

DAY 03 – API INTEGRATION REPORT OF COMFORTY

Introduction

In this project, we were tasked with importing data from two APIs into Sanity using migration scripts. We successfully completed this task and then fetched the data to display in our user interface.

Steps Involved

1. API Integration
 - We received two APIs from which we needed to import data into Sanity. The data
 - from these APIs was crucial for our project.
2. Schema Creation
 - We created a schema for the product and categories. This schema defines how the data will be structured in Sanity.
 - The schema is essential for organizing and querying the data effectively.
3. Migration Scripts
 - We developed migration scripts to import data from the APIs into Sanity.
 - These scripts ensure that the data is correctly formatted and inserted into the Sanity database.
4. Adding Migration to package.json
 - We added the migration script to the package.json file. This allows us to run the migration using npm commands.
 - Specifically, we used `npm run migrate` to execute the migration script.
5. Importing Data into Sanity
 - After running the migration script, the data from the APIs was successfully imported into Sanity.
 - This step ensured that our Sanity database was populated with the necessary data.
6. Fetching Data with GROQ
 - We used GROQ (Sanity's query language) to fetch the imported data from Sanity. GROQ
 - queries allow us to specify exactly what data we need, making it efficient for displaying in our UI.
7. Displaying Data in UI
 - Finally, we displayed the fetched data in our user interface.
 - This completes the full cycle of importing data into Sanity and then using it in our application.
 -

Conclusion

This project involved integrating data from external APIs into Sanity using migration scripts and then querying that data for display in our UI. The process was successful, and we were able to efficiently manage and display the data as needed. This report outlines the steps you took in a clear and simple manner, making it easy to understand the process involved. Share

Rewrite

HomeShopProductPagesAbout

Contact: (808-555-011)

S

Default

+ Create

Q

StructureVisionSchedules

Content

Products

Categories

Products

Search list

Ivory Charm

Gray Elegance

Nordic Spin

Scandi Dip Set

Modern Cozy

Library Stool Chair

SleekSpin

Rose Luxe Armchair

Windows Taskbar

USD/GBP +0.59%

4:37 PM 1/18/2025

HomeShopProductPagesAbout

Contact: (808-555-011)

S

Default

+ Create

Q

StructureVisionSchedules

Content

Products

Categories

Categories

Search list

Desk Chair

Wooden Chair

Wing Chair

Desk Chair

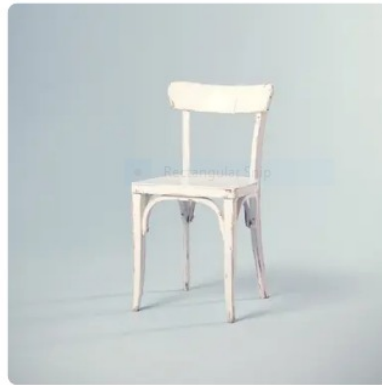
Categories

Desk Chair

Category Image



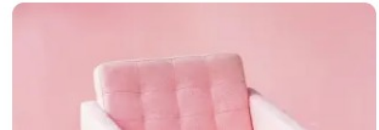
Citrus Edge
\$20



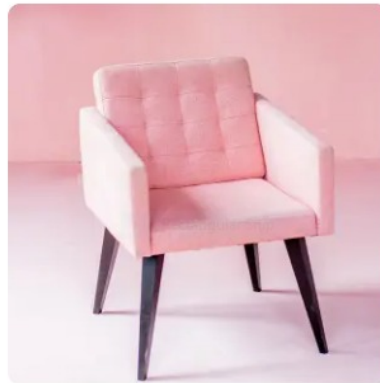
Library Stool Chair
\$20



Modern Cozy
\$20



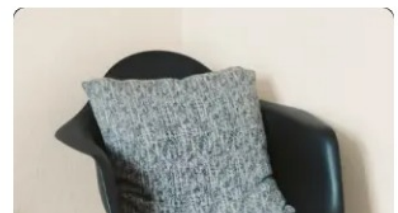
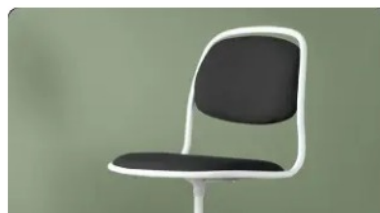
SleekSpin
\$20



Rose Luxe Armchair
\$20



Modern Cozy
\$20



EXPLORER

COMFORTY-WEBSITE

next

node_modules

public

scripts

dataMigrate.mjs

src

env

env.local

gitignore

eslint.config.mjs

next-env.d.ts

next.config.ts

package-lock.json

package.json

postcss.config.mjs

README.md

sanity.cli.ts

sanity.config.ts

tailwind.config.ts

tsconfig.json

OUTLINE

TIMELINE

scripts > JS dataMigrate.mjs > ...

1 // Import environment variables from .env.local

2 import "dotenv/config";

3

4 // Import the Sanity client to interact with the Sanity backend

5 import { createClient } from "@sanity/client";

6

7 // Load required environment variables

8 const {

9 NEXT_PUBLIC_SANITY_PROJECT_ID="8s8cig99", // Sanity project ID

10 NEXT_PUBLIC_SANITY_DATASET="production", // Sanity dataset (e.g., "production")

11 NEXT_PUBLIC_SANITY_AUTH_TOKEN="skqFGJHxR1Vjw9BULXC1skUL5VD4Tyd0aRN90K0n1emLmYjQtnNVPN08Hj2zyTFsoUjddWAemabkICt5d0j

12 BASE_URL = "https://giaic-hackathon-template-08.vercel.app", // API base URL for products and categories

13 } = process.env;

14

15 // Check if the required environment variables are provided

16 if (!NEXT_PUBLIC_SANITY_PROJECT_ID || !NEXT_PUBLIC_SANITY_AUTH_TOKEN) {

17 console.error("Missing required environment variables. Please check your .env.local file.");

18 process.exit(1); // Stop execution if variables are missing

19 }

20

21 // Create a Sanity client instance to interact with the target Sanity dataset

22 const targetClient = createClient({

23 projectId: NEXT_PUBLIC_SANITY_PROJECT_ID, // Your Sanity project ID

24 dataset: NEXT_PUBLIC_SANITY_DATASET || "production", // Default to "production" if not set

25 useCdn: false, // Disable CDN for real-time updates

26 apiVersion: "2023-01-01", // Sanity API version

27 token: NEXT_PUBLIC_SANITY_AUTH_TOKEN, // API token for authentication

28 });

29

30 // Function to upload an image to Sanity

31 async function uploadImageToSanity(imageUrl) {

32 try {

Ln 18, Col 62 Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier

EXPLORER

COMFORTY-WEBSITE

next

node_modules

public

scripts

src

app

sanity

lib

schemaTypes

categories.ts

index.ts

products.ts

env.ts

structure.ts

env

env.local

gitignore

eslint.config.mjs

next-env.d.ts

next.config.ts

package-lock.json

package.json

postcss.config.mjs

README.md

sanity.cli.ts

sanity.config.ts

OUTLINE

TIMELINE

src > sanity > schemaTypes > TS products.ts > products > fields

1 import { defineType } from "sanity"; // Importing the defineType function from Sanity to create a custom schema.

2

3 export const products = defineType({

4 name: "products", // Unique name for the schema, used internally in Sanity.

5 title: "Products", // Human-readable title shown in Sanity Studio.

6 type: "document", // This defines it as a document type, meaning it will store data for each product.

7 fields: [// Array of fields that define the structure of each product document.

8 {

9 name: "title", // Field name used in code.

10 title: "Product Title", // Label shown in Sanity Studio.

11 type: "string", // Data type for this field (text).

12 },

13 {

14 name: "price", // Field for the product's price.

15 title: "Price", // Label shown in Sanity Studio.

16 type: "number", // Stores numeric values.

17 },

18 {

19 title: "Price without Discount", // Original price before discount.

20 name: "priceWithoutDiscount",

21 type: "number",

22 },

23 {

24 name: "badge", // Field to display a badge like "New" or "Sale".

25 title: "Badge", // Label shown in Sanity Studio.

26 type: "string",

27 },

28 {

29 name: "image", // Field for uploading product images.

30 title: "Product Image",

31 type: "image",

32 },

33 },

Ln 25, Col 55 Spaces: 2 UTF-8 CRLF TypeScript Go Live Prettier

