

ABDUL SAMAD (SL)

Friday (9-12)

API Integration Report - [Leather Outwear website]

Reviewed API Documentation:

- o I carefully read the provided API documentation to understand the available endpoint (/products).
- o I identified the structure of the data returned by the API, including field names and data types.
- o And I changed my template from template-4 to template-1.

Set Up API Calls:

- o I used Postman to test the API endpoint and ensure the data was returned correctly. I created the .mjs file, fetched the API data within it, and then uploaded the data to Sanity.
- o There was a mistake in the provided repository for template 1, where we couldn't upload the data to sanity, i think many of you might be facing this issue, so here is the code :
- o But first install these packages :

```
npm i dotenv axios
```

- o Now Paste the following code in the `scripts/importData.mjs`:
- o Make sure to create it outside the src folder.

```
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.NEXT_PUBLIC_SANITY_TOKEN,
  apiVersion: '2021-08-31',
});

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);

    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop(),
    });

    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error(`Failed to upload image: ${imageUrl}`, error.message);
    return null;
  }
}

async function importData() {
  try {
    console.log('Fetching products from API...');
    const response = await
    axios.get('https://templatel-neon-nu.vercel.app/api/products');
    const products = response.data;
  }
}
```

```

    console.log(`Fetched ${products.length} products`);
    for (const product of products) {
        console.log(`Processing product: ${product.name}`);
        let imageRef = null;
        if (product.imageUrl) {
            imageRef = await uploadImageToSanity(product.imageUrl);
        }
        const sanityProduct = {
            _type: 'product',
            name: product.name,
            description: product.description,
            price: product.price,
            discountPercent: product.discountPercent,
            tags: product.category ? [product.category] : [],
            sizes: product.sizes || [],
            colors: product.colors || [],
            isNew: product.isNew || false,
            image: imageRef
            ? {
                _type: 'image',
                asset: {
                    _type: 'reference',
                    _ref: imageRef,
                },
            }
            : undefined,
        };
        console.log('Uploading product to Sanity:', sanityProduct.name);
        const result = await client.create(sanityProduct);
        console.log(`Product uploaded successfully: ${result._id}`);
    }

    console.log('Data import completed successfully!');
} catch (error) {
    console.error('Error importing data:', error.message);
}
importData();

```

o Make sure to have sanity token, project-id, and datasets in the env files:

o create the file in src/sanity/schemaTypes/product.ts:

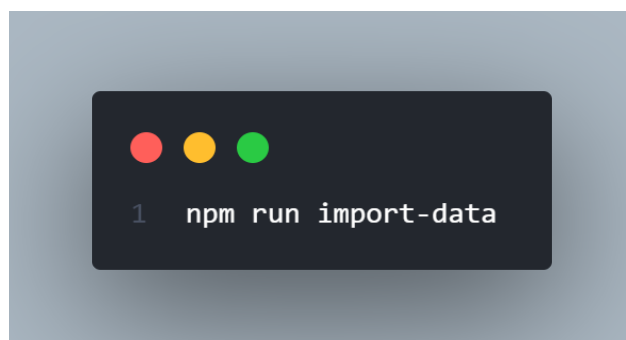
```
1  import { defineField } from "sanity";
2
3  export default defineField({
4    name: "product",
5    type: "document",
6    title: "Product",
7    fields: [
8      { name: "name", type: "string", title: "Name" },
9      {
10        name: "slug",
11        type: "slug",
12        title: "Slug",
13        options: {
14          source: "name", // Automatically generate the slug from the name
15          maxLength: 96, // Limit the slug length
16        },
17      },
18      { name: "description", type: "text", title: "Description" },
19      { name: "price", type: "number", title: "Price" },
20      { name: "discountPercent", type: "number", title: "Discount Percent" },
21      { name: "tags", type: "array", title: "Tags", of: [{ type: "string" }] },
22      { name: "sizes", type: "array", title: "Sizes", of: [{ type: "string" }] },
23      { name: "colors", type: "array", title: "Colors", of: [{ type: "string" }] },
24      { name: "isNew", type: "boolean", title: "Is New" },
25      {
26        name: "image",
27        type: "image",
28        title: "Image",
29        options: { hotspot: true },
30      },
31    ],
32  });
33
```

o Now update the package.json file :

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It displays a JSON configuration for the 'scripts' field of a package.json file, with line numbers 1 through 7 on the left.

```
1  "scripts": {  
2    "dev": "next dev",  
3    "build": "next build",  
4    "start": "next start",  
5    "lint": "next lint",  
6    "import-data": "node scripts/importData.mjs"  
7  },
```

o Now run the following command:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It displays the command to run the 'import-data' script, with line number 1 on the left.

```
1  npm run import-data
```

o Now the data should be inserted in the sanity.

o In this project, I successfully integrated the provided API into my Next.js frontend and migrated data into Sanity CMS. I adjusted the schema to match the API data structure and ensured the data was accurately displayed in the frontend. This exercise helped me gain practical experience in API integration, data migration, and schema validation, which are essential skills for building scalable marketplaces.

Tasks	Status
- API Understanding	✓
- Schema Validation	✓
- Data Migration	✓
- API Integration in Next.js	✓
- Submission Preparation	✓

RESULTS:

S

Default

+ Create

Q

StructureVisionSchedules


Content


Product


Product


Casual Green Bomber Jacket


Search list


Classic Black Straight-Leg Jeans


Gradient Graphic T-shirt


Classic White Pullover Hoodie


Casual Green Bomber Jacket


Classic Black Long Sleeve Button-...


Classic Black Pullover Hoodie


Gray Slim-Fit Jogger Pants


COURAGE GRAPHIC T-SHIRT


Sleeve Stripe T-Shirt

Vertical Striped Shirt

LOOSE FIT BERMUDA SHORTS

Checkered Shirt

Black Striped T-Shirt

Beige Slim-Fit Jogger Pants

Casual Green Bomber Jacket

Add comment

...

↩

Name

Casual Green Bomber Jacket

Slug

casual-green-bomber-jacket

Generate

Description

This stylish green bomber jacket offers a sleek and modern twist on a classic design. Made from soft and comfortable fabric, it features snap buttons and ribbed cuffs, giving it a sporty yet refined look. The minimalist style makes it perfect for layering over casual t-shirts or hoodies. Whether you're out with friends or just lounging, this jacket provides a laid-back yet fashionable vibe. Its muted green color adds a subtle, earthy tone that pairs well with a variety of outfits, making it a versatile addition to your casual wardrobe.

Price

300

Published 5 hr. ago


↑ Publish

...

SHOP.CO

ShopOn SaleNew ArrivalsBrands

QSearch for products...



Home > All Products

All Products

Showing 1-12 of 100 ProductsSort by: Most Popular


T-shirts>Shorts>Shirts>Hoodie>Jeans>

Price\$50\$200


Colors

SizeXX-SmallX-SmallSmallMediumLargeX-Large2X-Large3X-Large4X-Large


Dress Style




Casual Green Bomber Jacket\$300\$260-20%





Sleeve Stripe T-Shirt\$130\$90-40%



Vertical Striped Shirt\$229\$189-20%










SHOP.CO

ShopOn SaleNew ArrivalsBrands

QSearch for products...



Home > Shop > tshirt



Sleeve Stripe T-Shirt

★★★★★ 4 / 5

\$130

This product is a vibrant orange and black striped t-shirt with a sporty design. The shirt features vertical pinstripes in black on an orange base, complemented by contrasting black raglan sleeves for a casual, athletic-inspired look. Ideal for adding a bold touch to your casual wardrobe, this tee combines comfort and style, perfect for daily wear or sporting events. The soft fabric ensures a comfortable fit, while the unique design makes it a standout piece for any outfit.

Choose Size

SmallMediumLargeX-Large

Add to cart

STAY UPTO DATE ABOUT OUR LATEST OFFERS

Enter your email address

Subscribe to Newsletter

Home > Cart

Your Cart



Sleeve Stripe T-Shirt

Size : Small

\$130



+

5

-

Order Summary

SubTotal	\$650.00
Discount(-20%)	-\$130.00
Delivery Fee	\$30
Total	\$520.00

STAY UPTO DATE ABOUT OUR
LATEST OFFERS



Enter your email address

Subscribe to Newsletter