

Day 3 - API Integration Report

Step 1: Understand the API

- **Get the API details:** First, I would check the provided API documentation to understand the endpoints (like /products, /categories) and the kind of data it returns.

For example, the API might return data like this:

- id: Product ID
- name: Product name
- price: Product price
- description: Product description
- category: Product category

Step 2: Create or Update Sanity CMS Schema

- **Map API data to Sanity:** I would ensure that the data returned by the API fits into my Sanity CMS schema. I might need to adjust field names to match.

Example:

- API returns product_name, but in Sanity, I use name.
- Update the schema file in Sanity (like product.ts) to reflect this:

```

1  export const productSchema = {
2      name: 'product',
3      title: 'Product',
4      type: 'document',
5      fields: [
6          {
7              name: 'productName',
8              title: 'Product Name',
9              type: 'string',
10         },
11         {
12             name: 'category',
13             title: 'Category',
14             type: 'string',
15         },
16         {
17             name: 'price',
18             title: 'Price',
19             type: 'number',
20         },
21         {
22             name: 'inventory',
23             title: 'Inventory',
24             type: 'number',
25         },
26         {
27             name: 'colors',
28             title: 'Colors',
29             type: 'array',
30             of: [{ type: 'string' }],
31         },
32         {
33             name: 'status',
34             title: 'Status',
35             type: 'string',
36         },
37         {
38             name: 'image',
39             title: 'Image',
40             type: 'image', // Using Sanity's image type for image field
41             options: {
42                 hotspot: true,
43             },
44         },
45         {
46             name: 'description',
47             title: 'Description',
48             type: 'text',
49         },
50     ],
51 }
52

```

Step 3: Migrate Data into Sanity CMS

```

1  import { createClient } from '@sanity/client';
2  import axios from 'axios';
3  import dotenv from 'dotenv';
4  import { fileURLToPath } from 'url';
5  import path from 'path';
6
7  // Load environment variables from .env.local
8  const __filename = fileURLToPath(import.meta.url);
9  const __dirname = path.dirname(__filename);
10  dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12  // Create Sanity client
13  const client = createClient({
14    projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
15    dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
16    useCdn: false,
17    token: process.env.SANITY_API_TOKEN,
18    apiVersion: '2021-08-31'
19  });
20
21
22  async function uploadImageToSanity(imageUrl) {
23    try {
24      console.log('Uploading image: ${imageUrl}');
25      const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
26      const buffer = Buffer.from(response.data);
27      const asset = await client.assets.upload('image', buffer, {
28        filename: imageUrl.split('/').pop()
29      });
30      console.log('Image uploaded successfully: ${asset._id}');
31      return asset._id;
32    } catch (error) {
33      console.error('Failed to upload image:', imageUrl, error);
34      return null;
35    }
36  }
37
38  async function importData() {
39    try {
40      console.log('migrating data please wait...');
41
42      // API endpoint containing car data
43      const response = await axios.get('https://template-03-api.vercel.app/api/products');
44      const products = response.data.data;
45      console.log("products ==> ", products);
46
47      for (const product of products) {
48        let imageRef = null;
49        if (product.image) {
50          imageRef = await uploadImageToSanity(product.image);
51        }
52
53        const sanityProduct = {
54          _type: 'product',
55          productName: product.productName,
56          category: product.category,
57          price: product.price,
58          inventory: product.inventory,
59          colors: product.colors || [], // Optional, as per your schema
60          status: product.status,
61          description: product.description,
62          image: imageRef ? {
63            _type: 'image',
64            asset: {
65              _type: 'reference',
66              _ref: imageRef,
67            },
68          } : undefined,
69        };
70
71        await client.create(sanityProduct);
72      }
73
74      console.log('Data migrated successfully!');
75    } catch (error) {
76      console.error('Error in migrating data ==> ', error);
77    }
78  }
79
80  importData();

```

Step 4 : Integrated Api Into Sanity:

