[Day\_3\_API\_Integration\_and\_Data\_Migration](https://github.com/panaverse/learn-nextjs/blob/main/HACKATHONS/Marketplace_Builder_Hackathon_2025/Day_3_API_Integration_and_Data_Migration.pdf" \o "Day_3_API_Integration_and_Data_Migration.pdf):

General E-Commerce Hekto Furniture website

I have faced many types of error with the given api templete and documentation

Then I have follow these steps and then finally I have resolved my issues

**Table of Contents**

1. Setting Up Environment Variables  
2. Obtaining Sanity Project ID and API Token  
3. Creating the Sanity Schema  
4. Setting Up the Data Import Script  
5. Running the Import Script

* **Configuring Environment Variables:**  
  Begin by setting up your environment variables. If a .env.local file doesn't already exist in your project's root directory, create one. Then, add the following variables:

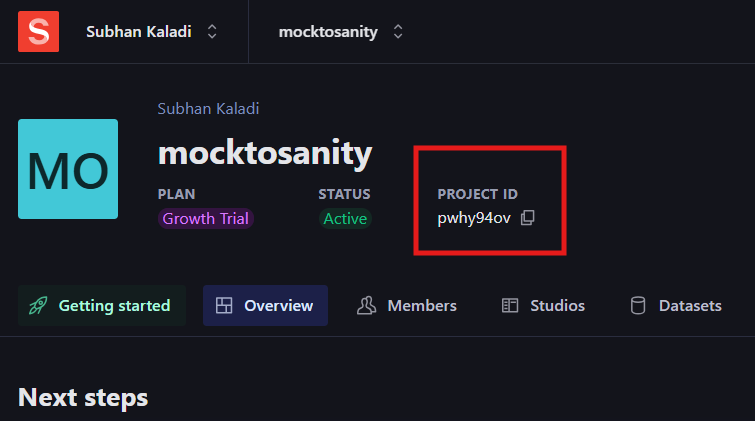
NEXT\_PUBLIC\_SANITY\_PROJECT\_ID=your\_project\_id  
 NEXT\_PUBLIC\_SANITY\_DATASET=production  
 SANITY\_API\_TOKEN=your\_sanity\_token

*Remember, variables prefixed with NEXT\_PUBLIC\_ will be exposed to the browser, so be cautious about what you prefix.*

* **Fetching Sanity Project ID and API Token**Project ID

**To find your Sanity project ID**:

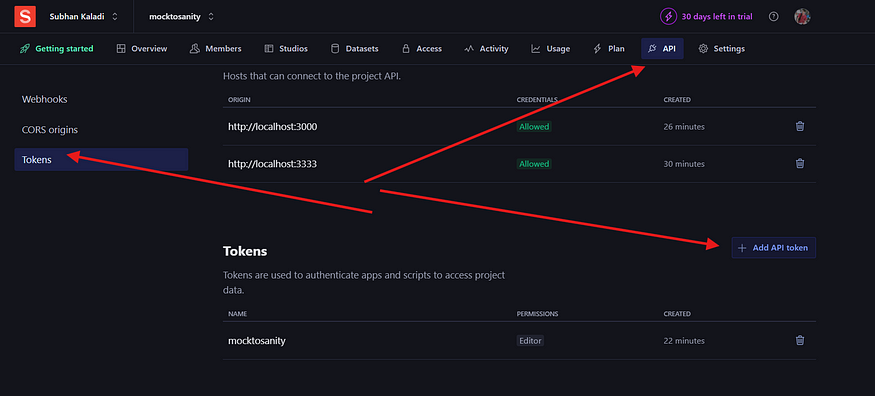
* Log in to your Sanity account at <https://www.sanity.io/manage>
* Select your project
* In the project dashboard, you’ll see the project ID listed

