

Day 3

API Integration Report

Prepared by: Syed Nauman Navaid

Slot: Thursday morning 9 to 12

API Integration Process

On Day 3, we were provided with an API to help us migrate product data into Sanity CMS. The template I was working on (Template 0) was prepared by Sir Ahmed Raza and Bilal Fareed. The schema was provided to correctly structure the data for Sanity CMS.

However, there were some issues with the original scripts and instructions. The integration process was based on older practices, particularly using the **Pages Router**, which is outdated as we are now using the **App Router** in **Next.js 14**.

Issues in the Initial Process:

1. **Using the Pages Router:** The original instructions were designed around the older Pages Router (under pages/api), but Next.js 14 uses the App Router structure.
2. **API Integration Approach:** The original process involved API routes defined inside pages/api. In the App Router, we define API routes inside app/api instead.

Corrected Process for API Integration in Next.js 14:

Next.js 14 introduces the **App Router**, and API routes are created differently. Here's the new structure and an example of how to correctly implement the API route

1. Folder Structure for API Route:

```
/app
└── /api
    └── /fetch-and-insert
        └── route.ts
```

So after correcting this issue I created the api route like mentioned above and place the script in it. I had to do some modifications in the script.

2. Original script (Provided):

```
3. import axios from 'axios';
4. import client from '../sanityClient';
5. export default async function handler(req, res) {
6.   try {
7.     // Fetch data from the API
8.     const { data } = await axios.get('https://template-0-beta.vercel.app/api/product');
9.     // Insert each product into Sanity
10.    for (const product of data) {
```

```

11. await client.create({
12.  _type: 'product',
13.  id: product.id,
14.  name: product.name,
15.  imagePath: product.imagePath,
16.  price: parseFloat(product.price),
17.  description: product.description,
18.  discountPercentage: product.discountPercentage,
19.  isFeaturedProduct: product.isFeaturedProduct,
20.  stockLevel: product.stockLevel,
21.  category: product.category,
22. });
23.
24. res.status(200).json({ message: 'Data inserted successfully!' });
25. } catch (error) {
26.  console.error(error);
27.  res.status(500).json({ error: 'Failed to fetch or insert data' });
28. }
29. }
30.

```

3. Modified Script (App Router)

The script is written for the App Router in Next.js 14, where API routes are placed under the `app/api` directory, and the handler function is defined with the GET method.

API Route Location: `app/api/products/route.ts`

Method: The GET method is used, which aligns with the new App Router design where request methods (e.g., GET, POST) are handled explicitly.

```

import axios from 'axios';
import { client } from '@sanity/lib/client';
import { NextResponse } from 'next/server';

export async function GET(request: Request) {
  try {
    // Fetch data from the external API
    const { data } = await axios.get('https://template-0-beta.vercel.app/api/product');

    // Insert each product into Sanity
    const insertPromises = data.map((product: any) =>
      client.createOrReplace({
        _type: 'product',
        _id: product.id, // Use product ID as the document ID

```

```

        name: product.name,
        imagePath: product.imagePath,
        price: parseFloat(product.price),
        description: product.description,
        discountPercentage: product.discountPercentage,
        isFeaturedProduct: product.isFeaturedProduct,
        stockLevel: product.stockLevel,
        category: product.category,
    })
    );

    // Wait for all products to be inserted
    await Promise.all(insertPromises);
    console.log(insertPromises);

    return NextResponse.json({ message: 'Data migration successful!' });
} catch (error) {
    console.error('Migration error:', error);
    return NextResponse.json({ error: 'Data migration failed.' }, { status: 500
});
}
}

```

4. Key Differences:

1. Route Location

- **Original Script:**
 - The route is located in `pages/api/fetch-and-insert.js` (following the Pages Router structure).
- **Modified Script:**
 - The route is located in `app/api/fetch-and-insert/route.ts` (following the App Router structure in Next.js 14).

2. Request Method

- **Original Script:**
 - Does not specify a method explicitly and defaults to `POST` under the `pages/api` router structure.
- **Modified Script:**
 - Explicitly defines the `GET` method with `export async function GET(request: Request)`, aligning with App Router conventions in Next.js 14.

3. API Route Handler

- **Original Script:**

- Uses a generic `export async function handler(req, res)` to handle the request and response.

- **Modified Script:**

- Specifically uses `export async function GET(request: Request)` to define the handler, which is the recommended approach in the App Router.

4. Response Handling

- **Original Script:**

- Utilizes `res.status(200).json()` and `res.status(500).json()` to send responses directly.

- **Modified Script:**

- Leverages `NextResponse.json()` to send responses, adhering to Next.js 14's App Router guidelines.

5. Data Insertion Logic

- **Original Script:**

- Inserts products one by one into Sanity CMS using `await client.create()` inside a loop.

- **Modified Script:**

- Employs `client.createOrReplace()` to either insert or replace documents in Sanity. This method avoids duplicates by ensuring the product ID is used as the document ID. It also utilizes `Promise.all()` to insert all products concurrently, significantly improving efficiency.

6. Error Handling

- **Original Script:**

- Logs errors directly and responds with a 500 status in case of failure.

- **Modified Script:**

- Catches errors, logs them, and returns a descriptive error message using `NextResponse.json()`, which is consistent with the App Router's response structure.

7. Client Import

• Original Script:

- Imports the Sanity client using a relative path: `import client from '../../../../sanityClient';`
-

• Modified Script:

- Imports the Sanity client using an alias path: `import { client } from '@sanity/lib/client';`, which is more consistent and organized in Next.js 14 projects.

❖ Console screenshot




```
[
  {
    imagePath: 'https://plus.unsplash.com/premium_photo-1668073439372-2cea1222b7?q=80&w=1887&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVuLWVudB8fHx8fA%3D%3D',
    _rev: '0e0ztuJY6NMdyb1BERZSbv',
    _createdAt: '2025-01-17T19:41:54Z',
    _id: '1',
    isFeaturedProduct: true,
    category: 'Chair',
    _type: 'product',
    description: 'A sleek outdoor chair with natural wooden elements and a modern look.',
    stockLevel: 25,
    discountPercentage: 10,
    price: 1200,
    name: 'Chair Wibe',
    _updatedAt: '2025-01-17T19:41:54Z'
  },
  {
    _updatedAt: '2025-01-17T19:41:54Z',
    _rev: 'YjfIo1g1LQZHJZg27P55i7',
    discountPercentage: 10,
    isFeaturedProduct: false,
    category: 'Table',
    price: 900,
    _createdAt: '2025-01-17T19:41:54Z',
    name: 'Alpha Table',
    _id: '10',
    imagePath: 'https://plus.unsplash.com/premium_photo-1681412205470-77848a519359?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVuLWVudB8fHx8fA%3D%3D',
    _type: 'product',
    description: 'A sturdy oak chair with a sleek and minimalist design.',
    stockLevel: 18
  },
]
```

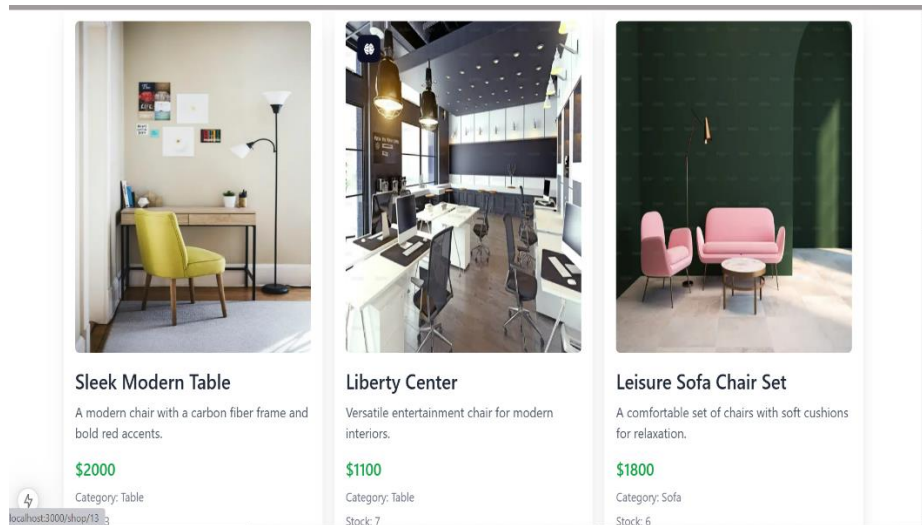
```

    _rev: 'YjfIo1g1LQZHJZg27P55sb',
    description: 'Classic wishbone chair with a dark walnut frame and cord seat.',
    price: 750,
    name: 'Replica Table',
    _updatedAt: '2025-01-17T19:41:54Z',
    imagePath: 'https://images.unsplash.com/photo-1499933374294-4584851497cc?q=80&w=1935&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fD
    MhXwaG90by1wYldlfHx8fGVufDB8fHx8fA%3D%3D',
    stockLevel: 25,
    discountPercentage: 10,
    _createdAt: '2025-01-17T19:41:54Z',
    _id: '11',
    isFeaturedProduct: false,
    category: 'Table',
    _type: 'product'
  },
  {
    _id: '12',
    isFeaturedProduct: true,
    _updatedAt: '2025-01-17T19:41:54Z',
    _type: 'product',
    price: 2000,
    name: 'Sleek Modern Table',
    stockLevel: 3,
    discountPercentage: 0,
    _createdAt: '2025-01-17T19:41:54Z',
    category: 'Table',
    imagePath: 'https://images.unsplash.com/photo-1486946255434-2466348c2166?q=80&w=1887&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fD
    MhXwaG90by1wYldlfHx8fGVufDB8fHx8fA%3D%3D',
    _rev: '0e0ztuJY6NMdyblBERZSoh',
    description: 'A modern chair with a carbon fiber frame and bold red accents.'
  },
}

```

❖ Front-end data fetched:

 <p>Chair Wibe</p> <p>A sleek outdoor chair with natural wooden elements and a modern look.</p> <p>\$1200</p>	 <p>Alpha Table</p> <p>A sturdy oak chair with a sleek and minimalist design.</p> <p>\$900</p>	 <p>Replica Table</p> <p>Classic wishbone chair with a dark walnut frame and cord seat.</p> <p>\$750</p>
--	---	--



❖ Populated Sanity fields:

