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.

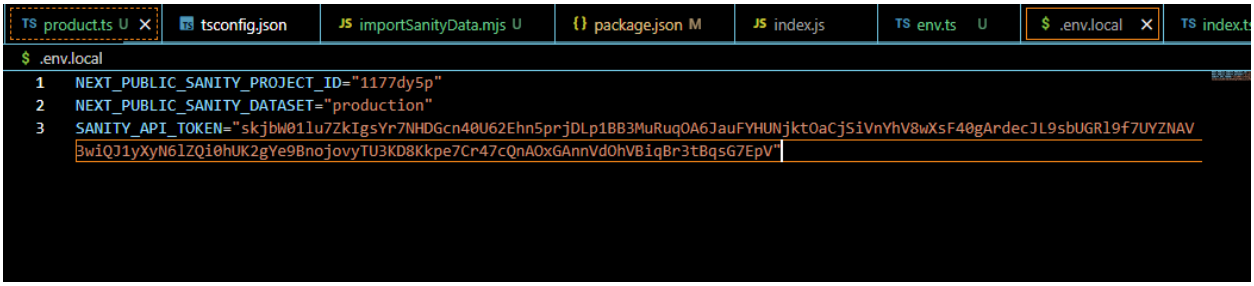
---

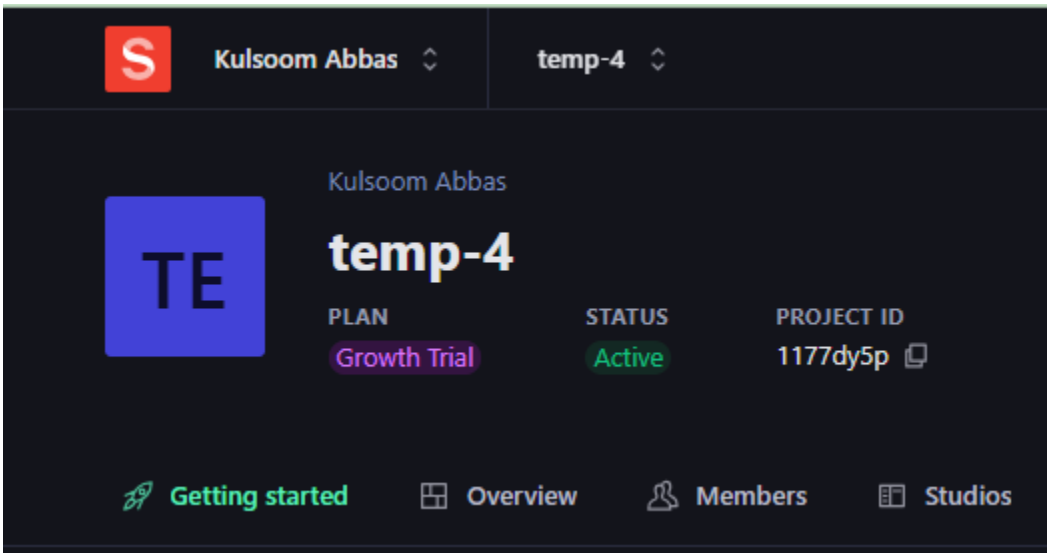
3. Setting Up Migration API

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

5. Install the required dependencies.

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





Creating the Sanity Schema

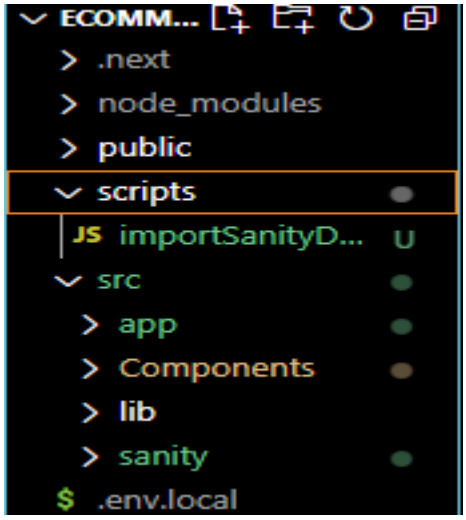
src > sanity > schemaTypes > TS product.ts > [@] default > fields

```
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Name',
      validation: (Rule: any) => Rule.required().error('Name is required'),
    },
    {
      name: 'image',
      type: 'image',
      title: 'Image',
      options: {
        hotspot: true,
      },
      description: 'Upload an image of the product.',
    },
    {
      name: 'price',
      type: 'string',
      title: 'Price',
      validation: (Rule: any) => Rule.required().error('Price is required'),
    },
    {
      name: 'description',
      type: 'text',
      title: 'Description',
      validation: (Rule: any) =>
        Rule.max(150).warning('Keep the description under 150 characters.'),
```

