

Day 3 - API Integration Report – Furniture Store

API Integration Process

- **Overview:**

The integration connects an external API providing **furniture products** and **categories** data to a Sanity CMS project.

- **Steps Taken:**

1. Environment Setup

- **Dotenv Configuration:**
 - Used the dotenv package to load environment variables from the `.env.local` file.
 - The following key environment variables were configured:
 - `NEXT_PUBLIC_SANITY_PROJECT_ID`: The unique Sanity project ID.
 - `NEXT_PUBLIC_SANITY_DATASET`: The dataset used for the Sanity project (e.g., `production`).
 - `SANITY_API_TOKEN`: The authentication token for secure access to the Sanity CMS.

By Areesha khan

2. Sanity Client Setup

- Imported the Sanity client from a custom file located at `@/sanity/lib/client`.
 - The client was pre-configured with the following parameters:
 - **Project ID:** Retrieved from the environment variables.
 - **Dataset:** Defined as per the environment variables (e.g., `production`).
 - **API Version:** Set to `2021-08-31` for compatibility.
 - **Authentication Token:** Used the `SANITY_API_TOKEN` environment variable for secure communication with Sanity CMS.
-

3. Data Fetching Process

- Used **Axios** to make concurrent API requests to two external APIs:
 - **Furniture Products API:**
 - Endpoint:
<https://hackathon-apis.vercel.app/api/products>
 - Fetches detailed product information such as name, description, price, image URL, etc.
 - **Categories API:**
 - Endpoint:
<https://furniture-api.vercel.app/api/categories>
 - Fetches product categories for organizing the furniture items.
-

By Areesha khan

4. Data Processing

- **Data Transformation:**
 - After fetching the data from the APIs, iterated through the response and transformed it into the required format compatible with the Sanity CMS.
 - Product images were uploaded to Sanity's **Asset Library** to store and manage the image files.
-

5. Sanity Document Creation

- **Document Upload:**
 - Used the `client.create()` method to create new documents for each product in the Sanity CMS.
 - For every fetched product, a corresponding document was created with all the relevant fields such as:
 - Product name
 - Description
 - Price
 - Image (uploaded to Sanity)
 - Tags, dimensions, quantity, features, etc.
-

6. Error Handling

- **API and Sanity Error Handling:**
 - Implemented `try-catch` blocks to handle errors during API calls and Sanity operations.
-

By Areesha khan

- Errors were logged to the console to help with debugging and to provide clear, user-friendly error messages for easier monitoring and troubleshooting.

Adjustments Made to Schemas:

Category Schema

- **Name:** A required string field for the category name.
- **Slug:** A required slug field that automatically generates a slug based on the category name.

Product Schema

- **Category:** A reference field linking to the Category schema, allowing you to categorize products.
- **Name:** A required string field for the product's title.
- **Slug:** A required slug field for generating URLs for the product.
- **Image:** A required image field for the product image.
- **Price:** A required number field for the product's price.
- **Quantity:** A number field with a minimum value of 0 for tracking product quantity.
- **Tags:** An array of strings for product categorization through tags.
- **Description:** A text field for a detailed description of the product.
- **Features:** An array of strings for listing key features of the product.
- **Dimensions:** An object with fields for height, width, and depth, detailing the product's dimensions.

By Areesha khan

Migration Steps and Tools Used

1. Migration Steps:

- **Preparation:**

Analyzed API data structure and ensured alignment with the existing Sanity schemas.

Created Sanity schemas for furniture products, categories, and related assets, including essential fields like name, description, price, images, and dimensions.

- **Data Import Script:**

Developed an `import-data.mjs` script to automate the data fetching, processing, and uploading from the API.

Utilized **axios** for API interaction and **@sanity/client** for seamless integration with the Sanity CMS.

- **Document Creation:**

Mapped API data fields to Sanity schema fields

Implemented fallback values for optional fields or missing data, ensuring all necessary product details were captured.

By Areesha khan

2:Tools Used:

Node.js Modules:

- **axios**: For making HTTP requests.
- **dotenv**: For managing environment variables, enabling secure access to sensitive information.
- **@sanity/client**: For interacting with the Sanity CMS.

Sanity Features:

- Asset management for image uploads.
- API versioning to maintain schema compatibility and ensure consistent content operations.

Utilities:

- **fileURLToPath** and **path**: For resolving and handling file paths in a dynamic and cross-platform manner.
- **slugify**: For converting product and category names into clean, URL-friendly slugs.

