Day 3 - API Integration Report for Nike E-Commerce Website

Overview

On Day 3, the focus was to integrate APIs into the Nike- e-commerce project and populate Sanity CMS with product data sourced from a local API. This report documents the API integration process, schema adjustments, data migration steps, and the tools utilized. Screenshots and code snippets are included to provide a comprehensive understanding.

API Integration Process

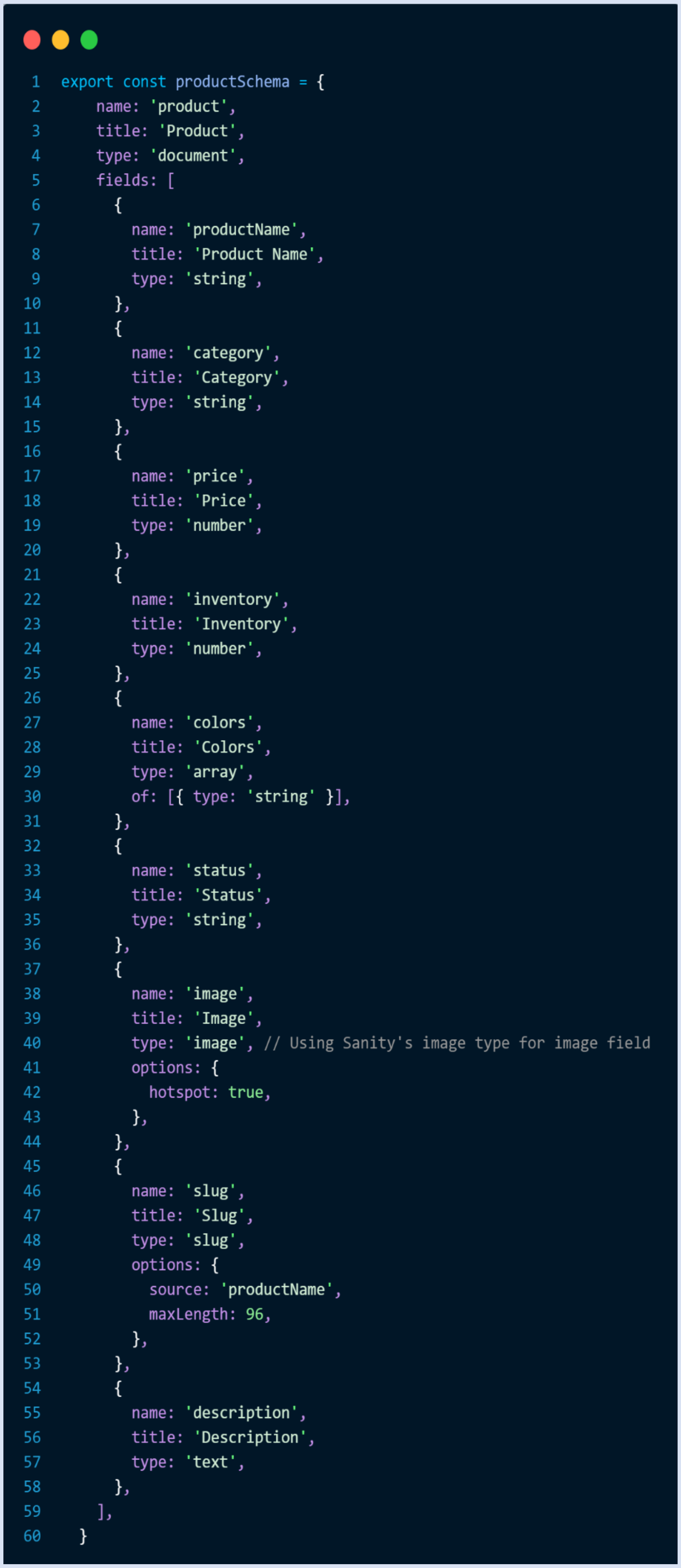
1. **Data Source**
   Data was fetched from the API endpoint: https://template-03-api.vercel.app/api/products
   - **Fields included:** Product Name, description, price, images, inventory, categories, and status.
2. **Integration Steps**
   - **Schema Design**: Created a custom schema for Nike products in Sanity CMS to match the data structure.
   - **Import Script**: Developed a script to fetch product data, process it, and upload it to Sanity CMS.
   - **Image Handling**: Implemented logic to handle multiple product images and upload them as unique assets in Sanity CMS.
   - **Data Validation**: Ensured all fields like slug, price, and stock level adhered to validation rules.
   - **API Endpoints**: Created endpoints to fetch data from Sanity CMS for frontend usage.

Schema Adjustments

Customized product schema to accommodate additional Nike-specific data fields like:

- Colors (e.g. red, blue).
- Slug

Schema Example

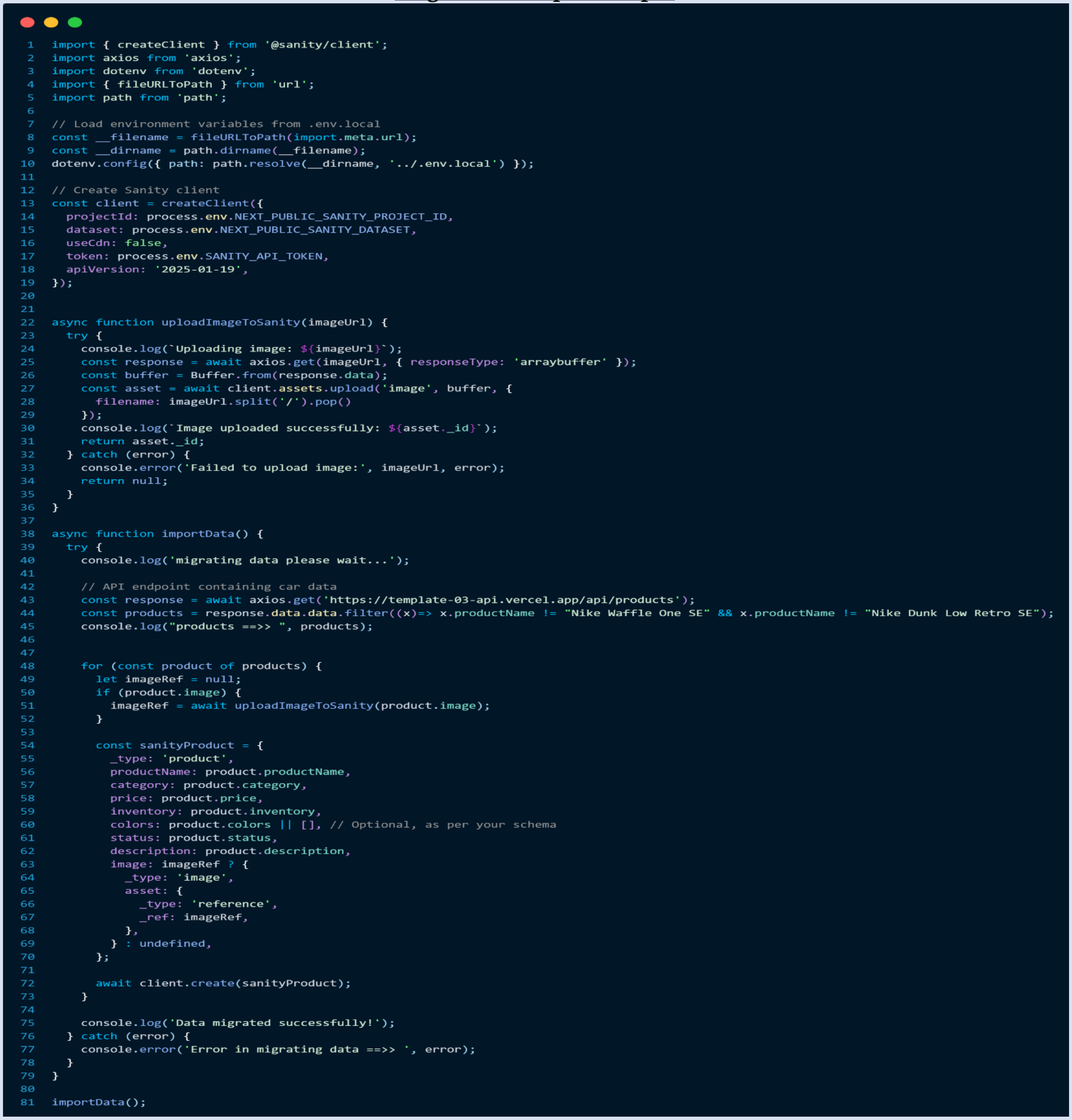


```
export const productSchema = {
    name: 'product',
    title: 'Product',
    type: 'document',
    fields: [
      {
        name: 'productName',
        title: 'Product Name',
        type: 'string',
      },
      {
        name: 'category',
        title: 'Category',
        type: 'string',
      },
      {
        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: 'slug',
        title: 'Slug',
        type: 'slug',
        options: {
          source: 'productName',
          maxLength: 96,
        },
      },
      {
        name: 'description',
        title: 'Description',
        type: 'text',
      },
    ],
  }
```

Migration Steps

1. **Tools Used**
   - **Sanity Client**: For uploading data to Sanity CMS.
   - **Axios**: For making API calls.
   - **.env**: To secure API keys and environment variables.

