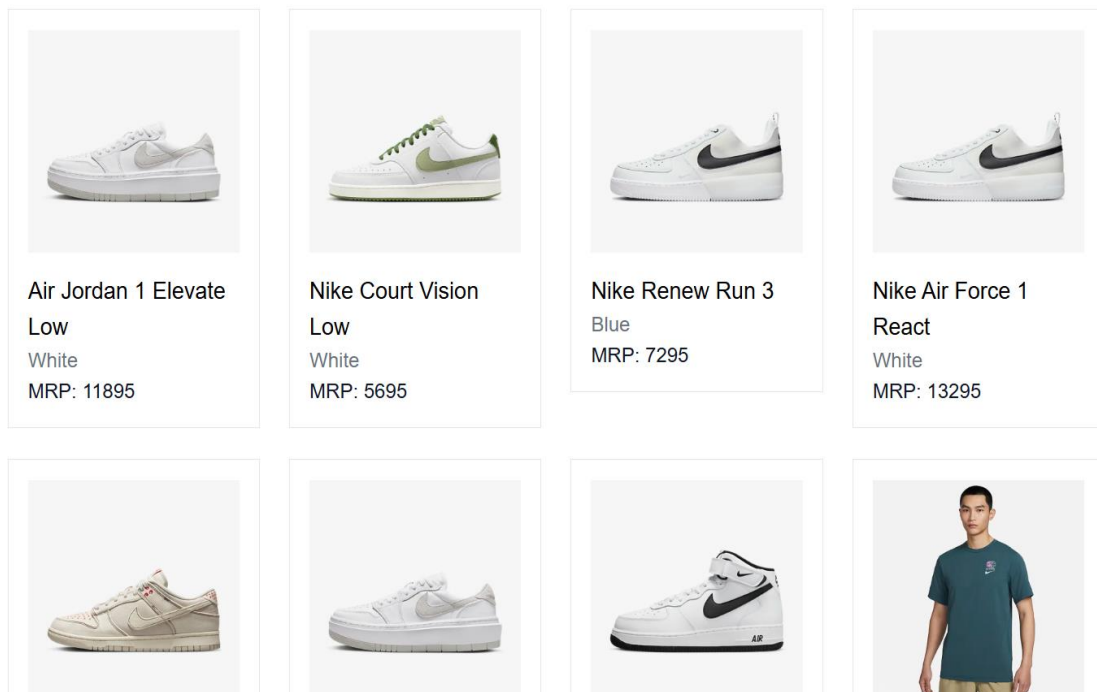
Screenshots

1. API Call Output

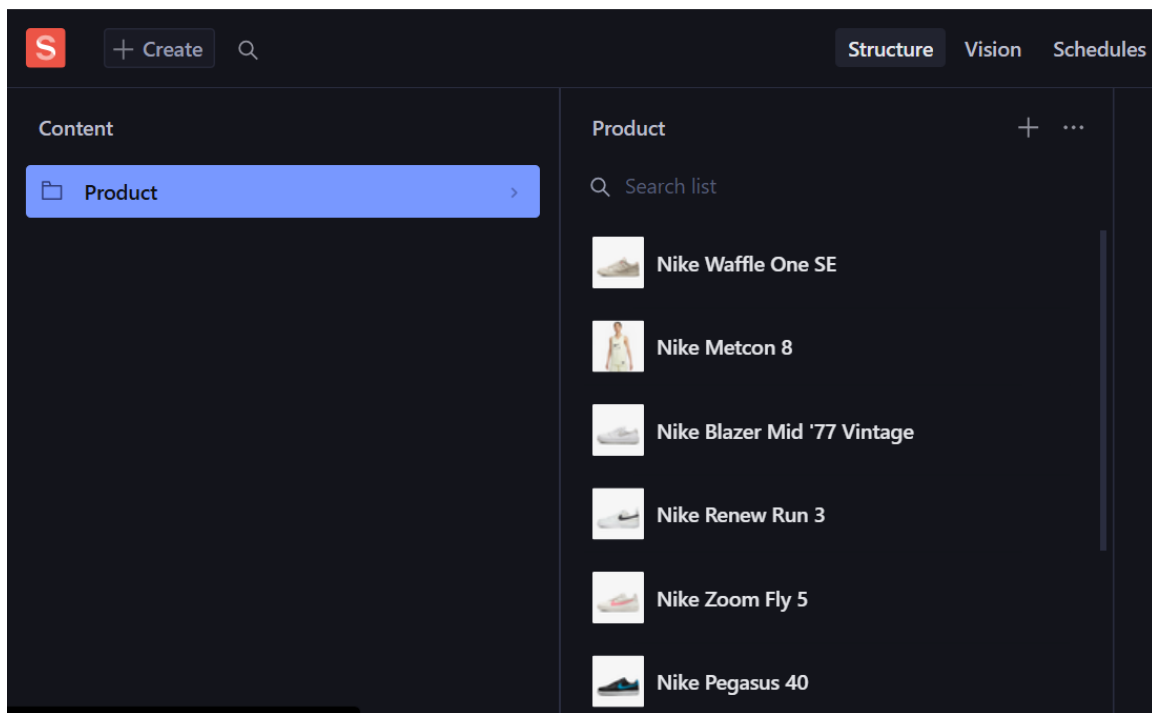
A screenshot of a terminal window titled 'next-server (v14.2.14)'. The terminal shows two GET requests: '/studio/vision' (200 in 11904ms) and '/favicon.ico' (200 in 337ms). Below these, a JSON array of three shoe product objects is displayed. The first object is for 'Air Jordan 1 Elevate Low' with an inventory of 8, price of 11895, and status 'Promo Exclusion'. The second is for 'Nike Court Vision Low' with an inventory of 25, price of 5695, and status 'Just In'. The third object is partially visible, showing a price of 7295 and color 'Blue'.

```
next-server (v14.2.14)
GET /studio/vision 200 in 11904ms
GET /favicon.ico 200 in 337ms
[
  {
    inventory: 8,
    status: 'Promo Exclusion',
    imageUrl: 'https://cdn.sanity.io/images/tcbr17hb/production/46998b7aa7bbf7d305755b2964da5bf167434d40-348x348.png',
    description: 'The Air Jordan 1 Elevate Low features a clean, minimal design with premium materials. Its platform sole adds a touch of height and modern edge, while ensuring comfort and durability for all-day wear.',
    colors: [ 'White' ],
    _id: '1hj3tJP0IaKgRmKaWhsacU',
    productName: 'Air Jordan 1 Elevate Low',
    category: "Women's Shoes",
    price: 11895
  },
  {
    inventory: 25,
    imageUrl: 'https://cdn.sanity.io/images/tcbr17hb/production/1e45fde83014ff9ae9e4cbd280747842f522f47b-348x348.png',
