

## Day 3 - API Integration Report - [SHOP.CO]

### ➤ API integration process

1. Install and configure the Sanity client in project.
2. Write a GROQ query to fetch product data, including all necessary fields.
3. Use the `sanity.fetch()` method to retrieve data.
4. Map over the fetched products and render the information dynamically.
5. Add error handling to show fallback UI if the API fails.
6. Create a Typescript interface to ensure consistency.

### ➤ Adjustments made to schema

- **Added Options to `category`:**
  - Defined a dropdown list for predefined categories: T-Shirt, Short, Jeans, Hoodie, Shirt.
- **Enhanced `colors` Field:**
  - Converted `colors` to an array of strings for flexibility in defining multiple color options.
- **Enhanced `sizes` Field:**
  - Converted `sizes` to an array of strings to support multiple size options.
- **Added new Field:**
  - Introduced a `boolean` field to mark whether a product is newly added.
- **Included `discountPercent` Field:**
  - Added a field to define the discount percentage for each product.

## ➤ Migration steps and tools used.

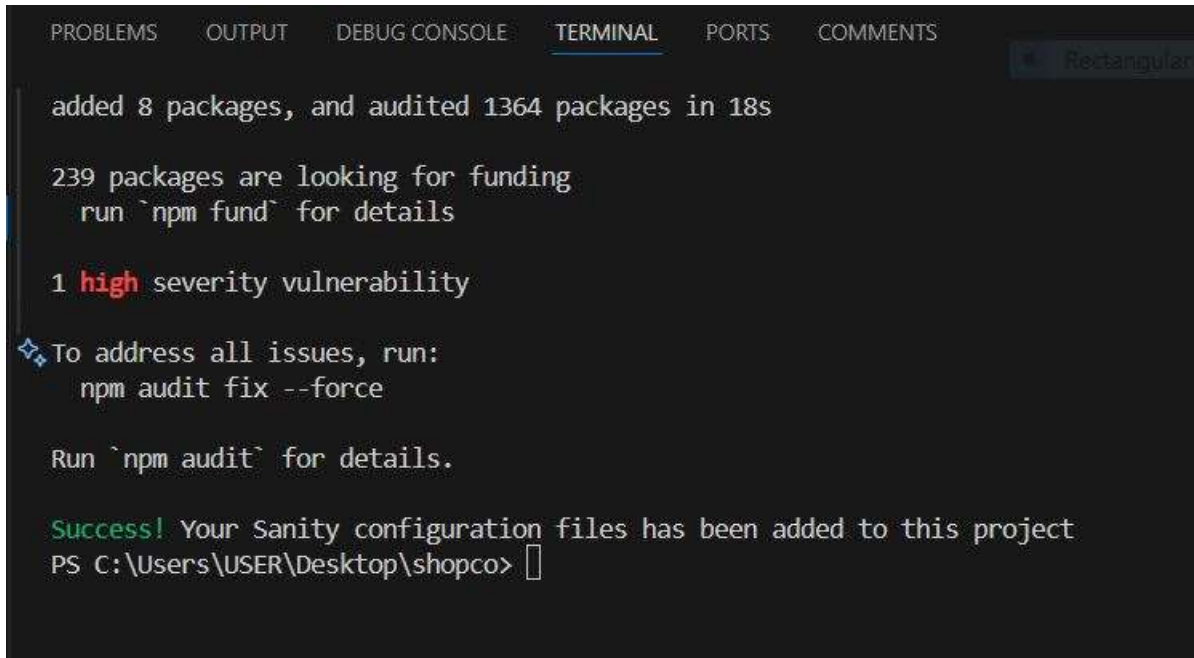
### Migration Steps

- **Schema Update:** Added new fields and enhanced `category` options.
- **Backup:** Exported the existing dataset for safety.
- **Deployment:** Deployed changes using `sanity deploy`.
- **Data Migration:** Used GROQ queries and manual edits to update existing data.
- **Validation:** Verified data integrity and tested API responses.

### Tools Used

- **Sanity Studio:** Schema adjustments and manual edits.
- **Sanity CLI:** For deployment.
- **GROQ:** Querying and updating data.
- **Next.js:** Frontend integration and testing.