By Areesha khan

Screenshot of:

API calls

```
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/9b6a4fc8c65bbb4e5793fb
Image uploaded successfully: image-9b6a4fc8c65bbb4e5793fb0e1116b510d73dc9e8-630x375-png
Category created successfully {
  _createdAt: '2025-01-23T17:12:51Z',
  _id: 'tableware-1',
  _rev: 'jZ1YpFE1TDqhTIemwnmc9v',
  _type: 'category',
  _updatedAt: '2025-01-23T17:12:51Z',
  name: 'Tableware',
  slug: { _type: 'slug', current: 'tableware' }
}
✅ Imported product: The Poplar suede sofa
Processing product: The Steel Chair
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/b3bbbd269d31ce9b57b7a8
Image uploaded successfully: image-b3bbbd269d31ce9b57b7a867bf0b7f6fdc5b3208-721x759-png
Category created successfully {
  _createdAt: '2025-01-23T17:12:55Z',
  _id: 'chairs-2',
  _rev: 'NGbH4pjCdaYIxrHyZFGQhq',
  _type: 'category',
  _updatedAt: '2025-01-23T17:12:55Z',
  name: 'Chairs',
  slug: { _type: 'slug', current: 'chairs' }
}
✅ Imported product: The Steel Chair
Processing product: Rustic Vase Set
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/2219cafc285ec13a2ed3f8
Image uploaded successfully: image-2219cafc285ec13a2ed3f88aa36cbea852a11735-305x375-png
Category created successfully {
  _createdAt: '2025-01-23T17:12:58Z',
  _id: 'plant-pots-3',
  _rev: 'jZ1YpFE1TDqhTIemwnmfvd',
  _type: 'category',
  _updatedAt: '2025-01-23T17:12:58Z',
  name: 'Plant Pots',
  slug: { _type: 'slug', current: 'plant-pots' }
}
```

```

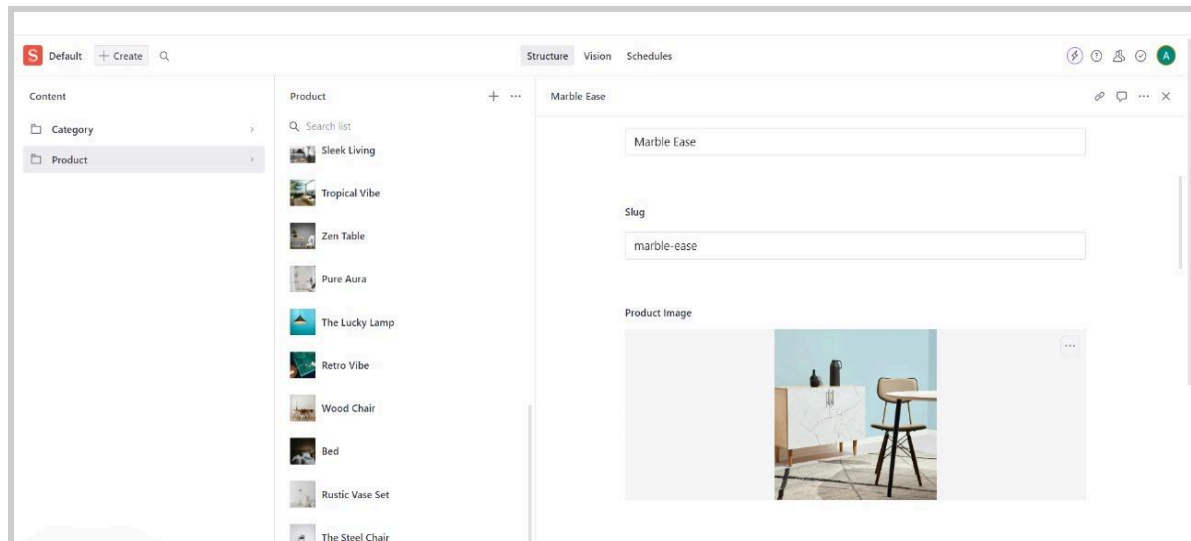
✅ Imported product: Wood Chair
Processing product: Retro Vibe
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/6e986db673401a7d253e8bdee86aceb4b71b4895-4144x4144.jpg
Image uploaded successfully: image-326af1f82195f9f3a6a9e8c882691aa81f2d0617-4144x4144-jpg
✅ Imported product: Retro Vibe
Processing product: The Lucky Lamp
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/c10750f3c15f8807251977fe5e042b0a6cee55ab-305x375.png
Image uploaded successfully: image-c10750f3c15f8807251977fe5e042b0a6cee55ab-305x375-png
Category created successfully {
  _createdAt: '2025-01-23T17:13:20Z',
  _id: 'ceramics-7',
  _rev: 'NGbH4pjcDaYIxrHyZFgF0w',
  _type: 'category',
  _updatedAt: '2025-01-23T17:13:20Z',
  name: 'Ceramics',
  slug: { _type: 'slug', current: 'ceramics' }
}
✅ Imported product: The Lucky Lamp
Processing product: Pure Aura
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/db6d068faf67dab6145cadcd8e572944c6b9e935-3648x5472.jpg
Image uploaded successfully: image-db6d068faf67dab6145cadcd8e572944c6b9e935-3648x5472-jpg
✅ Imported product: Pure Aura
Processing product: Zen Table
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/73e36a9afc074b751d9f2693d2b251ba12b36276-3433x3064.jpg
Image uploaded successfully: image-2e9468c5de6d27c850d6cba74ee16bcbf200d77a-3433x3064-jpg
Category created successfully {
  _createdAt: '2025-01-23T17:13:30Z',
  _id: 'tables-9',
  _rev: 'ip9ZSUgFyzGIsQgTg0Lreq',
  _type: 'category',
  _updatedAt: '2025-01-23T17:13:30Z',
  name: 'Tables',
  slug: { _type: 'slug', current: 'tables' }
}
✅ Imported product: Zen Table
Processing product: Tropical Vibe
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/9dd7acc17365cd290757246a27b64a3d817cbe13-5501x3095.jpg
Image uploaded successfully: image-f575d30bda78fa7680a74e131a9b25add2247210-5501x3095-jpg
✅ Imported product: Tropical Vibe
Processing product: Sleek Living
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/b7565060672485ede215bbbb4d6b063b1a83b49-3024x4032.jpg
Image uploaded successfully: image-b7565060672485ede215bbbb4d6b063b1a83b49-3024x4032-jpg
✅ Imported product: Sleek Living
Processing product: Serene Seat
Uploading image: https://cdn.sanity.io/images/ri847jqu/production/19920122d503e8a40aa994d43f5f090547b48e9d-3777x5990.jpg