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

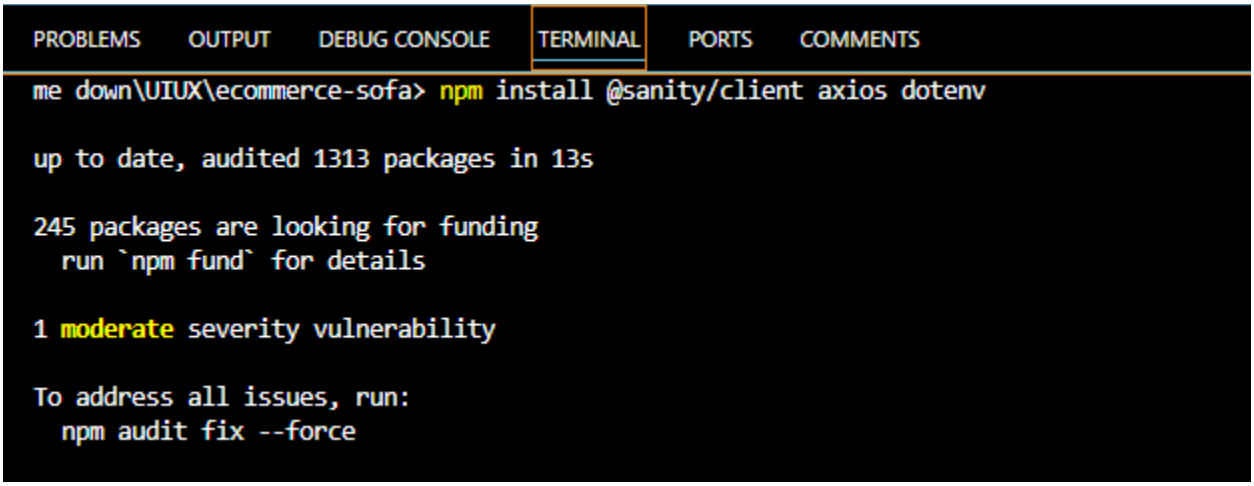
```
import { type SchemaTypeDefinition } from 'sanity'
import product from '@/sanity/schemaTypes/product'
export const schema: { types: SchemaTypeDefinition[] } = {
  types: [product],
}

