

DAY:3

Sanity Migration API Documentation

Introduction

Sanity provides a powerful content management system (CMS), and sometimes we need to migrate content from one structure to another or from one environment to another. This guide explains how to use Sanity's migration API to handle these tasks.

Table of Contents

1. [What is Sanity Migration API?](#)
 2. [Use Cases](#)
 3. [Setting Up Migration API](#)
 4. [Writing Migration Scripts](#)
 5. [Screenshots and Examples](#)
 6. [Conclusion](#)
-

1. What is Sanity Migration API?

Sanity migration API is designed to help developers automatically migrate content and schema changes between different environments or versions of Sanity's content structure.

It can be used for:

- Updating document fields
- Renaming or restructuring schemas
- Moving data between different datasets
- And more...

2. Use Cases

Here are some scenarios where you might want to use the migration API:

- **Schema Changes:** If you've updated your content schema and need to apply these changes to the existing content in the Sanity database.
 - **Data Updates:** When you need to modify existing data, such as changing a field type or renaming a field across all documents.
 - **Environment Synchronization:** Moving content between different Sanity environments, like staging to production.
-

3. Setting Up Migration API

4. Initialize the project and set up your Sanity project.

5. Install the required dependencies.

6. Set up environment variables in your .env file:

```
TS products U X  tsconfig.json  JS importSanityData.mjs U  {} package.json M  JS index.js  TS env.ts U  $ .env.local X  TS index.t
$ .env.local
1 NEXT_PUBLIC_SANITY_PROJECT_ID="1177dy5p"
2 NEXT_PUBLIC_SANITY_DATASET="production"
3 SANITY_API_TOKEN="skjbw01lu7ZkIgsYr7NHDGcn40U62Ehn5prjDLp1BB3MuRuq0A6JauFYHUNjkt0aCjSiVnYhV8wXsF40gArdecJL9sbUGR19F7UYZNAV
  BwiQJ1yXyN61ZQi0hUK2gYe9BnojovyTU3KD8Kkpe7Cr47cQnAOxGAnnVd0hVBiqBr3tBqsG7EpV"
```



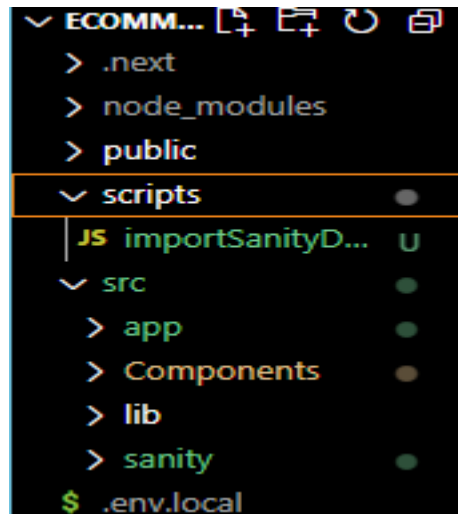
Creating the Sanity Schema

```
src > sanity > schemaTypes > TS product.ts > [0] default > fields
1 export default {
2   name: 'product',
3   type: 'document',
4   title: 'Product',
5   fields: [
6     {
7       name: 'name',
8       type: 'string',
9       title: 'Name',
10      validation: (Rule: any) => Rule.required().error('Name is required'),
11    },
12    {
13      name: 'image',
14      type: 'image',
15      title: 'Image',
16      options: {
17        hotspot: true,
18      },
19      description: 'Upload an image of the product.',
20    },
21    {
22      name: 'price',
23      type: 'string',
24      title: 'Price',
25      validation: (Rule: any) => Rule.required().error('Price is required'),
26    },
27    {
28      name: 'description',
29      type: 'text',
30      title: 'Description',
31      validation: (Rule: any) =>
32        | Rule.max(150).warning('Keep the description under 150 characters.'),
33    },
34  ],
35 }
```

Then, update your `sanity/schemaTypes/index.ts` file to include the new product schema:

```
1 import { type SchemaTypeDefinition } from 'sanity'
2 import product from '@sanity/schemaTypes/product'
3 export const schema: { types: SchemaTypeDefinition[] } = {
4   types: [product],
5 }
6
```

Setting Up the Data Import Script



Now, let's create a script to import data from an external API into Sanity. Create a new file `scripts/importSanityData.mjs` in your project root:

```
scripts > JS importSanityData.mjs > client > dataset
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 const __filename = fileURLToPath(import.meta.url);
8 const __dirname = path.dirname(__filename);
9 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
10
11 const client = createClient({
12   projectId: "1177dySp",
13   dataset: "production",
14   token: process.env.SANITY_API_TOKEN,
15   apiVersion: '2025-01-15',
16   useCdn: false,
17 });
18
19 async function uploadImageToSanity(imageUrl) {
20   try {
21     console.log(`Uploading Image : ${imageUrl}`);
22     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
23     const buffer = Buffer.from(response.data);
24     const asset = await client.assets.upload('image', buffer, {
25       filename: imageUrl.split('/').pop(),
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   }
}
```

Now, let's install the necessary packages. Run the following command in your terminal:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

me down\UIUX\ecommerce-sofa> npm install @sanity/client axios dotenv

up to date, audited 1313 packages in 13s

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

1 moderate severity vulnerability

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

Running the Import Script

```
1  {
2    "name": "ecommerce-sofa",
3    "version": "0.1.0",
4    "private": true,
5    > Debug
6    "scripts": {
7      "dev": "next dev",
8      "build": "next build",
9      "start": "next start",
10     "lint": "next lint",
11     "import-data": "node scripts/importSanityData.mjs"
12   },
13   "dependencies": {
14     "@sanity/client": "^6.25.0",
```

Now you can run the import script using:

`npm run import-data`

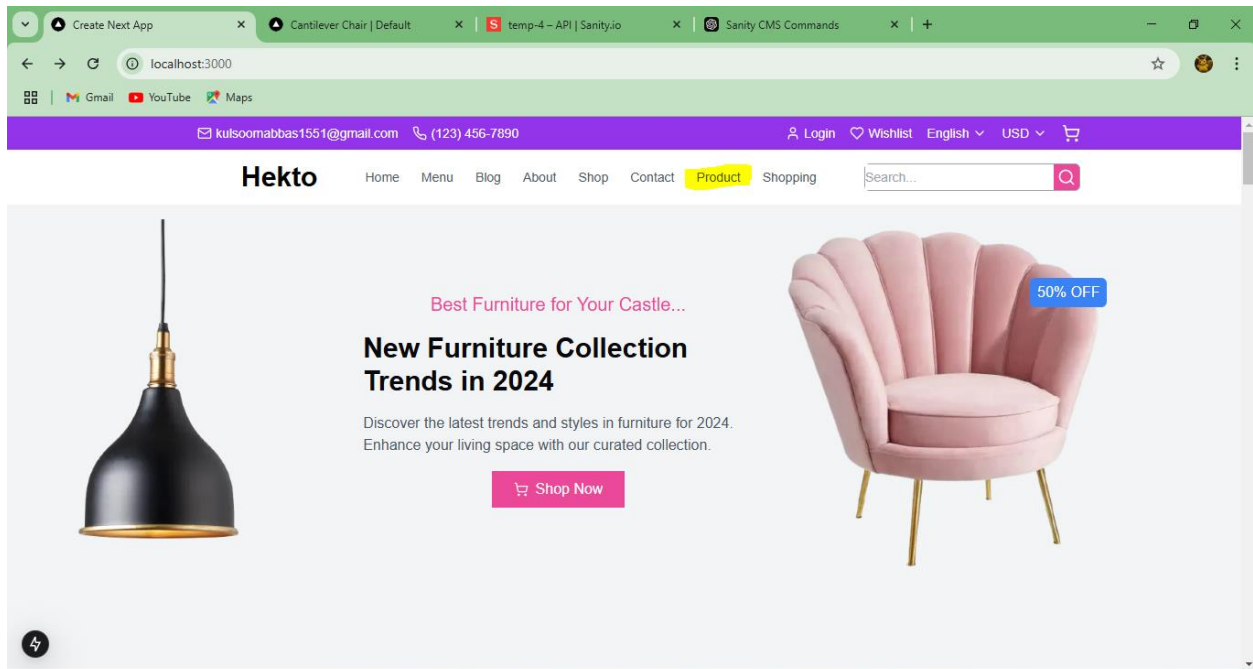
```
> ecommerce-sofa@0.1.0 import-data
> node scripts/importSanityData.mjs

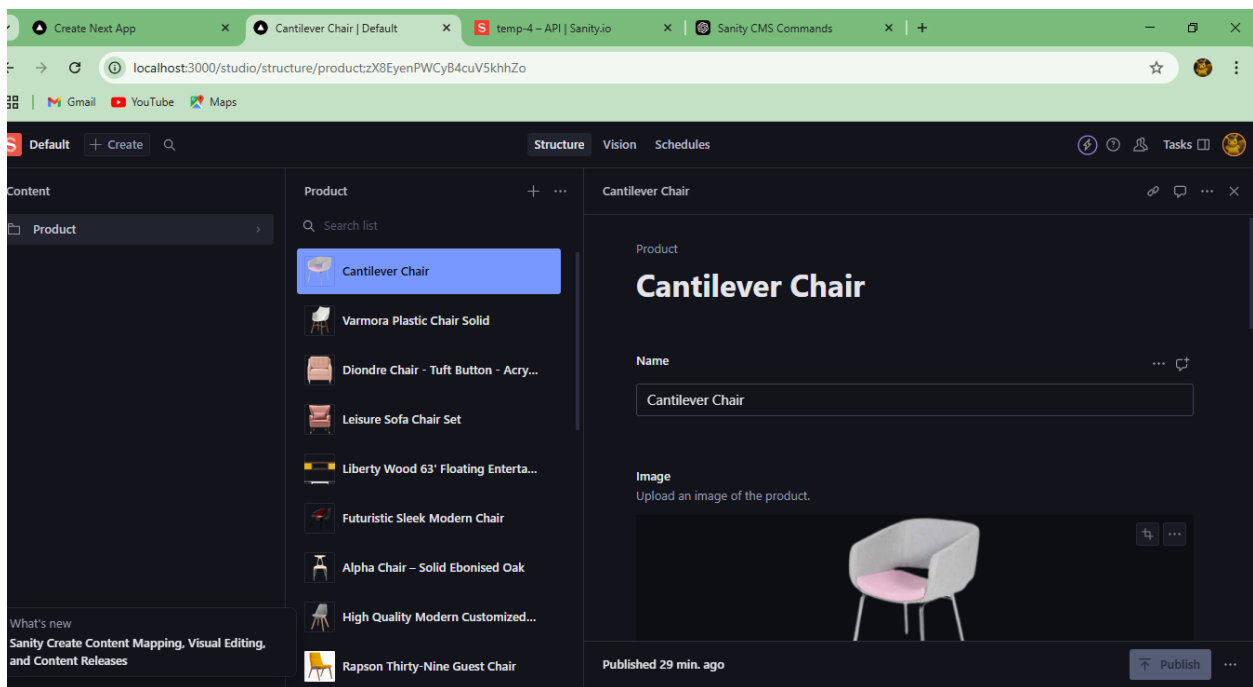
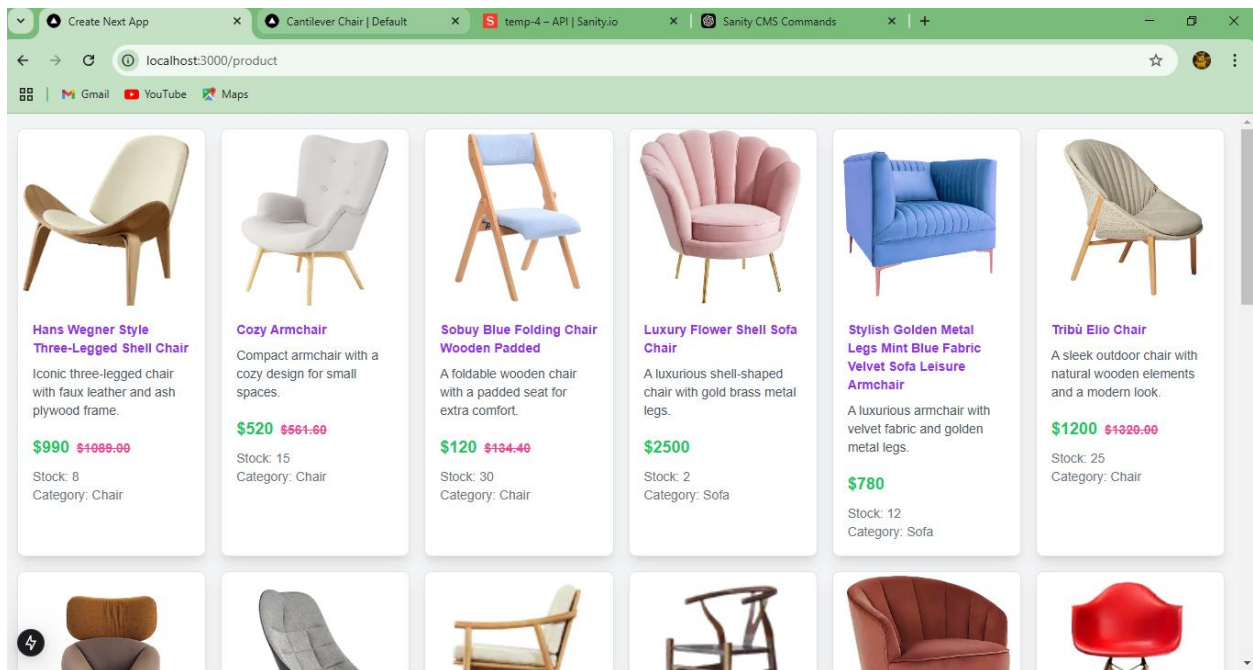
Fetching Product Data From API ...
Processing Item: Tribù Elio Chair
Uploading Image : https://next-ecommerce-template-4.vercel.app/product/Chair (5).png
Image Uploaded Successfully : image-9967004db2d5e1235db7525ed72aaa32bc4f95a7-720x522-png
Uploading Chair - Tribù Elio Chair to Sanity !
Uploaded Successfully: D24v8QimRkTY37EjyrbHv
-----

Processing Item: Armchair Tortuga
Uploading Image : https://next-ecommerce-template-4.vercel.app/product/Chair (35).png
Image Uploaded Successfully : image-d6aed5e3bb1c45542716cb31036c740f1113625e-1398x1194-png
Uploading Chair - Armchair Tortuga to Sanity !
Uploaded Successfully: D24v8QimRkTY37Ejyrb08
-----

Processing Item: Uchiwa Quilted Lounge Chair
Uploading Image : https://next-ecommerce-template-4.vercel.app/product/Chair (6).png
Image Uploaded Successfully : image-2e1f600d1709908be26d3768263fa9b3d7fe4ead-402x402-png
Uploading Chair - Uchiwa Quilted Lounge Chair to Sanity !
Uploaded Successfully: NwaIKVf0z91aTr5XfQ0Y0h
-----

Processing Item: Stylish Golden Metal Legs Mint Blue Fabric Velvet Sofa Leisure Armchair
Uploading Image : https://next-ecommerce-template-4.vercel.app/product/Chair (3).png
Image Uploaded Successfully : image-7a9f255386e1ee47303b1a578d6b7d43829b469b-1018x1100-png
Uploading Sofa - Stylish Golden Metal Legs Mint Blue Fabric Velvet Sofa Leisure Armchair to Sanity !
Uploaded Successfully: NwaIKVf0z91aTr5XfQ0Yk8
-----
```





9. Conclusion

Sanity's migration API is an invaluable tool when dealing with schema changes, data updates, or syncing between environments. By following this guide, you should be able to write, run, and monitor migration scripts for your Sanity dataset with ease.

For more information, refer to the Sanity Documentation.