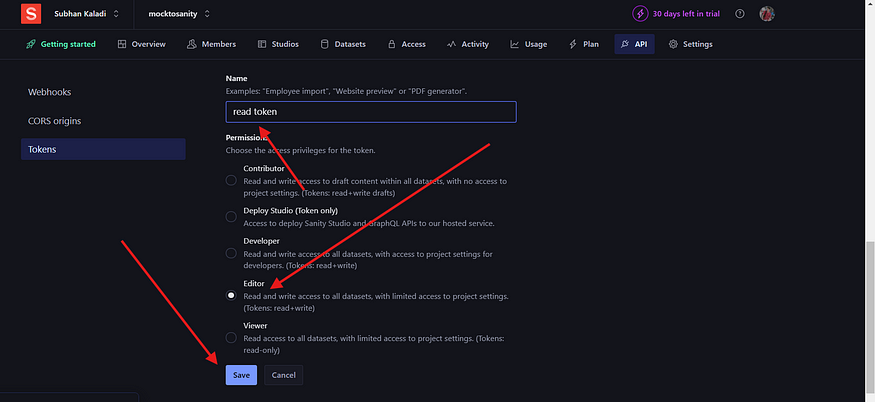


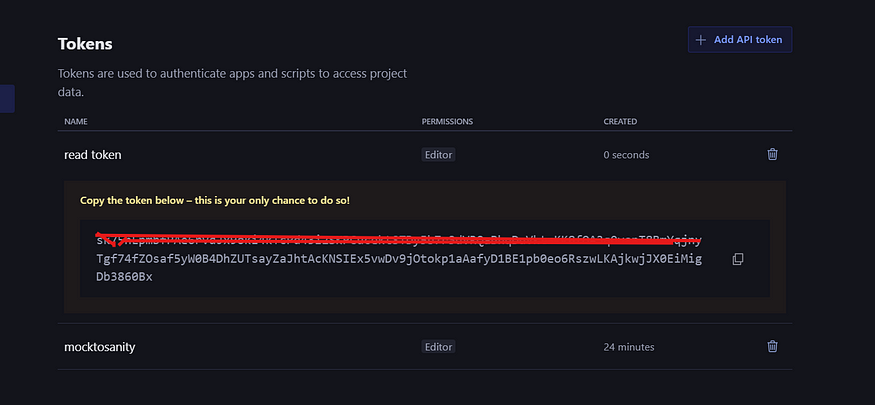
Use this ID for the `NEXT\_PUBLIC\_SANITY\_PROJECT\_ID` in your `.env.local` file.

**API Token**:  
To generate a Sanity API token:

1. Go to <https://www.sanity.io/manage> and select your project  
2. Navigate to the “API" tab  
3. Under "Tokens," click "Add API token"  
4. Give your token a name and select the appropriate permissions (usually "Editor" for full read/write access)  
5. Copy the generated token







Use this token for the `SANITY\_API\_TOKEN` in your `.env.local` file.

**3. Creating the Sanity Schema**

Now, let’s create a schema for our products. In your Sanity schema folder (usually `sanity/schemaTypes`), create a new file called `product.ts`:

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.'),

},

{

name: 'discountPercentage',

type: 'number',

title: 'Discount Percentage',

validation: (Rule: any) =>

Rule.min(0).max(100).warning('Discount must be between 0 and 100.'),

},

{

name: 'isFeaturedProduct',

type: 'boolean',

title: 'Is Featured Product',

},

{

name: 'stockLevel',

type: 'number',

title: 'Stock Level',

validation: (Rule: any) => Rule.min(0).error('Stock level must be a positive number.'),

},

{

name: 'category',

type: 'string',

title: 'Category',

options: {

list: [

{ title: 'Chair', value: 'Chair' },

{ title: 'Sofa', value: 'Sofa' },

],

},

validation: (Rule: any) => Rule.required().error('Category is required'),

},

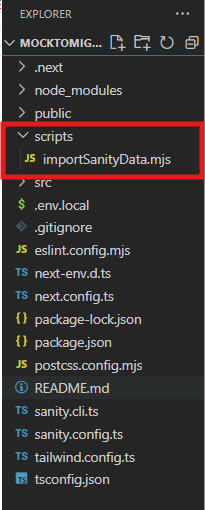
],

};