    description: 'The Nike Court Vision Low blends basketball-inspired style with everyday comfort. Featuring a sleek design and durable materials, this versatile shoe is perfect for any casual wardrobe.',
    _id: '1hj3tJP0IaKgRmKaWhsdCA',
    productName: 'Nike Court Vision Low',
    category: "Men's Shoes",
    price: 5695,
    colors: [ 'White' ],
    status: 'Just In'
  },
  {
    price: 7295,
    colors: [ 'Blue' ],
```

2. Frontend Display



3. Populated Sanity CMS Fields



Code Snippets

API Integration

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

Migration Script

```
async function uploadImageToSanity(imageUrl) {
  const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
  const buffer = Buffer.from(response.data);
  const asset = await client.assets.upload('image', buffer, {
    filename: imageUrl.split('/').pop()
  });
  return asset._id;
}

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

  for (const product of products) {
    let imageRef = null;
    if (product.image) {
      imageRef = await uploadImageToSanity(product.image);
    }

    const sanityProduct = {
      _type: 'product',
      productName: product.productName,
      category: product.category,
      price: product.price,
      inventory: product.inventory,
      colors: product.colors || [],
      status: product.status,
      description: product.description,
      image: imageRef ? {
        _type: 'image',
        asset: {
          _type: 'reference',
          _ref: imageRef,
        },
      } : undefined,
    };

    await client.create(sanityProduct);
  }
}

importData();
```