DAY 3 - API INTEGRATION AND DATA MIGRATION

Steps for Day 3

Installation of Sanity in Next.js:

Step 1: Create a new Next.js project

```
npx create-next-app .
```

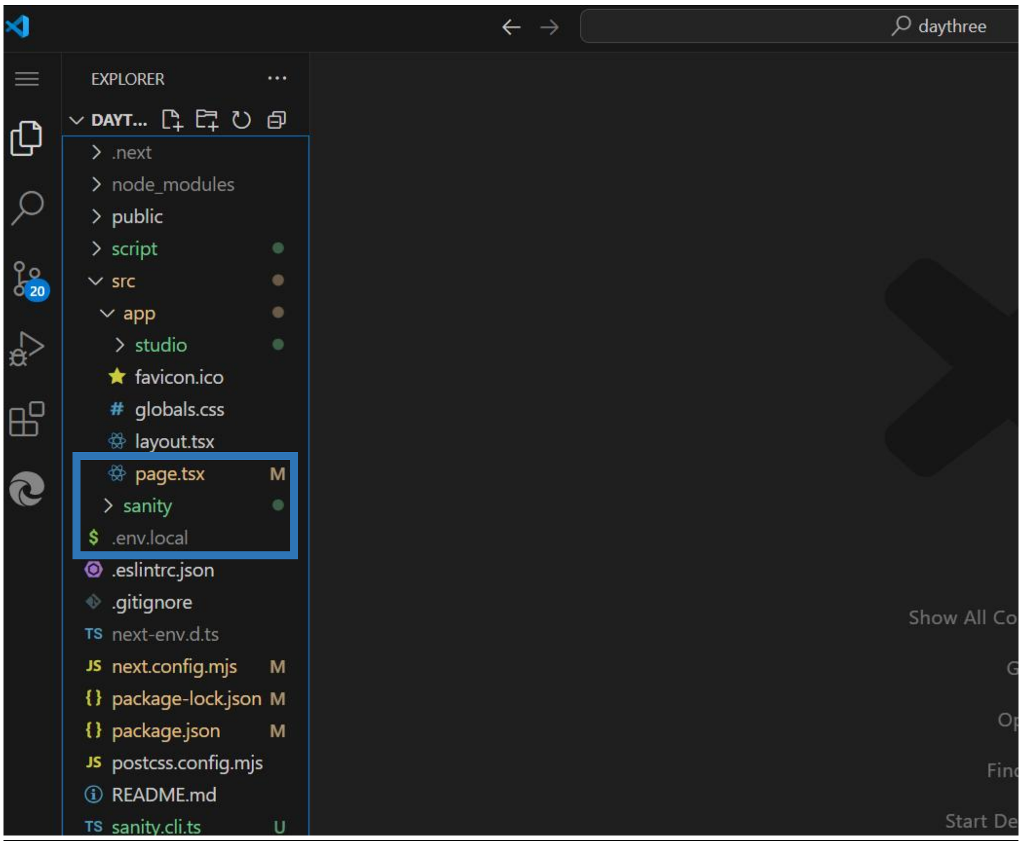
Step 2: Install Sanity Studio

You start by setting up your content editing environment. It's called Sanity Studio, and you can configure and customize it with JavaScript. It runs in the browser. To develop locally, we need to run a development server so you can see your changes instantly.

To get started, run this in your command line:

```
npm create sanity@latest -- --template clean --create-project "learning-sanity-project" --dataset production
```

Check Sanity Installed in your project(if yes)



:

Setting up Environment Variables:

First, let's set up our environment variables. Create a `` `.env.local` `` file in your project root if it doesn't already exist. Add the following variables:

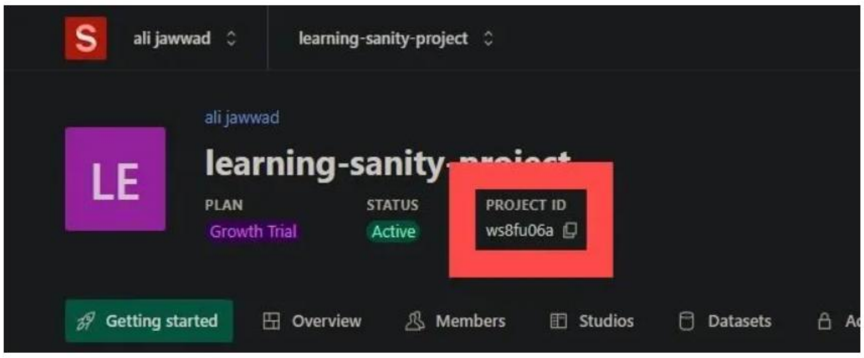
```
NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id
NEXT_PUBLIC_SANITY_DATASET=production
SANITY_API_TOKEN=your_sanity_token
```

Obtaining Sanity Project ID and API Token:

Project ID

To find your Sanity project ID:

1. Log in to your Sanity account at https://www.sanity.io/manage
2. Select your project
3. In the project dashboard, you'll see the project ID listed



ali jawwad

learning-sanity-project

PLAN Growth Trial

STATUS Active

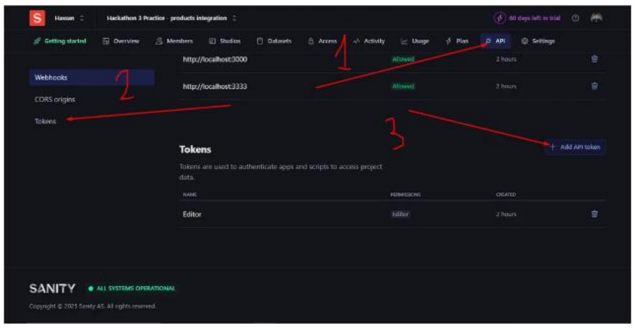
PROJECT ID ws8fu06a

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



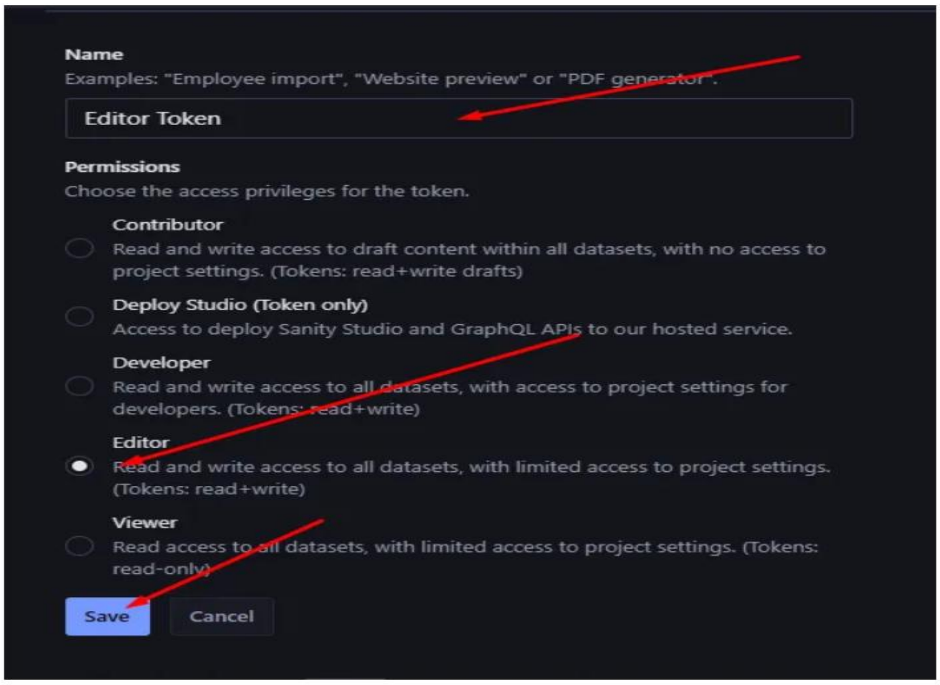
http://localhost:3000	Allowed	2 hours	
http://localhost:3333	Allowed	2 hours	

