Day 3 API INTEGRATION In Sanity Project

Migrating Data to Sanity using Next.js

I recently worked on integrating my API with Sanity CMS for an eCommerce project. The goal was to fetch data from my Mocki API, transform it as needed, and push it into Sanity CMS. This process allowed seamless data migration while maintaining structured content in Sanity.

Here's a step-by-step breakdown of how I achieved this.

1. Fetching Data from the API

The first step was retrieving data from my Mocki API. Using Next.js, I made a GET request to fetch the product details and checked the response in the console:

```
async function importData() {
try {
console.log('Fetching products from API...');
const response = await axios.get('https://mockapi.io/projects/6
const products = response.data;
console.log(Fetched ${products.length} products);
} catch (error) {
console.error('Error fetching data:', error);
}
}
```

2. Aligning API Data with Sanity Schema

Once the data was fetched, the next task was to compare the structure of the API data with the Sanity CMS schema. The Sanity schema defines the structure of the content that will be stored in the CMS. In this case, a product schema was defined to handle the product data, which included fields such as

- My Api & My Sanity Schema :

```
{
    "name": "Trenton modular sofa_3",
    "price": "25,000.00",
    "reviews": 45,
    "description": "The Trenton modular sofa_3 is a modern, customizable seating solution for any space, combining comfort and style with its plush cushions and sleek design.",
    "size": [
      "L",
      "XL",
      "XS"
    ],
    "color": [
      "purple",
      "black",
      "brown"
    ],
    "id": "1",
    "image": "https://i.imgur.com/DNTglZE.png",
    "stockInHand": 350,
    "stockSold": 180,
    "discountPercentage": 10,
    "isFeaturedProduct": true
  },
```

```
const products = {
name: 'product',
title: 'product',
type: 'document',
fields: [
{ name: 'name', title: 'Name', type: 'string' },
{ name: 'price', title: 'Price', type: 'number' },
{ name: 'category', title: 'Category', type: 'string' },
{ name: 'description', title: 'Description', type: 'text' },
{
name: 'image',
title: 'Image',
type: 'image',
options: { hotspot: true },
},
{ name: 'rating', title: 'Rating', type: 'number' },
{ name: 'reviews', title: 'Reviews', type: 'number' },
{ name: 'size', title: 'Size', type: 'array', of: [{ type: 'str
{ name: 'color', title: 'Color', type: 'array', of: [{ type: 's
{ name: 'id', title: 'ID', type: 'string' },
{ name: 'stockInHand', title: 'Stock In Hand', type: 'number' },
{ name: 'stockSold', title: 'Stock Sold', type: 'number' },
{ name: 'discountPercentage', title: 'Discount Percentage', type
{ name: 'isFeaturedProduct', title: 'Is Featured Product', type
],
};

export default products;
```

I compared the data structure between the API and Sanity schema to ensure alignment. This verification step confirmed that each API data field had a matching field in the Sanity schema structure.

3. Writing the Data Migration Script

After confirming the data structure, I wrote a script to:

1. **Fetch product data from the API.**
2. **Format the data to match the Sanity schema.**
3. **Push the formatted data to Sanity using its API.**

```
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: '2021-08-31',
})

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`)
    const response = await axios.get(imageUrl, { responseType:
    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('Fetching products from API...')
    const response = await axios.get('https://677fda9a0476123f7
    const products = response.data
    console.log(`Fetched ${products.length} products`)

    for (const product of products) {
      console.log(`Processing product: ${product.name}`)

      // Convert price to number (remove commas)
      const price = parseFloat(product.price.replace(/,/g, ''))

      // Upload image and get reference
      let imageRef = null
      if (product.image) {
        imageRef = await uploadImageToSanity(product.image)
      }

      // Create Sanity product document
      const sanityProduct = {
        _type: 'product',
        name: product.name,
        price: price,
        category: product.category || 'General',
        description: product.description,
        reviews: product.reviews,
        size: product.size,
        color: product.color,
        id: product.id,
        stockInHand: product.stockInHand,
        stockSold: product.stockSold,
        isFeaturedProduct: product.isFeaturedProduct,
        discountPercentage: product.discountPercentage,
        rating: product.reviews, // Assuming reviews act as rat
        image: imageRef
          ? {
              _type: 'image',
              asset: { _type: 'reference', _ref: imageRef },
            }
          : undefined,
      }

      console.log('Uploading product to Sanity:', sanityProduct
      const result = await client.create(sanityProduct)
      console.log(`Product uploaded successfully: ${result._id}
    }

    console.log('Data import completed successfully!')
  } catch (error) {
    console.error('Error importing data:', error)
  }
}

importData()
```

4 . Modifying Package.json

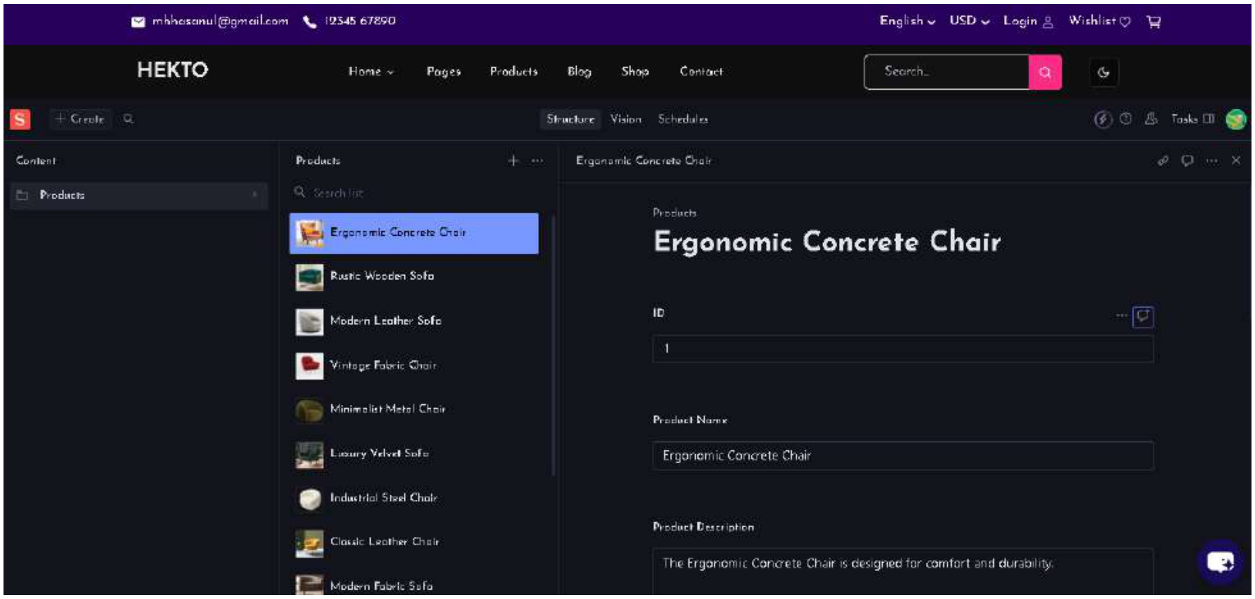
```
"import-data": "node scripts/importSanityData.mjs"
```

5 . Running the Import Command to Import data in SANITY

With everything properly set up, I successfully executed the migration script using the following command:

> 💡 npm run import-data

As a result data from API was inserted in my Sanity Studio as you can see in the image below :



Ergonomic Concrete Chair
Rustic Wooden Sofa
Modern Leather Sofa
Vintage Fabric Chair
Minimalist Metal Chair
Luxury Velvet Sofa
Industrial Steel Chair
Classic Leather Chair
Modern Fabric Sofa

Ergonomic Concrete Chair

Products

Ergonomic Concrete Chair

ID: 1

Product Name: Ergonomic Concrete Chair

Product Description: The Ergonomic Concrete Chair is designed for comfort and durability.

Conclusion

We successfully automated the migration of data from an external API to Sanity CMS. The process involved:

1. Retrieving product data from the API.
2. Transforming and structuring the data to fit the Sanity schema.
3. Writing a migration script for smooth data import.
4. Setting up environment variables and executing the script.
5. Running the import command to complete the process.

This automation ensured a smooth and error-free transition, making the data transfer quick, reliable, and hassle-free. :)