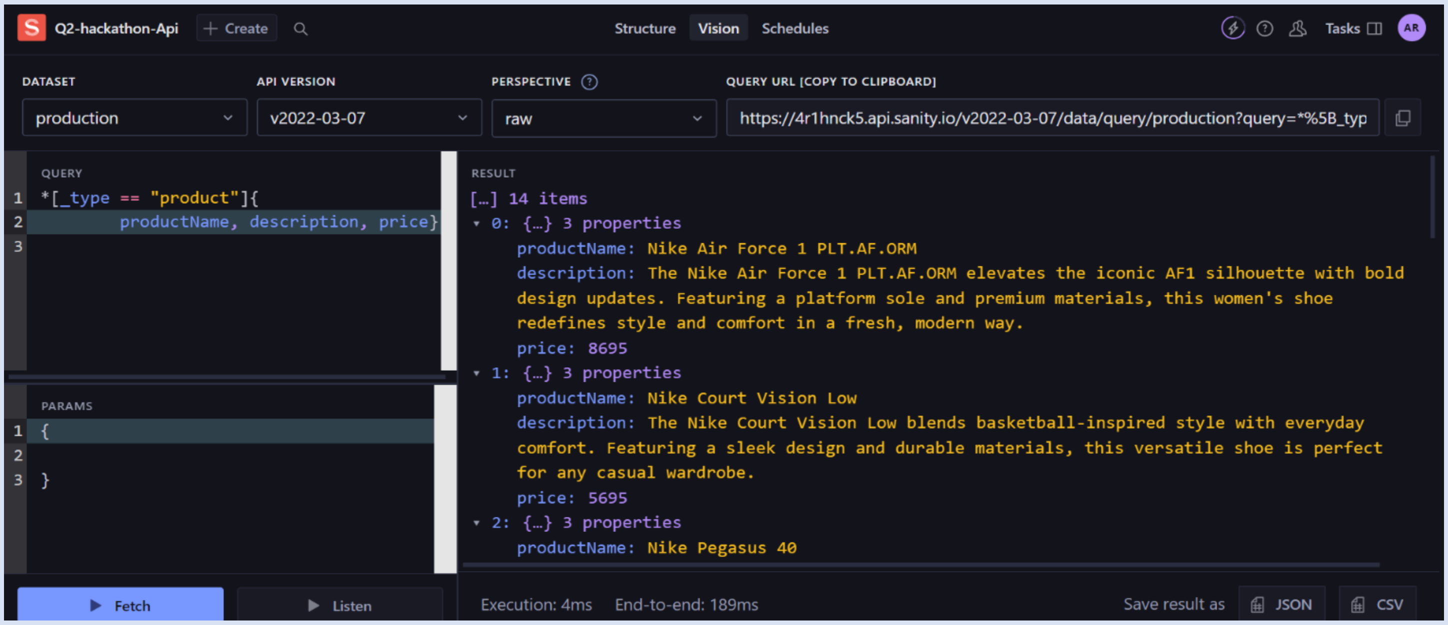
Migration Script Example



```javascript
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

// Load environment variables from .env.local
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

// Create Sanity client
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2025-01-19',
});


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

async function importData() {
  try {
    console.log('migrating data please wait...');

    // API endpoint containing car data
    const response = await axios.get('https://template-03-api.vercel.app/api/products');
    const products = response.data.data.filter((x)=> x.productName != "Nike Waffle One SE" && x.productName != "Nike Dunk Low Retro SE");
    console.log("products ==>> ", products);


    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 || [], // Optional, as per your schema
        status: product.status,
        description: product.description,
        image: imageRef ? {
          _type: 'image',
          asset: {
            _type: 'reference',
            _ref: imageRef,
          },
        } : undefined,
      };

      await client.create(sanityProduct);
    }

    console.log('Data migrated successfully!');
  } catch (error) {
    console.error('Error in migrating data ==>> ', error);
  }
}

importData();
```

API Response Validation

- Screenshot of API responses tested in Postman or browser.



DATASET production
API VERSION v2022-03-07
PERSPECTIVE raw
QUERY URL [COPY TO CLIPBOARD] https://4r1hnck5.api.sanity.io/v2022-03-07/data/query/production?query=*%5B_typ

QUERY
```
*[_type == "product"]{
       productName, description, price}
```

PARAMS
```
{

}
```

RESULT
```
[…] 14 items
▾ 0: {…} 3 properties
    productName: Nike Air Force 1 PLT.AF.ORM
    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.
    price: 8695
▾ 1: {…} 3 properties
    productName: Nike Court Vision Low
    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.
    price: 5695
▾ 2: {…} 3 properties
    productName: Nike Pegasus 40
```

Execution: 4ms End-to-end: 189ms

Conclusion

The integration established a seamless process for importing, validating, and displaying Nike product data from the local API into Sanity CMS. Custom schemas and migration scripts ensured data consistency and project alignment. Future steps include enhancing API queries and implementing advanced features like search and filtering.