```

Setting Up the Data Import Script

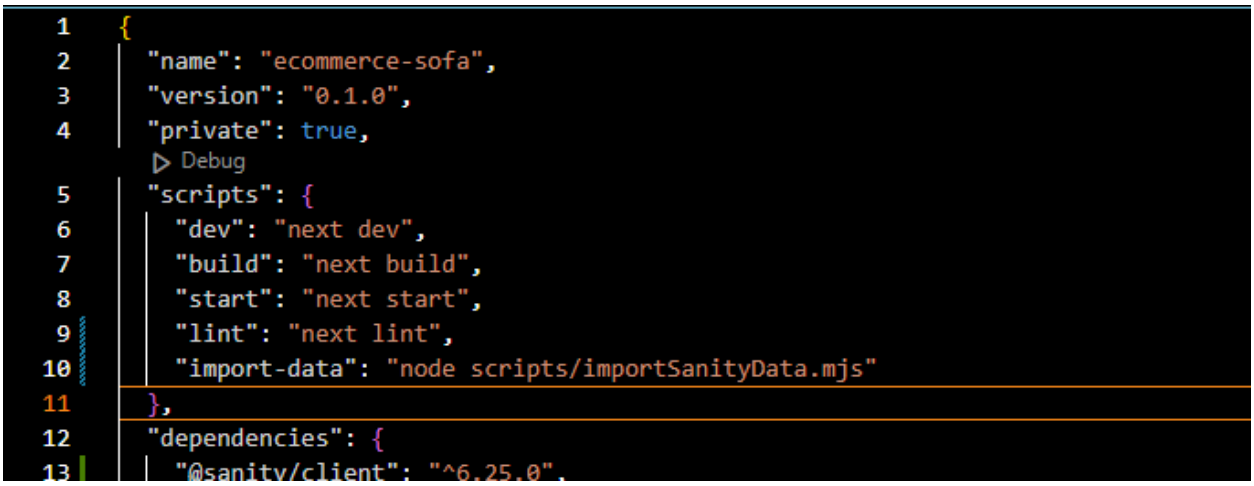


Now, let's create a script to import data from an external API into Sanity. Create a new file `scripts/importSanityData.mjs` in your project root:

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

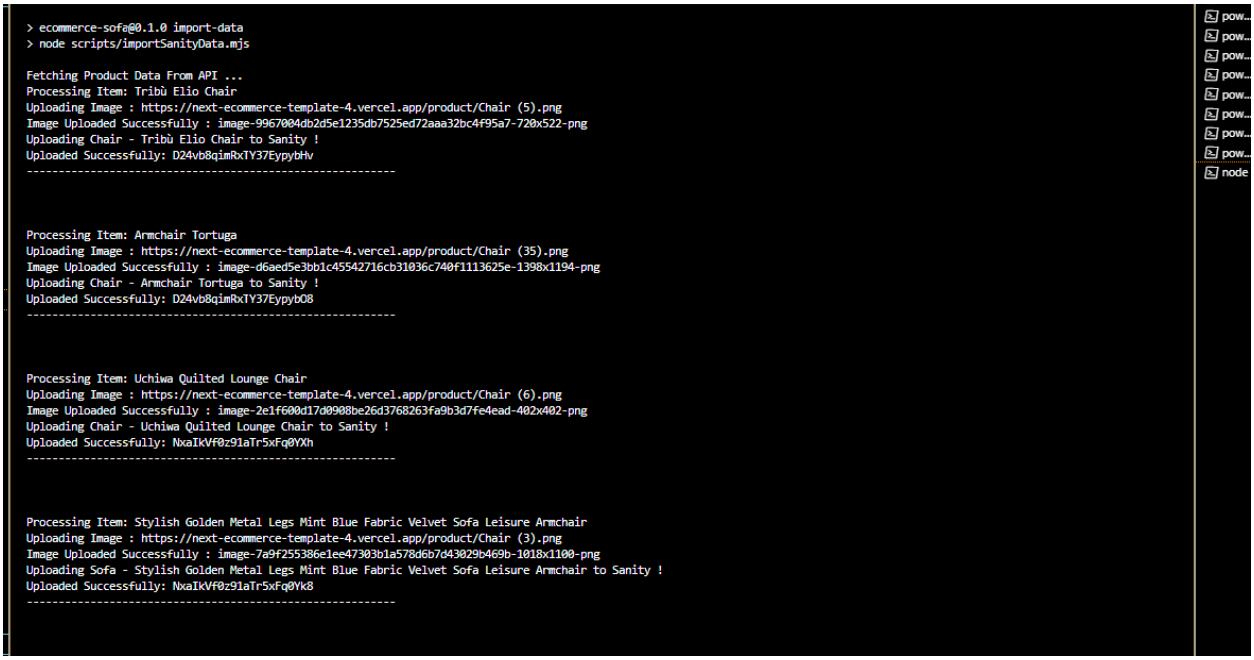


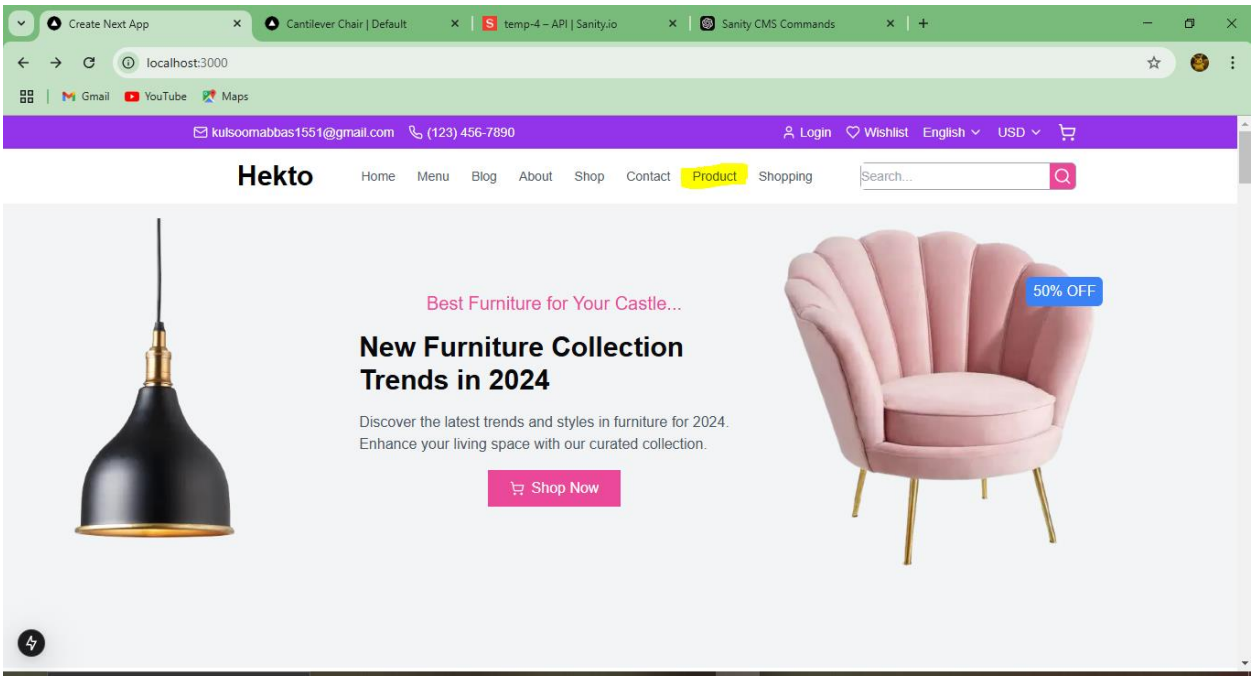
Running the Import Script

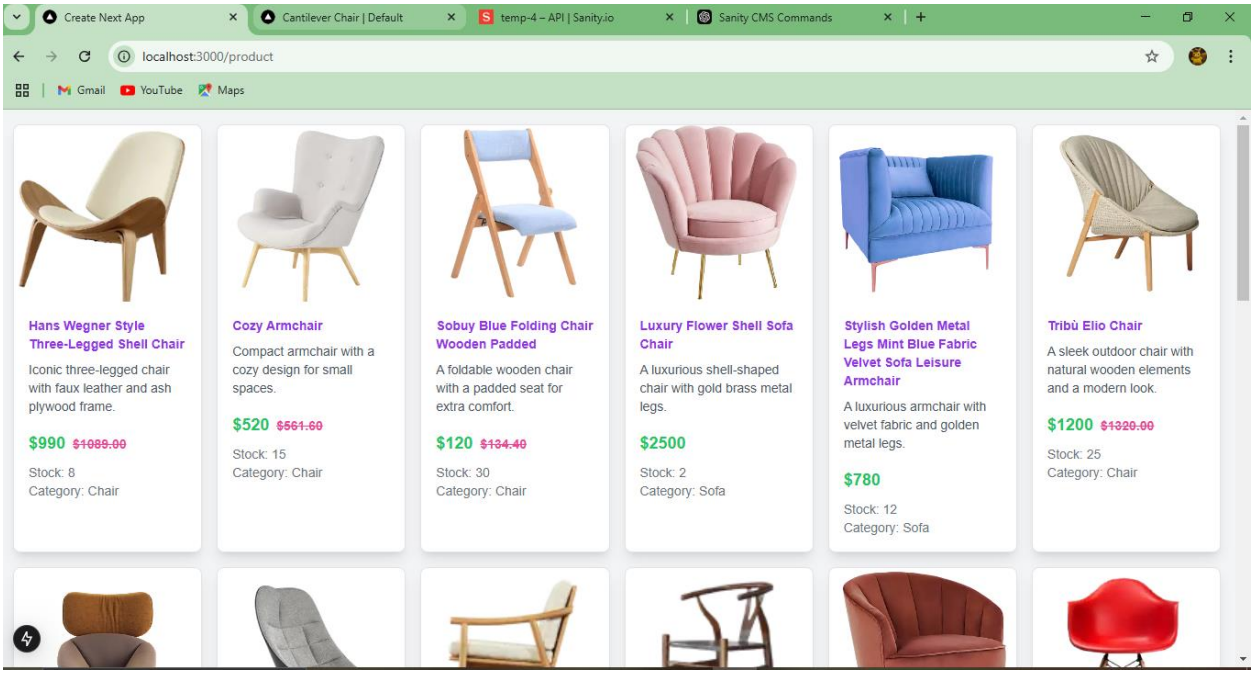


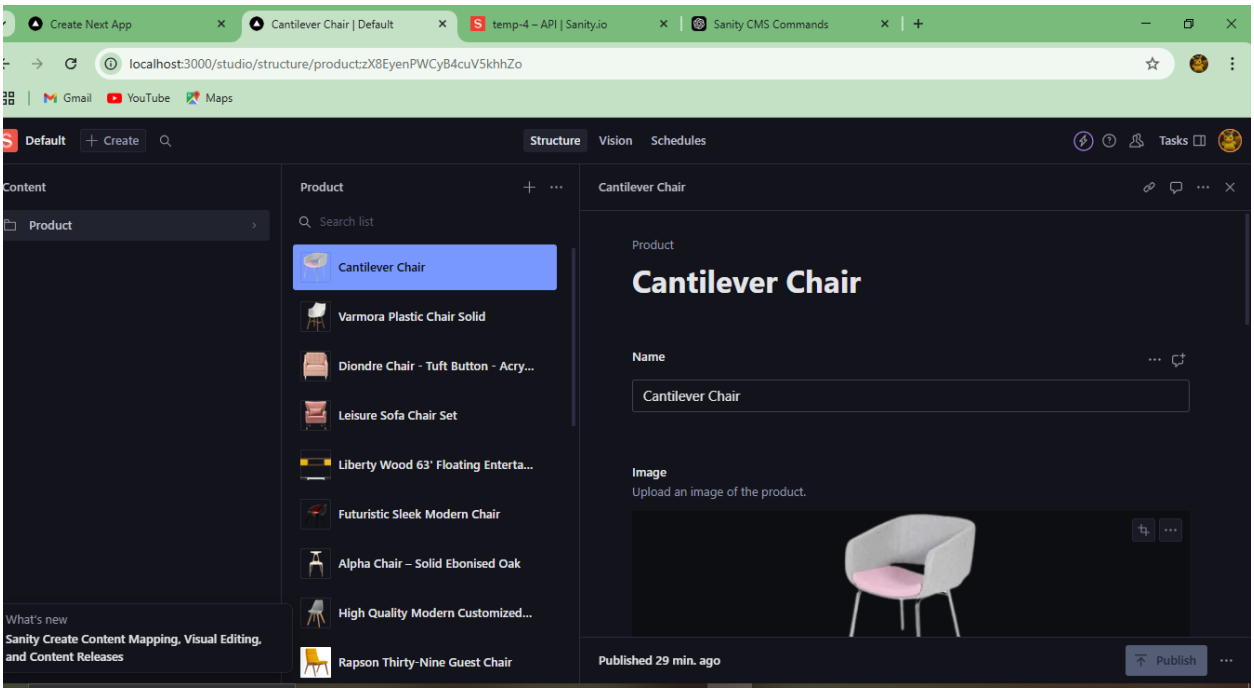
Now you can run the import script using:

npm run import-data









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.