

Day 3 - API Integration and Data Migration Report

Objectives

- Integrate APIs into a Next.js project.
- Migrate data from external APIs into Sanity CMS.
- Validate and align schemas with data sources.

Overview of Steps

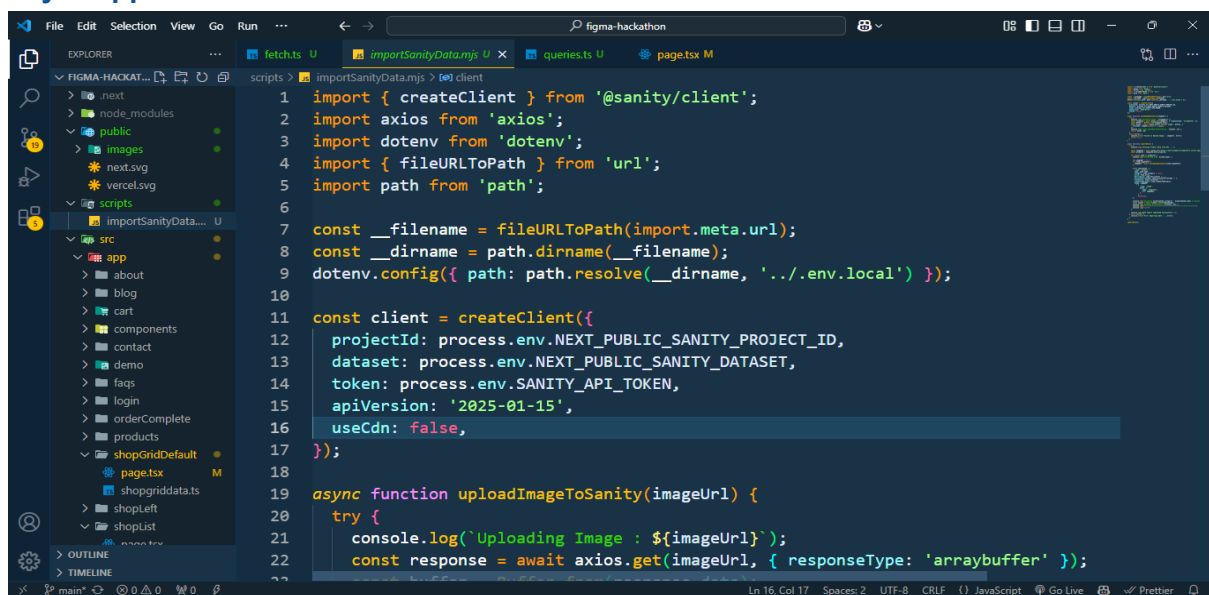
This report outlines the process of integrating data from MockAPI into Sanity CMS, including:

- Fetching data from MockAPI.
- Aligning API data with the Sanity schema.
- Configuring the environment for data migration.
- Automating data migration using a script.

1. Fetching Data from MockAPI

Data was fetched from MockAPI using a script created within the Next.js project. The functionality was verified using tools such as Postman.

Key Snippet:



```
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 const __filename = fileURLToPath(import.meta.url);
8 const __dirname = path.dirname(__filename);
9 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
10
11 const client = createClient({
12   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
13   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
14   token: process.env.SANITY_API_TOKEN,
15   apiVersion: '2025-01-15',
16   useCdn: false,
17 });
18
19 async function uploadImageToSanity(imageUrl) {
20   try {
21     console.log(`Uploading Image : ${imageUrl}`);
22     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
```

Verification:

API responses were validated using Postman to confirm proper data retrieval.

2. Aligning API Data with Sanity Schema

API data was compared to the Sanity schema to ensure compatibility. Necessary adjustments were made, such as mapping API fields to schema fields.

Schema Snippet:



```
src > sanity > schemaTypes > products.ts > default > fields > name
1  export default {
2    name: "product",
3    type: "document",
4    title: "Product",
5    fields: [
6      {
7        name: "name",
8        type: "string",
9        title: "Name",
10       validation: (Rule: any) => Rule.required().error("Name is required"),
11      },
12      {
13        name: "image",
14        type: "image",
15        title: "Image",
16        options: {
17          hotspot: true,
18        },
19        description: "Upload an image of the product.",
20      },
21    ],
22  },
```

3. Environment Configuration

Sanity CMS was configured with credentials stored in a `.env.local` file. These included:

- `SANITY_PROJECT_ID`
- `SANITY_DATASET`
- `SANITY_TOKEN`

This setup ensured secure and efficient interaction with Sanity CMS.

4. Modifying package.json

To streamline the process of running the migration script, I added a custom command to the `package.json` file. This allows me to run the migration script with a simple command in the terminal.

```
package.json > ...
1  {
2    "name": "figma-hackathon",
3    "version": "0.1.0",
4    "private": true,
5    "scripts": {
6      "dev": "next dev",
7      "build": "next build",
8      "start": "next start",
9      "lint": "next lint",
10     "import-data": "node scripts/importSanityData.mjs"
11   },
```

5. Importing Data into Sanity

With everything configured, I ran the migration script to import the data into Sanity. The following command triggered the data migration:

```
"npm run import-data"
```

Conclusion

This process successfully automated the migration of data from MockAPI into Sanity CMS. The essential steps included:

1. Fetching data from MockAPI.
 2. Aligning API fields with Sanity schema fields.
 3. Setting up the environment securely.
 4. Executing a migration script.
-

Achievements

- API Integration: ✓
- Schema Validation: ✓
- Data Migration: ✓