

Day 3 - API Integration Report - Havenly Habitats

Prepared by: Marium umair

Introduction

This report details the API integration process, adjustments made to the schemas, and data migration steps for Havenly Habitats. Additionally, it includes screenshots showcasing successful API calls, data displayed on the frontend, and populated Sanity CMS fields.

API Integration Process

1. Setup and Configuration

- Integrated third-party product API: <https://hackathon-apis.vercel.app/api/products>
- Used **Axios** for API requests in Next.js.
- Implemented **Sanity CMS** as the backend.
- **Environment variables** were utilized to protect API keys.

2. Fetching Data from API

- Implemented an **Axios GET request** to fetch product data.
- Used `useEffect` in Next.js to handle the asynchronous API call.
- Implemented error handling to manage failed requests.

3. Data Processing & Transformation

- Extracted necessary fields such as name, category, price, image, and features.
- Used `slugify` to generate URL-friendly product slugs.
- Implemented a function to handle missing or incomplete data.

4. Pushing Data to Sanity CMS

- Utilized Sanity's **createOrReplace** API to insert/update product data.
- Used `client.assets.upload()` to store images in Sanity.
- Ensured references between **products and categories** were correctly linked.

Schema Adjustments

1. Product Schema Updates

- Added dimensions object to store height, width, and depth.
- Updated category field to reference the category schema.
- Introduced tags for improved filtering options.

2. Category Schema Adjustments

- Ensured unique slugs for each category.
- Added descriptions to categories for better classification.

Data Migration Steps & Tools Used

1. Migration Plan

- Fetched existing data from the API.
- Mapped the API response to match Sanity's schema.
- Batched data uploads to avoid API rate limits.

2. Tools Used

- **Axios:** For data retrieval from external APIs.
- **Sanity Client:** For managing and storing data in the headless CMS.
- **Postman:** For testing API responses before implementation.
- **Vercel Logs:** To monitor and debug API calls in real time.

Screenshots

1. API Calls

(Include screenshots showing API request & response)

2. Data Successfully Displayed in Frontend

(Include screenshots of the Next.js UI showing product data)

3. Populated Sanity CMS Fields

(Include screenshots from the Sanity dashboard with imported data)

Conclusion

The API integration and data migration process for Havenly Habitats was successfully executed. Data from external sources was transformed and uploaded to Sanity CMS, ensuring a structured and efficient backend. The frontend now dynamically displays real-time product data, improving overall marketplace functionality.