

API Integration and fetching data from sanity- Furniro

Reviewed API Documentation:

- I carefully reviewed the provided **Sanity CMS** documentation to understand the data schema for products.
 - I manually added product data into **Sanity CMS** using the Sanity Studio.
 - I ensured that the schema fields in Sanity matched the API structure, including field names and data types.
-

Dynamic API Fetching:

- I dynamically fetched product data from **Sanity CMS** using its API in my **Next.js** frontend.
- I utilized the official Sanity client for data retrieval.

Steps:

Install Sanity Client:

I installed the Sanity client for Next.js:

bash


CopyEdit

```
npm install @sanity/client
```

1.

Create API Utility File:

I created a utility file (`sanityClient.ts`) for Sanity configuration:



```
1  import { createClient } from 'next-sanity'
2
3
4  export const client = createClient(
5    {
6      projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
7      dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
8      apiVersion: '2023-05-03',
9      useCdn: false,
10   })
11
```

2.

Fetch Data in the Component:

I dynamically fetched products in my frontend using the `fetch` function:

```
3.  useEffect(() => {
4.    const fetchProducts = async () => {
5.      try {
6.        console.log("Fetching products...");
7.        const query = '*[_type == "product"]';
8.        const fetchedProducts = await client.fetch(query);
9.        console.log("Fetched products:", fetchedProducts);
10.       setProducts(fetchedProducts);
11.      } catch (error) {
12.        console.error("Error fetching products:", error);
13.        setError("Failed to fetch products. Please try again
        later.");
      }
    }
  })
}
```

```
14.     } finally {  
15.         setLoading(false);  
16.     }  
17. };  
18. fetchProducts();  
19. }, []);
```

Schema Adjustment in Sanity CMS:

- I adjusted the schema in Sanity Studio to ensure compatibility with the API data.
 - Key fields include:
 - **name**: String
 - **description**: Text
 - **price**: Number
 - **image**: Image type
 - **category**, **colors**, and **sizes**: Array
-

Results:

In this task, I successfully:

- Added product data manually to Sanity CMS.
 - Fetched data dynamically from Sanity using its API in my Next.js frontend.
 - Ensured accurate data rendering in the UI with compatibility between the schema and API.
-

Task Status:

- **Sanity Data Entry** ✓
 - **Dynamic API Fetching** ✓
 - **Frontend Rendering** ✓
 - **Schema Compatibility** ✓
-

Achievements:

This task enhanced my practical skills in:

- Configuring and using Sanity CMS with Next.js.
- Creating dynamic and scalable data pipelines for modern web applications.