

Day 3 API Integration And Data Migration

Marketplace: [E-Commerce]

Overview:

API integration and data migration are critical processes in modern software development, enabling seamless communication between different systems and ensuring the efficient transfer of data. This document provides an in-depth overview of these concepts, their importance, and best practices to follow during implementation.

1. API Integration

API (Application Programming Interface) integration involves connecting two or more applications or systems to enable data sharing and functionality exchange. APIs act as intermediaries, allowing different software applications to interact with each other effectively.

2. Sanity Schema

The Sanity product schema was adjusted to ensure compatibility with the product data from the external API. The following fields were added:

```

export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'id',
      title: 'ID',
      type: 'string',
    },
    {
      name: 'name',
      title: 'Name',
      type: 'string',
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image',
    },
    {
      name: 'imagePath',
      title: 'Image Path',
      type: 'url',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
    {
      name: 'discountPercentage',
      title: 'Discount Percentage',
      type: 'number',
    },
  ],
}

```

3. Migration Steps and Tools Used Migration Process:

4 Data Fetching:-

Used Axios to fetch the product data from the external API.

Image Uploading:-

For products with images, images were uploaded to Sanity using the asset upload feature, where images were fetched and uploaded as buffer data.

Sanity Client:-

Used Sanity client to create new documents within the CMS using the data received from the API.

Migration Script:-

The script (importSanityData.mjs) was written to automate the migration of product data and images.

Tools Used:

Axios: For making HTTP requests to the external API.

Sanity Client: For pushing the data to Sanity CMS.

dotenv: For managing environment variables securely.

Migration Script

```

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: '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.message);
    return null;
  }
}

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

    // Fetch products from the API
    const response = await axios.get('https://template-0-beta.vercel.app/api/product');
    const products = response.data;

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

    for (const product of products) {
      let imageRef = null;

      if (product.imagePath) {
        imageRef = await uploadImageToSanity(product.imagePath);
      }

      const sanityProduct = {
        _type: 'product',
        id: product.id,
        name: product.name,
        category: product.category,
        description: product.description,
        discountPercentage: product.discountPercentage,
        isFeaturedProduct: product.isFeaturedProduct,
        stockLevel: product.stockLevel,
        price: parseFloat(product.price),
        image: imageRef
        ? {
            _type: 'image',
            asset: {
              _type: 'reference',
              _ref: imageRef,
            },
          }
        : undefined,
        imagePath: product.imagePath,
      };
      // Store original image URL
    }
  }
}

```

Data Successfully Displayed in Frontend:



Pink Lounge Chair
Rs. 1600 -20%

Limited Stock



Cozy Sofa
Rs. 520 -8%

Limited Stock



Liberty Center
Rs. 1100 -8%

Limited Stock



Matilda Velvet Bed
Rs. 600 -15%

Limited Stock



Home > Shop | Matilda Velvet Bed



Matilda Velvet Bed

Rs. 600 -15%

★★★★★ | 5 Customer Reviews

A vibrant pink velvet chair with a retro-inspired design.

- 1 +

Add To Cart



Add to Wishlist

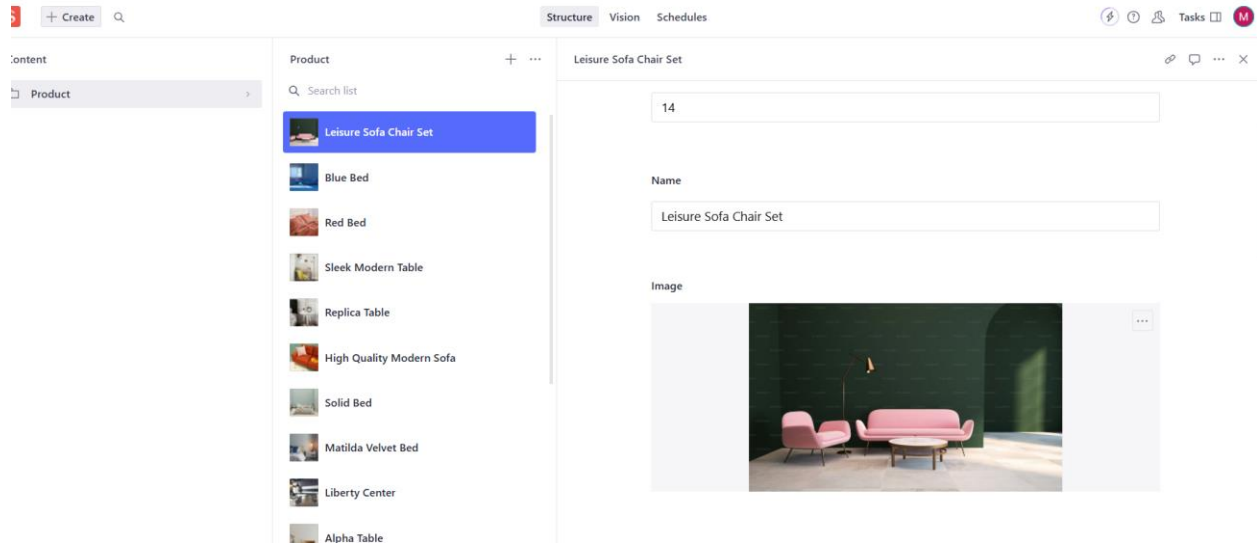
SKU : ADF566

Category : Bed

Tags : Sofa, Chair, Home, Shop

Share : [f](#) [in](#) [t](#)

Sanity Studio:



Conclusion:

This report outlines the API integration and data migration process that was carried out for my marketplace application. The Sanity schema was adjusted to accommodate the incoming product data, and a Node.js migration script was developed to automate the migration process. The product data is now successfully displayed on the frontend, and the CMS fields are populated with the data.

API Integration Report Checklist:

1. API Understanding: ✓
2. Schema Validation: ✓
3. Data Migration: ✓
4. API Integration in Next.js: ✓
5. Submission Preparation: ✓