Tokens

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

NAME	PERMISSIONS	CREATED
Editor	Editor	2 hours

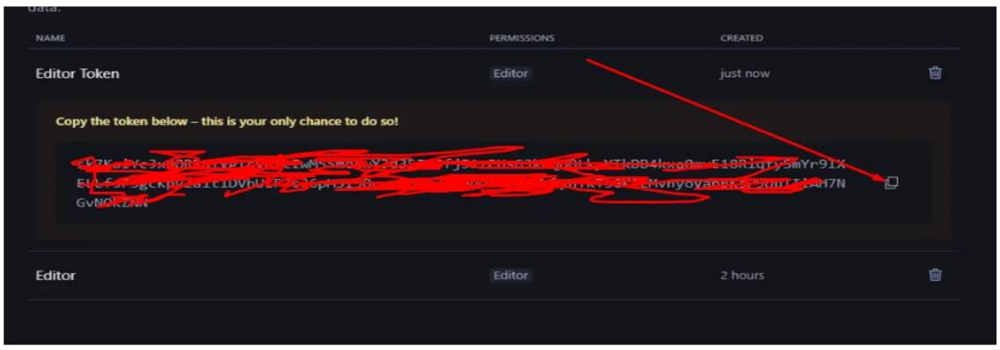
Copyright © 2025 Sanity AS. All rights reserved.



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

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)



data.

NAME	PERMISSIONS	CREATED
Editor Token	Editor	just now

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

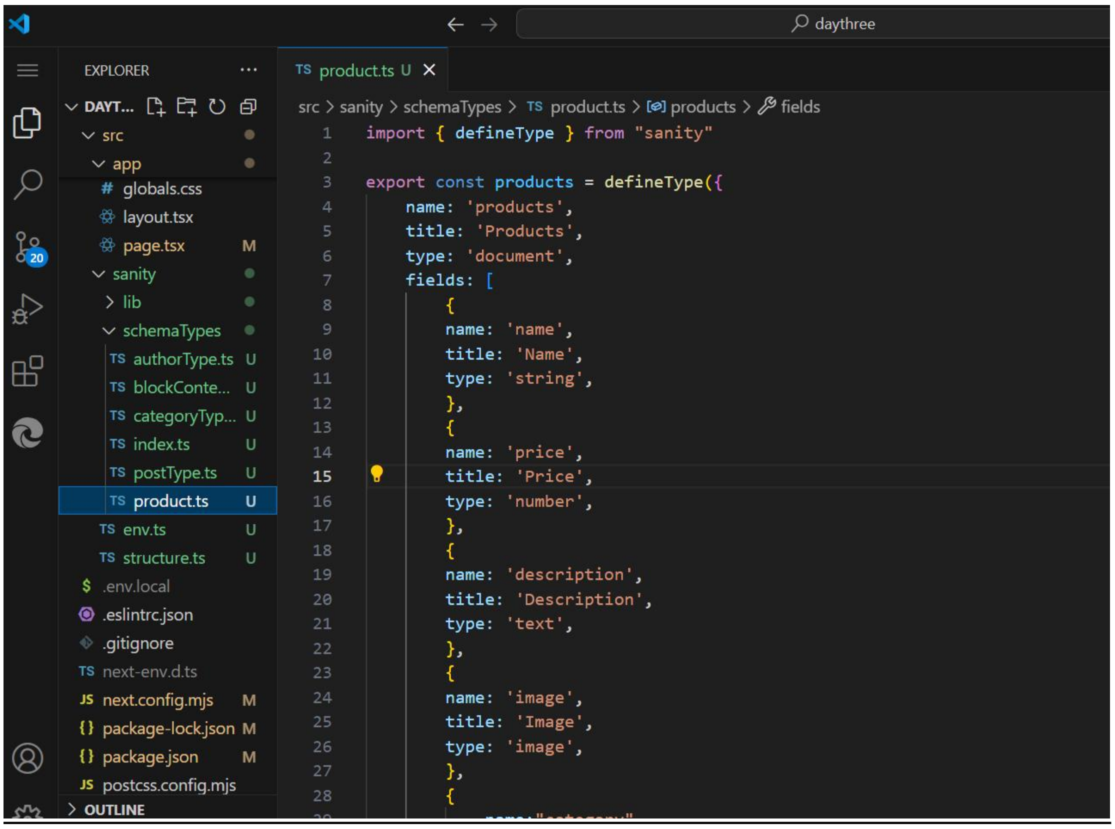
[redacted]

NAME	PERMISSIONS	CREATED
Editor	Editor	2 hours

Use this token for the `SANITY_API_TOKEN` in your `.env.local` file.

