Marketplace-Builder-Hackathon

Day-3: API Integration and Data Migration in Sanity

EXPLANATION OF MIGRATING DATA FROM API TO SANITY IN NEXTJS

The focus for Day 3 of the hackathon was on API Integration and Data Migration. The task was to integrate external APIs into the marketplace project. implemented a solution that allows us to migrate and import data from an external API into sanity CMS. By the end of the day, the objective was to have a fully integrated marketplace with dynamic data fetched from APIs and correctly displayed on the frontend. Below is step by step explanation how I achieved it:

Created by Emad Ahmed Roll Num => 227065

1. Fetching Data from External APIs

The first step in this process was to fetch data from the external API. In my Next.js application, where the frontend code was already present, I created a folder named "scripts" and within that, I created a file called "data-migration.mjs" where I wrote the code to fetch data from the API.

Afterward, I tested it both in the application and through Postman API, and my data was successfully fetched.

```
import sanityClient from '@sanity/client';

const client = sanityClient({
  projectId: '8p9t123f',
  dataset: 'production',
  useCdn: true,
  apiVersion: '2025-01-13',
  token:
  'skZmy130OQiIxde8zb8sqH40I2D0usUaQOaqsthKGotYwgHWP568Z1ujRLDHIfle44oevfr4R6jLR618c
  50fgGltRXSwJdiydpuqN0LLb9SlKTsH3qsfXjbLrRy4jiISmvLpoTfD1jNMhUWrUTSSKuBbpwuH4DWZqEf
  1Labmue5OY4T1KYUV',
});

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);

    const response = await fetch(imageUrl);
    if (!response.ok) {
      throw new Error(`Failed to fetch image: ${imageUrl}`);
    }
```

2. Comparing API data with Sanity Schema

After the data was fetched, the next task was to compare the data structure of the API with the Sanity schema. The Sanity schema will be used to handle the product data. The main purpose of this step was to ensure that the data coming from the API matches the Sanity schema. This step ensures that the data from the API perfectly matches the Sanity schema.

```
import { defineType } from "sanity"

export const product = defineType({
    name: "product",
    title: "Product",
    type: "document",
    fields: [
        {
            name: "title",
            title: "Title",
            validation: (rule) => rule.required(),
            type: "string"
        },
        {
            name:"description",
            type:"text",
            validation: (rule) => rule.required(),
            title:"Description",
        },
```

3. Data Migration Script

After confirming the schema and the API data structure, the next step was to write a migration script in existing file data-migration.mjs that wood automate the process of importing the fetched data into Sanity. A script was created that would:

1. Fetch data from API

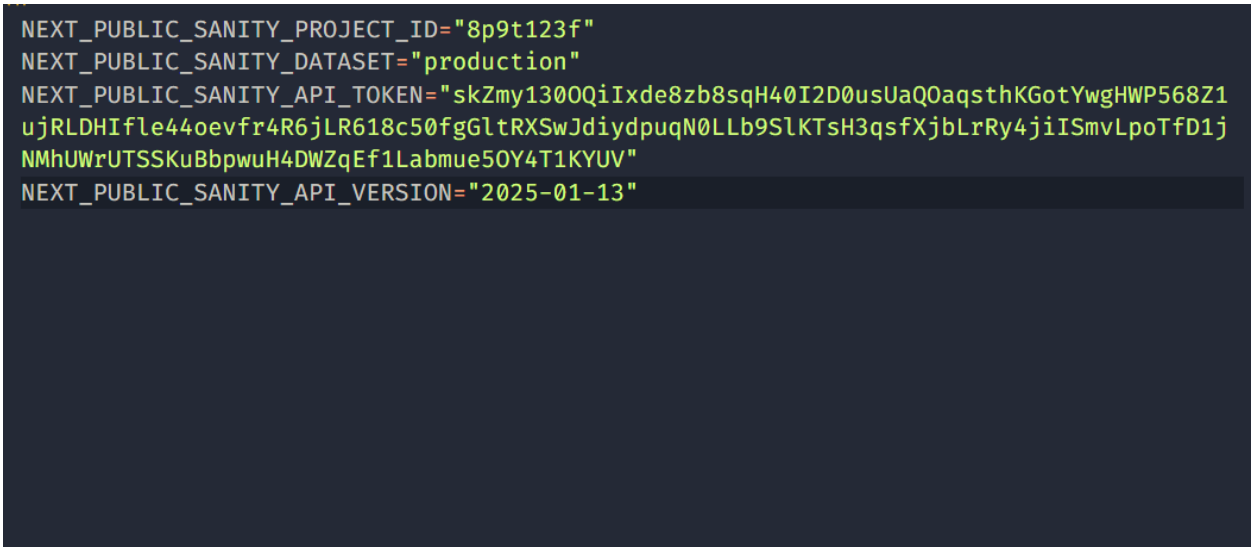
2. Validate data with sanity schema

3. Use the sanity client to push data into sanity

```
async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);

    const response = await fetch(imageUrl);
    if (!response.ok) {
      throw new Error(`Failed to fetch image: ${imageUrl}`);
    }

    const buffer = await response.arrayBuffer();
    const bufferImage = Buffer.from(buffer);

    const asset = await client.assets.upload('image', bufferImage, {
      filename: imageUrl.split('/').pop(),
    });
```

4. Setting Up The Environment

Once the script was written, the next step was to configure Sanity. For the configuration, a token for the project needs to be generated from Sanity's official website, and by looking at the ID, DATA, and TOKEN, it needs to be written in the .env.local file, and also provided in the data-migration.mjs file.



5. Modifying package.json

To run the migration script, the necessary npm package was edit to the package json file under the scripts section. This made it easier to run the script through the command line.

6. Importing Data into Sanity

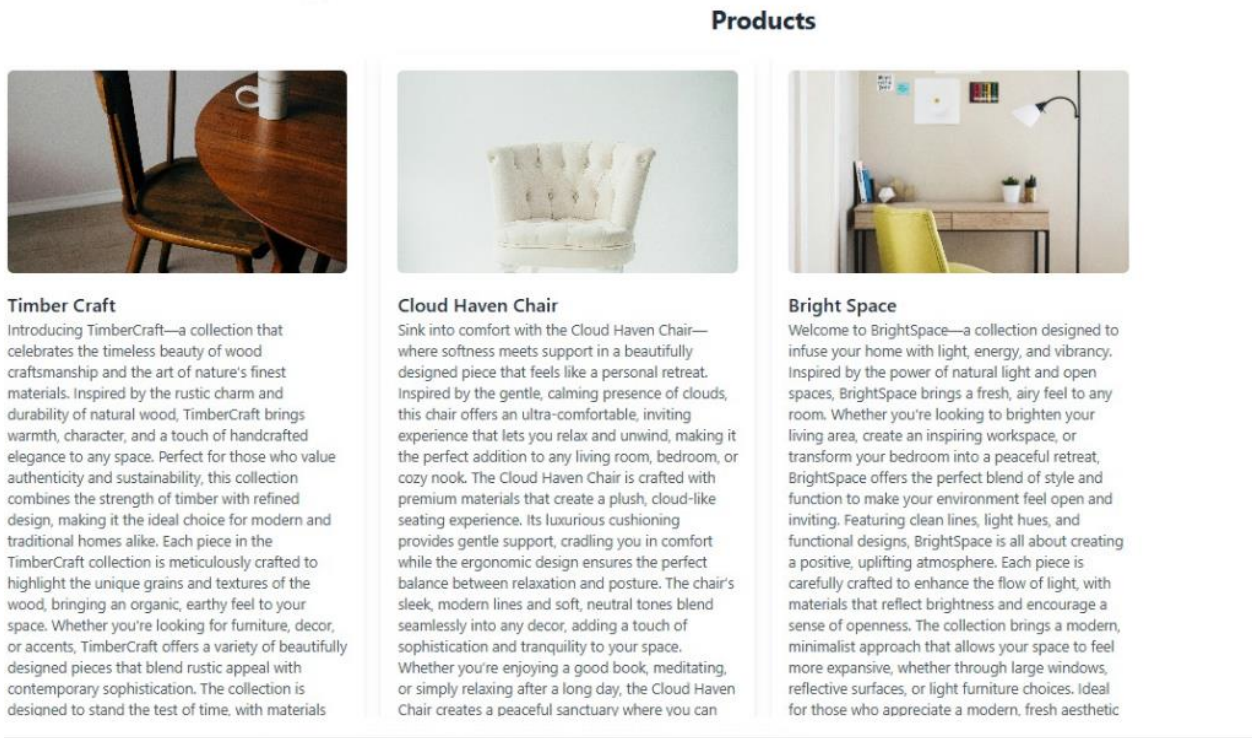
With everything setup, I executed the migration script to send data into sanity by running the following command:

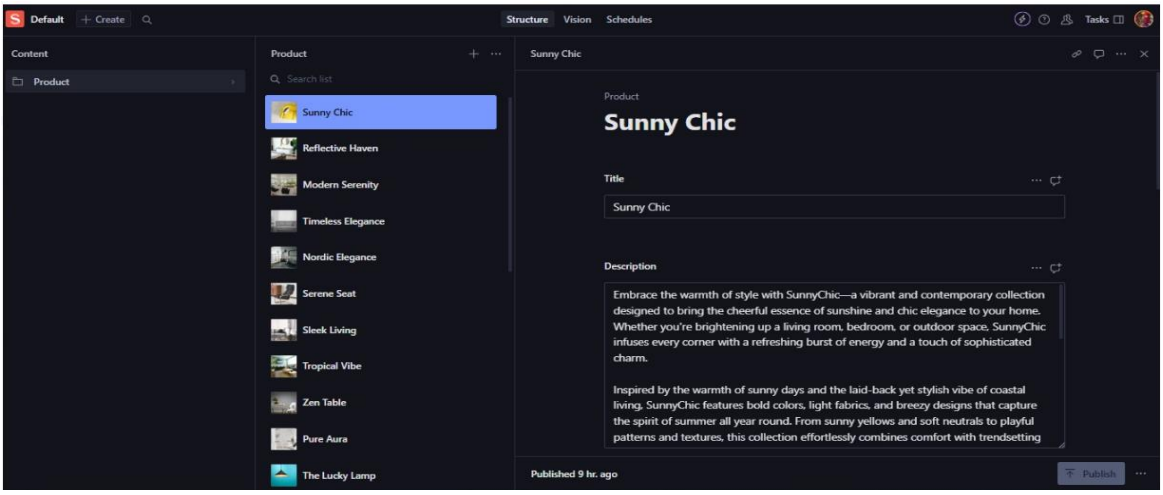
npm run migrate

```
import { client } from "@/sanity/lib/client";
import { NextResponse } from "next/server";


export async function GET() {
  try {
    const data = await client.fetch(`*[_type=="product"]{
    _id,
    title,
    "imageUrl" :productImage.asset -> url,
    price,
    tags,
    dicountPercentage,
    description,
    isNew
}`);

    return  NextResponse.json(data, { status: 200 });
  } catch (error) {
    console.error('Error fetching data from Sanity:', error);
    return new NextResponse('Error fetching data', { status: 500 });
  }
}
```

7. Displaying data in frontend

After fetching data from Sanity, I mapped it and printed it in cards, and successfully my data was displayed on the frontend.





Conclusion

This process successfully automated the migration of data from an external API to Sanity CMS. The Key steps included:

- ✓ Fetching Data from External APIs
- ✓ Comparing API data with Sanity Schema
- ✓ Data Migration Script
- ✓ Setting Up The Environment
- ✓ Modifying package.json
- ✓ Importing Data into Sanity
- ✓ Displaying data in frontend