# Sanity Installation



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
added 8 packages, and audited 1364 packages in 18s

239 packages are looking for funding
  run `npm fund` for details

1 high severity vulnerability

❖ To address all issues, run:
  npm audit fix --force

Run `npm audit` for details.

Success! Your Sanity configuration files has been added to this project
PS C:\Users\USER\Desktop\shopco> 
```

lib/products.ts



```
1  import { defineType } from "sanity"
2
3  export default defineType({
4    name: 'products',
5    title: 'Products',
6    type: 'document',
7    fields: [
8      {
9        name: 'name',
10       title: 'Name',
11       type: 'string',
12     },
13     {
14       name: 'price',
15       title: 'Price',
16       type: 'number',
17     },
18     {
19       name: 'description',
20       title: 'Description',
21       type: 'text',
22     },
23     {
24       name: 'image',
25       title: 'Image',
26       type: 'image',
27     },
28     {
29       name: "category",
30       title: "Category",
31       type: 'string',
32       options: {
33         list: [
34           {title: 'T-Shirt', value: 'tshirt'},
35           {title: 'Short', value: 'short'},
36           {title: 'Jeans', value: 'jeans'} ,
37           {title: 'Hoddie', value: 'hoodie'} ,
38           {title: 'Shirt', value: 'shirt'} ,
39         ]
40       }
41     },
42     {
43       name: "discountPercent",
44       title: "Discount Percent",
45       type: 'number',
46     },
47   ],
48 })
```

# Token Generation

Getting started

Overview

Members

Studios

Datasets

Access

Activity

Usage

Plan

API

Tokens are used to authenticate apps and scripts to access project data.

Webhooks

CORS origins

Tokens

Name

Examples: "Employee import", "Website preview" or "PDF generator".

SHOP.CO

Permissions

Choose the access privileges for the token.

Contributor

☐

Read and write access to draft content within all datasets, with no access to project settings. (Tokens: read+write drafts)

☐

Deploy Studio (Token only)

Access to deploy Sanity Studio and GraphQL APIs to our hosted service.

☒

Developer

Read and write access to all datasets, with access to project settings for developers. (Tokens: read+write)

☐

Editor

Read and write access to all datasets, with limited access to project settings. (Tokens: read+write)

☐

Viewer

Read access to all datasets, with limited access to project settings. (Tokens: read-only)

Save

Cancel

## Tokens

+ Add API token

Tokens are used to authenticate apps and scripts to access project data.

NAME	PERMISSIONS	CREATED	
SHOP.CO	Developer	0 seconds	

Copy the token below – this is your only chance to do so!

skJcvPKxSOxme7BnyXmR3H0HScJgLRs4VOR3l3npSiVXlSt7o0TnSH4y8BuUcEOXCAULWkH5ZsLH7oq4Nms674  
DVMyNJa3IpYS70h0xrpY17pSE7nSqXuYTUftRRaV0fAjDXC50L0nutZkCKZ8zr5tuYBtKyp1KUdRaqE9zDjj4q  
zJVVqZmD

# .env.local

```
$ .env.local
1 NEXT_PUBLIC_SANITY_PROJECT_ID="abxbskbb"
2 NEXT_PUBLIC_SANITY_DATASET="production"
3 SANITY_API_TOKEN="sk2dA7K1BSQwQw2TYVjW7BUHqFOR0b0xa1ZFQ6b51oSgbFSECyCSaM4Jt4qbCAq7PVIWe8xFqjQTP6Kq93s0S90kkeonf28AbiMzwAFRSrg8blyHcSyh107kGn3d3Hk0yR9zHVOaxJf9nK0c0G79BA9gryrHseRyOMWztpizWC1eViE25No"
```

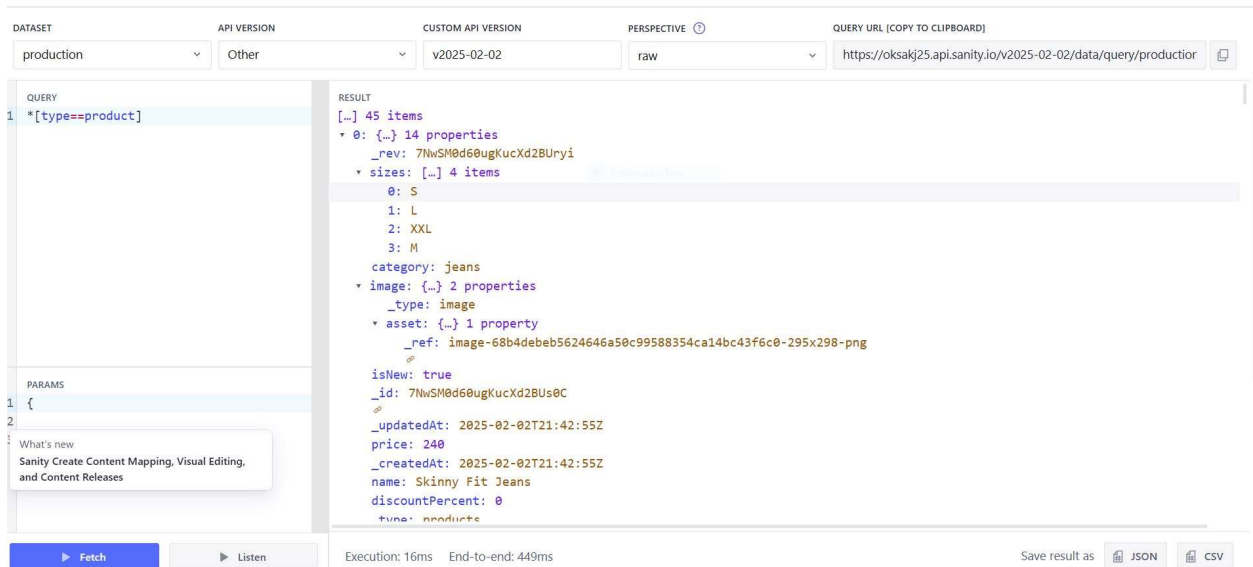
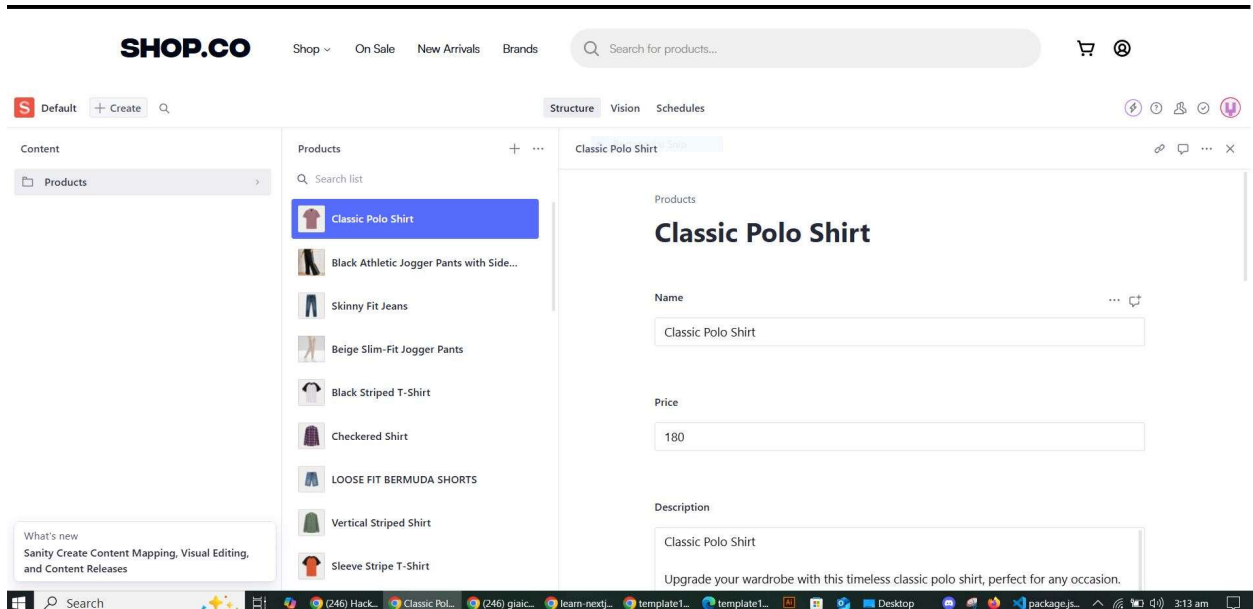
# Package.json

The screenshot shows the VS Code interface. In the Explorer sidebar on the left, the file `package.json` is selected. The main editor displays the content of `package.json` with line numbers 1 through 18. The file is a Next.js project configuration.

```
1  {} package.json > {} scripts
2  {
3    "name": "yt",
4    "version": "0.1.0",
5    "private": true,
6    "type": "module",
7    "scripts": {
8      "dev": "next dev",
9      "build": "next build",
10     "start": "next start",
11     "lint": "next lint",
12     "import-data": "node script/importData.js"
13   },
14   "dependencies": {
15     "@radix-ui/react-accordion": "^1.2.2",
16     "@radix-ui/react-checkbox": "^1.1.3",
17     "@radix-ui/react-dialog": "^1.1.2",
18     "@radix-ui/react-dropdown-menu": "^2.1.4",
19     "@radix-ui/react-label": "^2.1.0",
```

# Script/importData.js

```
1 import { createClient } from '@sanity/client';
2
3 const client = createClient({
4   projectId: 'abxbskbb',
5   dataset: 'production',
6   useCdn: true,
7   apiVersion: '2025-01-13',
8   token: 'sk2da7k1B5QwQz2TVJjWjBUHqFOR0b0xalZFQ6b51oSgbF5ECyCSaN47l4qbCAq7PVlWe8xfJQTP6kq93s059Dkkeeonf28AbIMzwaAFRsg8blyHc5yh1O7kGn3d3HkOyR9zW0ax3f9nkOC0G79BA9gryrHseRyOMwextpizkCieVie25No',
9 });
10
11 async function uploadImageToSanity(imageUrl) {
12   try {
13     console.log('Uploading image: ${imageUrl}');
14
15     const response = await fetch(imageUrl);
16     if (!response.ok) {
17       throw new Error('Failed to fetch image: ${imageUrl}');
18     }
19
20     const buffer = await response.arrayBuffer();
21     const bufferImage = Buffer.from(buffer);
22
23     const asset = await client.assets.upload('image', bufferImage, {
24       filename: imageUrl.split('/').pop(),
25     });
26
27     console.log('Image uploaded successfully: ${asset._id}');
28     return asset._id;
29   } catch (error) {
30     console.error('Failed to upload image:', imageUrl, error);
31     return null;
32   }
33 }
34
35 async function uploadProduct(product) {
36   try {
37     const imageId = await uploadImageToSanity(product.imageUrl);
38
39     if (imageId) {
40       const document = {
41         _type: 'products',
42         name: product.name,
43         description: product.description,
44         price: product.price,
45         image: {
46           _type: 'image',
47           assets: [
48             {
49               _ref: imageId,
50             },
51           ],
52           category: product.category,
53           discountPercent: product.discountPercent,
54           isNew: product.isNew,
55           colors: product.colors,
56           sizes: product.sizes
57         };
58
59       const createdProduct = await client.create(document);
60       console.log('Product ${product.name} uploaded successfully:', createdProduct);
61     } else {
62       console.log('Product ${product.name} skipped due to image upload failure.');
```





[illegible]

Products From API's Data



New

Casual Green Bomber Jacket

Category: *hoodie*

This stylish green bomber jacket offers a sleek and modern twist on a classic design. Made from soft...

Available Colors:



Sizes:



\$300

20% OFF

Add to Cart



Classic White Pullover Hoodie

Category: *hoodie*

This white pullover hoodie offers a clean, minimalist look with its soft fabric and timeless design....

Available Colors:



Sizes:



\$150

10% OFF

Add to Cart



LOOSE FIT BERMUDA SHORTS

Category: *short*

These Loose Fit Bermuda Shorts are designed for comfort and laid-back style. Made from lightweight, ...

Available Colors:



Sizes:



\$78

20% OFF

Add to Cart



New

Skinny Fit Jeans

Category: *jeans*

Upgrade your everyday wardrobe with these classic straight-fit denim jeans. Designed for comfort and...

Available Colors:



Sizes:



\$240

Add to Cart