```

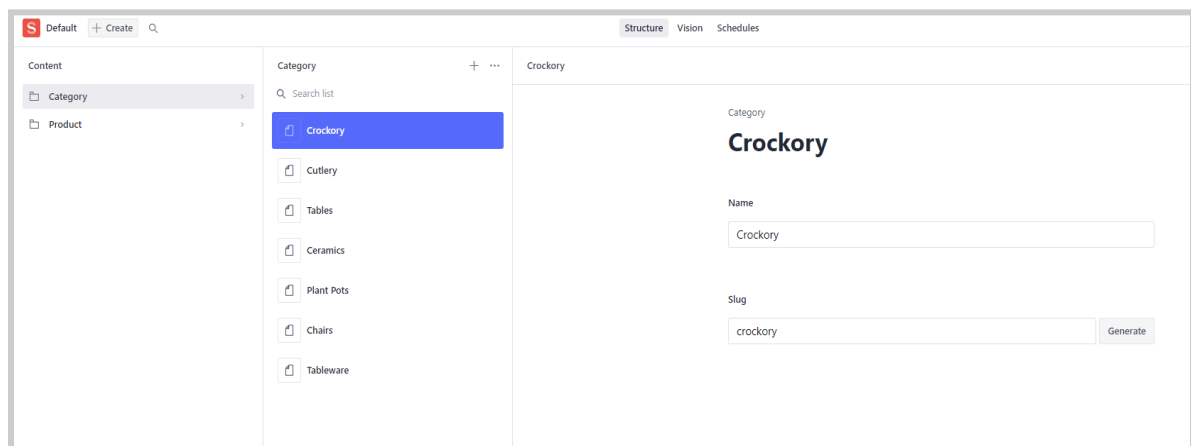
By Areesha khan

Populated sanity CMS fields:

Products



category



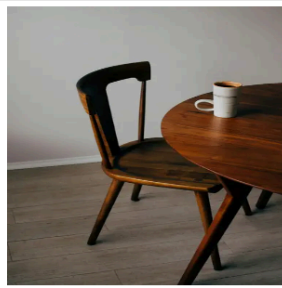
By Areesha khan

Data successfully displayed in the frontend:

Furniture Data



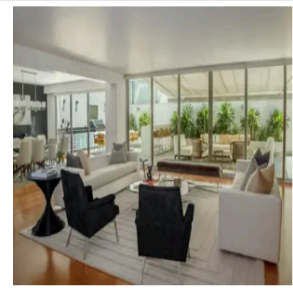
Nordic Elegance
\$280.00



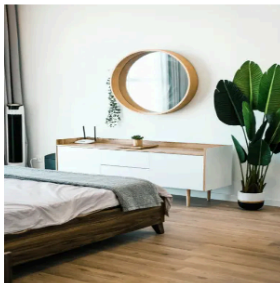
TimberCraft
\$320.00



Timeless Elegance
\$320.00



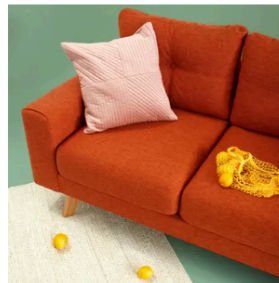
Modern Serenity
\$480.00



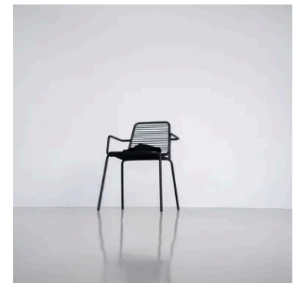
Reflective Haven
\$300.00



Sunny Chic
\$400.00



Bold Nest
\$260.00



The Steel Chair
\$250.00

By Areesha khan

Code Snippets for API integration and migration scripts:

```
import { createClient } from '@sanity/client'; 324.4k (gzipped: 85.4k)
import axios from 'axios'; 62.3k (gzipped: 23.1k)
import dotenv from 'dotenv'; 6.8k (gzipped: 3k)
import { fileURLToPath } from 'url';
import path from 'path';
import slugify from 'slugify'; 8.8k (gzipped: 3.8k)

const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

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);
    return null;
  }
}

async function createCategory(category, counter) {
  try {
    const categoryExist = await client.fetch(`*[ _type=="category" && slug.current==${slug}[0]`, { slug: category.slug });
    if (categoryExist) {
      return categoryExist._id;
    }
    const catObj = {
      _type: "category",
      _id: `${category.slug}-${counter}`,
      name: category.name,
      slug: {
        _type: 'slug',
        current: slugify(category.name || 'default-category', { lower: true, strict: true })
      }
    };
    const response = await client.createOrReplace(catObj);
    console.log('Category created successfully', response);
    return response._id;
  } catch (error) {
    console.error('Failed to create category:', category.name, error);
    return null;
  }
}

async function importData() {
  try {
    console.log('Fetching products from API...');
```

By Areesha khan

```

try {
  console.log('Fetching products from API...');
  const response = await axios.get('https://hackathon-apis.vercel.app/api/products');
  const products = response.data;
  let counter = 1;

  for (const product of products) {
    console.log(`Processing product: ${product.name}`);
    let imageRef = null;
    let catRef = null;

    if (product.image) {
      imageRef = await uploadImageToSanity(product.image);
    }
    if (product.category?.name) {
      catRef = await createCategory(product.category, counter);
    }

    const sanityProduct = {
      _id: `product-${counter}`,
      _type: 'product',
      name: product.name,
      slug: {
        _type: 'slug',
        current: slugify(product.name, { lower: true, strict: true })
      },
      price: product.price,
      category: catRef ? {
        _type: 'reference',
        _ref: catRef
      } : undefined,
      tags: product.tags || [],
      quantity: 50,
      image: imageRef ? {
        _type: 'image',
        asset: {
          _type: 'reference',
          _ref: imageRef
        }
      } : undefined,
      description: product.description || "Default product description",
      features: product.features || ["Premium material", "Handmade upholstery"],
      dimensions: product.dimensions || {
        _type: 'dimensions',
        height: "110cm",
        width: "75cm",
        depth: "50cm"
      }
    };

    await client.createOrReplace(sanityProduct);
    console.log(`✅ Imported product: ${sanityProduct.name}`);
    counter++;
  }
  console.log('✅ Data import completed!');
} catch (error) {
  console.error('Error importing data:', error);
}
}

importData();

```

By Areesha khan

Final check:

Day 3 Checklist:

API Understanding: ✓

Schema Validation: ✓

Data Migration: ✓

API Integration in Next.js: ✓

Submission Preparation: ✓

By Areesha khan