

Day 3 - API Integration Report - Avion Date: 18-01-2025

Overview:

On Day 3, the focus was on integrating the API for fetching product data into the Avion Furniture E-commerce website. This involved setting up API endpoints, fetching data using Axios, and ensuring seamless integration with the front-end components.

API Integration Process:

Step 1: API Endpoint Setup

API Endpoint: <https://hackathon-apis.vercel.app/api/products>

Data Structure: The API provides product data including:

productName, description, image, _id, features, dimensions, category, price

Step 2: Fetch Data from API

Library Used: Axios was used to fetch data from the API.

Code Snippet:

```
1 const response = await axios.get('https://hackathon-apis.vercel.app/api/products');
2 const products = response.data;
```

Product Schema:

```
1 import { defineType, defineField } from "sanity";
2
3 export const product = defineType({
4   name: "product",
5   title: "Product",
6   type: "document",
7   fields: [
8     defineField({
9       name: "category",
10      title: "Category",
11      type: "reference",
12      to: [
13        { type: "category",
14          },
15      ],
16    }),
17  ],
18  defineField({
19    name: "name",
20    title: "Title",
21    validation: (rule) => rule.required(),
22    type: "string",
23  }),
24  defineField({
25    name: "slug",
26    title: "Slug",
27    validation: (rule) => rule.required(),
28    type: "slug",
29  }),
30  defineField({
31    name: "image",
32    type: "image",
33    validation: (rule) => rule.required(),
34    title: "Product Image",
35  }),
36  defineField({
37    name: "price",
38    type: "number",
39    validation: (rule) => rule.required(),
40    title: "Price",
41  }),
42  defineField({
43    name: "quantity",
44    title: "Quantity",
45    type: "number",
46    validation: (rule) => rule.min(0),
47  }),
48  defineField({
49    name: "tags",
50    type: "array",
51    title: "Tags",
52    of: [
53      { type: "string",
54        },
55    ],
56  }),
57  defineField({
58    name: "description",
59    title: "Description",
60    type: "text",
61    description: "Detailed description of the product",
62  }),
63  defineField({
64    name: "features",
65    title: "Features",
66    type: "array",
67    of: [ { type: "string" } ],
68    description: "List of key features of the product",
69  }),
70  defineField({
71    name: "dimensions",
72    title: "Dimensions",
73    type: "object",
74    fields: [
75      { name: "height", title: "Height", type: "string" },
76      { name: "width", title: "Width", type: "string" },
77      { name: "depth", title: "Depth", type: "string" },
78    ],
79    description: "Dimensions of the product",
80  }),
81 ],
82 },
83 );
```

Frontend Display:

New Ceramics



Marble Ease

£419



Vase Set

£150



The Dandy chair

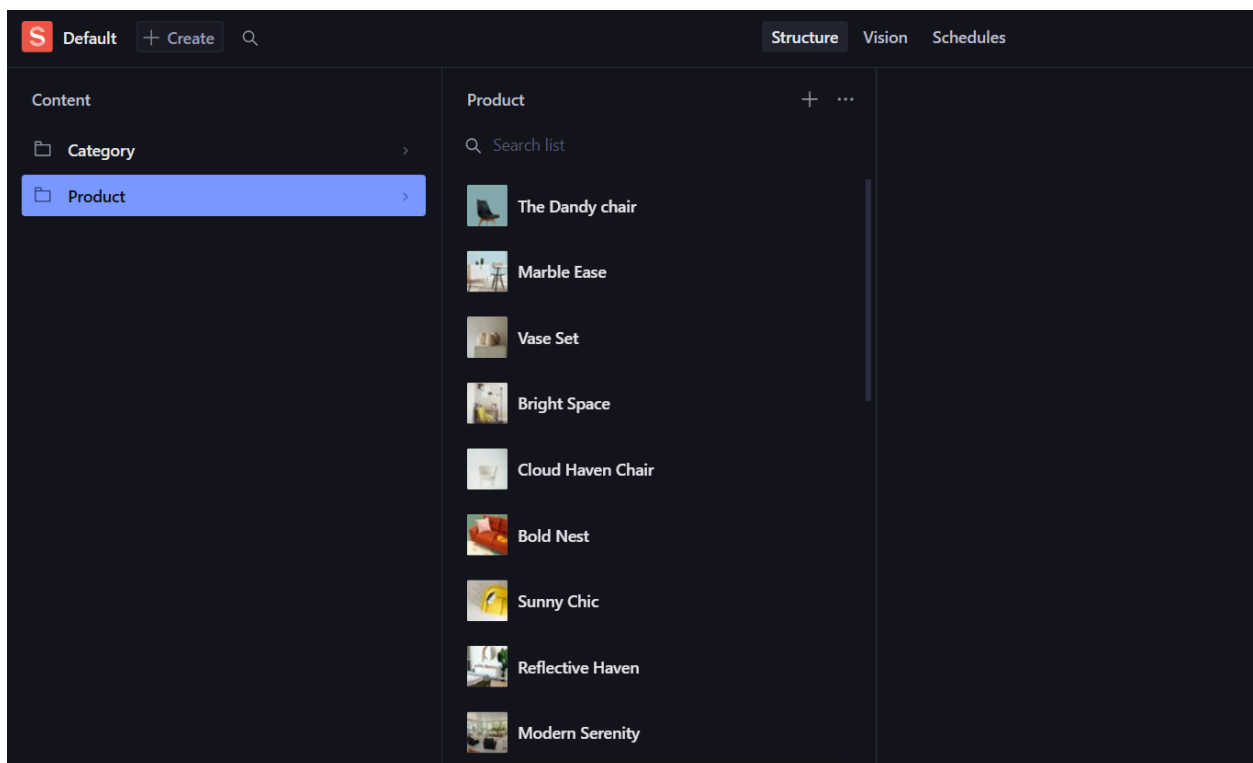
£150



Bright Space

£180

Populated Sanity CMS Fields:



M.AREEB MALIK

Migration Script:

```
1 import axios from "axios";
2 import { client } from "../sanityClient.js";
3 import slugify from "slugify";
4
5 async function uploadImageToSanity(imageUrl: string): Promise<string | null> {
6   try {
7     // Fetch the image from the URL and convert it to a buffer
8     const response = await axios.get(imageUrl, {
9       responseType: "arraybuffer",
10      timeout: 10000,
11    });
12    const buffer = Buffer.from(response.data);
13
14    // Upload the image to Sanity
15    const asset = await client.assets.upload("image", buffer, {
16      filename: imageUrl.split("/").pop(), // Extract the filename from URL
17    });
18
19    // Debugging: Log the asset returned by Sanity
20    console.log("Image uploaded successfully:", asset);
21
22    return asset._id; // Return the uploaded image asset reference ID
23  } catch (error) {
24    console.error("X Failed to upload image", imageUrl, error);
25    return null;
26  }
27  //throw error;
28 }
29
30 interface Category {
31   _id: string;
32   _type: string;
33   name: string;
34   slug: string;
35 }
36
37 async function createCategory(category: Category, counter: number) {
38   try {
39     const categoryExist = await client.fetch(
40       `*[_type=="category" & slug==${slug}][]`,
41       { slug: category.slug }
42     );
43     if (categoryExist) {
44       return categoryExist._id;
45     }
46     const catObj = {
47       _type: "category",
48       _id: category.slug + "-" + counter,
49       name: category.name,
50       slug: category.slug,
51     };
52     const response = await client.createOrReplace(catObj);
53
54     // Debugging: Log the asset returned by Sanity
55     console.log("Category created successfully", response);
56
57     return response._id; // Return the uploaded image asset reference ID
58   } catch (error) {
59     console.error("X Failed to category", category.name, error);
60     //throw error;
61   }
62 }
63
64 async function importData() {
65   try {
66     // Fetch data from external API
67     const response = await axios.get(
68       "https://hackernews-api.vercel.app/api/products"
69     );
70     const products = response.data;
71     //console.log(products);
72     let counter = 1;
73     // Iterate over the products
74     for (const product of products) {
75       let imageUrl = null;
76       let catRef = null;
77
78       // Upload image and get asset reference if it exists
79       if (product.image) {
80         //imageUrl = await uploadImageToSanity(product.imageUrl);
81         imageUrl = await uploadImageToSanity(product.image);
82       }
83
84       if (product.category.name) {
85         catRef = await createCategory(product.category, counter);
86       }
87
88       const sanityProduct = {
89         _id: "product-${counter}", // Prefix the ID to ensure validity
90         _type: "product",
91         name: product.name,
92         slug: {
93           _type: "slug",
94           current: slugify(product.name || "default-product", {
95             lower: true, // Ensure the slug is lowercase
96             strict: true, // Remove special characters
97           }),
98         },
99         price: product.price,
100        category: {
101          _type: "reference",
102          _ref: catRef || catRef : undefined,
103        },
104        tags: product.tags > product.tags : [],
105        quantity: 10,
106        image: imageUrl
107        ? {
108          _type: "image",
109          asset: {
110            _type: "reference",
111            _ref: imageUrl, // Set the correct asset reference ID
112          },
113        }
114        : undefined,
115        description: product.description
116        ? product.description
117        : "A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather upholstery.",
118        features: product.features
119        ? product.features
120        : [
121          "Premium material",
122          "Handmade upholstery",
123          "Quality timeless classic",
124        ],
125        dimensions: product.dimensions
126        ? product.dimensions
127        : {
128          _type: "dimensions", // Custom object type for dimensions
129          height: "180cm",
130          width: "75cm",
131          depth: "55cm",
132        },
133      };
134      counter++;
135      // Log the product before attempting to upload it to Sanity
136      console.log("Uploading product:", sanityProduct);
137
138      // Import data into Sanity
139      await client.createOrReplace(sanityProduct);
140      console.log("Imported product: ${sanityProduct.name}");
141    }
142
143    console.log("Data import completed!");
144  } catch (error) {
145    console.error("X Error importing data:", error);
146  }
147 }
148
149 importData();
150
```

Final Checklist:

| API Understanding | Schema Validation | Data Migration | API Integration in Next.js | Submission Preparation |
|----------------------|----------------------|-------------------|----------------------------------|---------------------------|
| ✓ | ✓ | ✓ | ✓ | ✓ |

M.AREEB MALIK