

API Integration Report

Step 1: Fetch Data from API and Store in Sanity

1. Fetch data from the external API using tools like @sanity/client, axios or dotenv.
2. Process and format the API response to match the Sanity schema.
3. Use Sanity's client library to insert data into Sanity CMS using its GROQ query language.
4. Validate successful data insertion by checking the Sanity Studio dashboard.

```
scripts > $ importSanityData.mjs > ...
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 // Create Sanity client
12 const client = createClient({
13   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
14   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
15   useCdn: false,
16   token: process.env.SANITY_API_TOKEN,
17   apiVersion: '2021-08-31'
18 })
19
20 Tabnine | Edit | Test | Explain | Document
21 async function uploadImageToSanity(imageUrl) {
22   try {
23     console.log('Uploading image: ${imageUrl}')
24     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' })
25     const buffer = Buffer.from(response.data)
26     const asset = await client.assets.upload('image', buffer, {
27       filename: imageUrl.split('/').pop()
28     })
29     console.log('Image uploaded successfully: ${asset.id}')
30     return asset.id
31   } catch (error) {
32     console.error('Failed to upload image:', imageUrl, error)
33     return null
34   }
35 }
```

Step 2: Fetch Data from Sanity

1. Use Sanity's client library to query the data using GROQ.
2. Write queries to retrieve specific fields or filter data (e.g., fetch all products or specific categories).
3. Handle API responses, including errors or empty datasets.
4. Test the queries in Sanity's Vision tool to ensure accuracy.

```
"use client"
import React, { useEffect, useState } from "react";
import Image from "next/image";
import Link from "next/link";
import { client } from "@sanity/lib/client";

Tabnine | Edit | Explain
const Shop = () => {
  const [products, setProducts] = useState<any[]>([]);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    const fetchData = async () => {
      try {
        const query = `*[_type == "product"]{
          id,
          name,
          description,
          price,
          quantity,
          discountPercentage,
          rating,
          tags,
          "image": image.asset->url
        }`;
        const result = await client.fetch(query);
        setProducts(result);
      } catch (error) {
        console.error("Error fetching products:", error);
      } finally {
        setLoading(false);
      }
    };
    fetchData();
  }, []);
}
```

Step 3: Display Data in Frontend (Shop.tsx)

1. Create a React component (Shop.tsx) to fetch data from Sanity at runtime.
2. Use the useEffect hook to fetch data when the component mounts (if using client-side rendering).
3. Pass data as props if using server-side rendering (getServerSideProps or getStaticProps in Next.js).

E-Commerce Accessories and Products



Mens Casual Premium Slim Fit T-Shirts

\$22.3

Slim-fitting style, contrast raglan long sleeves, three-button henley placket, light weight & soft fabric for breathable and comfortable wearing. And Solid stitched shirts with round neck made for durability and a great fit for casual fashion wear and diehard baseball fans. The Henley style



Mens Cotton Jacket

\$55.99

great outerwear jackets for Spring/Autumn/Winter, suitable for many occasions, such as working, hiking, camping, mountain/rock climbing, cycling, travelling or other outdoors. Good gift choice for you or your family member. A warm hearted love to Father, husband or son in this Thanksgiving or Christmas Day.



Pierced Owl Rose Gold Plated Stainless Steel Double

\$10.99

Rose Gold Plated Double Flared Tunnel Plug Earrings. Made of 316L Stainless Steel



WD 2TB Elements Portable External Hard Drive - USB 3.0

\$64

USB 3.0 and USB 2.0 Compatibility Fast data transfers improve PC Performance High Capacity; Compatibility Formatted NTFS for Windows 10, Windows 8.1, Windows 7; Reformatting may be required for other operating systems; Compatibility may vary

4. Map over the data and display it in the desired UI format, including images, descriptions, and prices.

Step 4: Flow of Data

- **Data Flow Diagram:**

1. **External API:** Data is fetched using an HTTP request (GET /endpoint).
2. **Sanity CMS:** Processed data is inserted into Sanity using GROQ queries.
3. **Frontend (Shop.tsx):** Data is retrieved from Sanity and displayed to users.
4. **Error Handling:** Any errors (e.g., API failure, data mismatch) are logged and handled gracefully at each stage.

Example Flow:

1. **Fetch Data from API**

- External API endpoint: `https://api.example.com/products`
- Data format: JSON (e.g., `{ id, name, price, image }`).

2. **Store in Sanity**

- Sanity schema: `{ _type: "product", name, price, description, image }`.

3. **Fetch Data in Frontend**

- GROQ query: `*[_type == "product"]{ _id, name, price, image }`.

4. **Display in Shop.tsx**

- Data mapped into cards with Image, name, and price fields.

Conclusion:

This report outlines a structured API integration process to fetch data, store it in Sanity CMS, and display it in the frontend. The flow ensures that data is processed efficiently and rendered accurately, providing a seamless user experience.

Prepared by Kashaf Tariq