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

import { type SchemaTypeDefinition } from 'sanity'  
import product from './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/import-data.mjs` in your project root:

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') });

const client = createClient({

projectId: process.env.NEXT\_PUBLIC\_SANITY\_PROJECT\_ID,

dataset: process.env.NEXT\_PUBLIC\_SANITY\_DATASET,

token: process.env.SANITY\_API\_TOKEN,

apiVersion: '2025-01-15',

useCdn: false,

});

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);

return null;

}

}

async function importData() {

try {

console.log('Fetching Product Data From API ...');

const response = await axios.get("https://next-ecommerce-template-4.vercel.app/api/product")

const products = response.data.products;

for (const item of products) {

console.log(`Processing Item: ${item.name}`);

let imageRef = null;

if (item.imagePath) {

imageRef = await uploadImageToSanity(item.imagePath);

}

const sanityItem = {

\_type: 'product',

name: item.name,

category: item.category || null,

price: item.price,

description: item.description || '',

discountPercentage: item.discountPercentage || 0,

stockLevel: item.stockLevel || 0,

isFeaturedProduct: item.isFeaturedProduct,

image: imageRef

? {

\_type: 'image',

asset: {

\_type: 'reference',

\_ref: imageRef,

},

}

: undefined,

};

console.log(`Uploading ${sanityItem.category} - ${sanityItem.name} to Sanity !`);

const result = await client.create(sanityItem);

console.log(`Uploaded Successfully: ${result.\_id}`);

console.log("----------------------------------------------------------")

console.log("\n\n")

}

console.log('Data Import Completed Successfully !');

} catch (error) {

console.error('Error Importing Data : ', error);

}

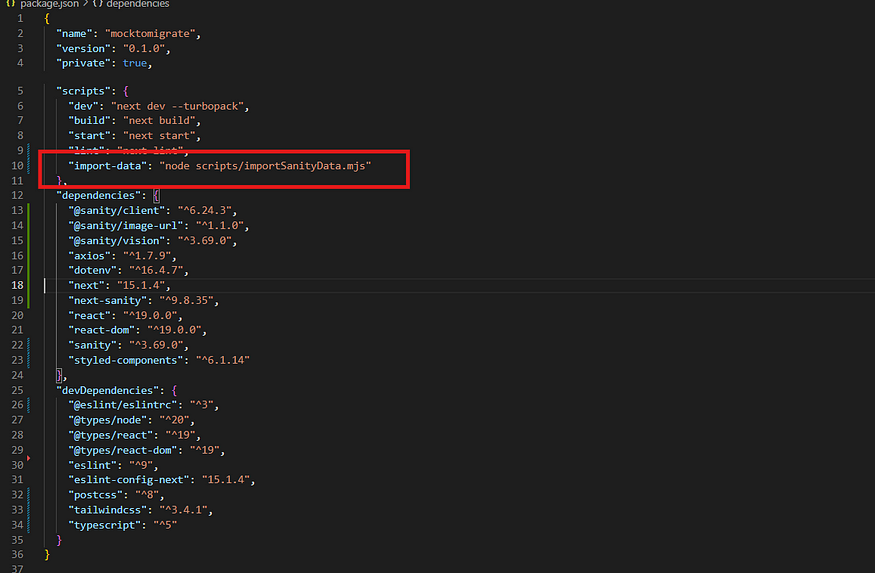
}

importData();

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

**npm install @sanity/client axios dotenv**

**Running the Import Script**



To run the import script, we need to add a new script to our `package.json` file. Open your `package.json` and add the following to the `"scripts"` section:

"scripts": {  
 "dev": "next dev --turbopack",  
 "build": "next build",  
 "start": "next start",  
 "lint": "next lint",  
 "import-data": "node scripts/import-data.mjs"  
}

Now you can run the import script using:

**npm run import-data**

This script will fetch products from the FakeStoreAPI, upload any associated images to Sanity’s asset store, and then create new product documents in your Sanity dataset.