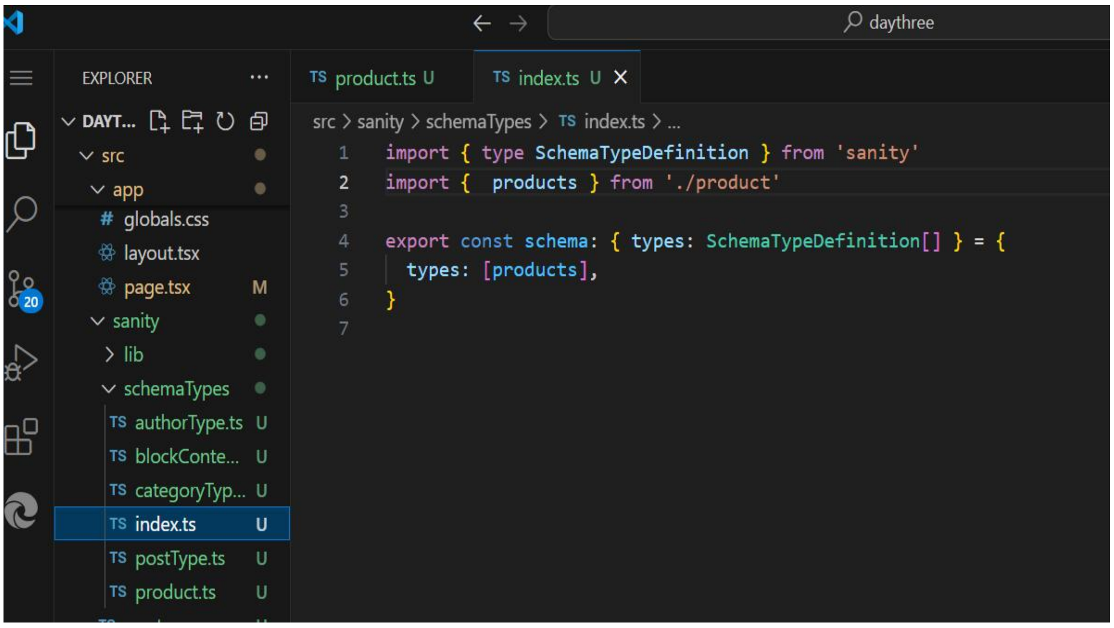
Creating the Sanity Schema:

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



```
import { defineType } from "sanity"

export const products = defineType({
    name: 'products',
    title: 'Products',
    type: 'document',
    fields: [
        {
        name: 'name',
        title: 'Name',
        type: 'string',
        },
        {
        name: 'price',
        title: 'Price',
        type: 'number',
        },
        {
        name: 'description',
        title: 'Description',
        type: 'text',
        },
        {
        name: 'image',
        title: 'Image',
        type: 'image',
        },
        {
```

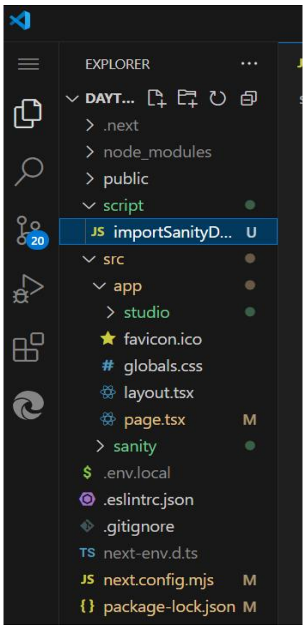
Then, update your `` `sanity/schemaTypes/index.ts` `` file to include the new product schema:

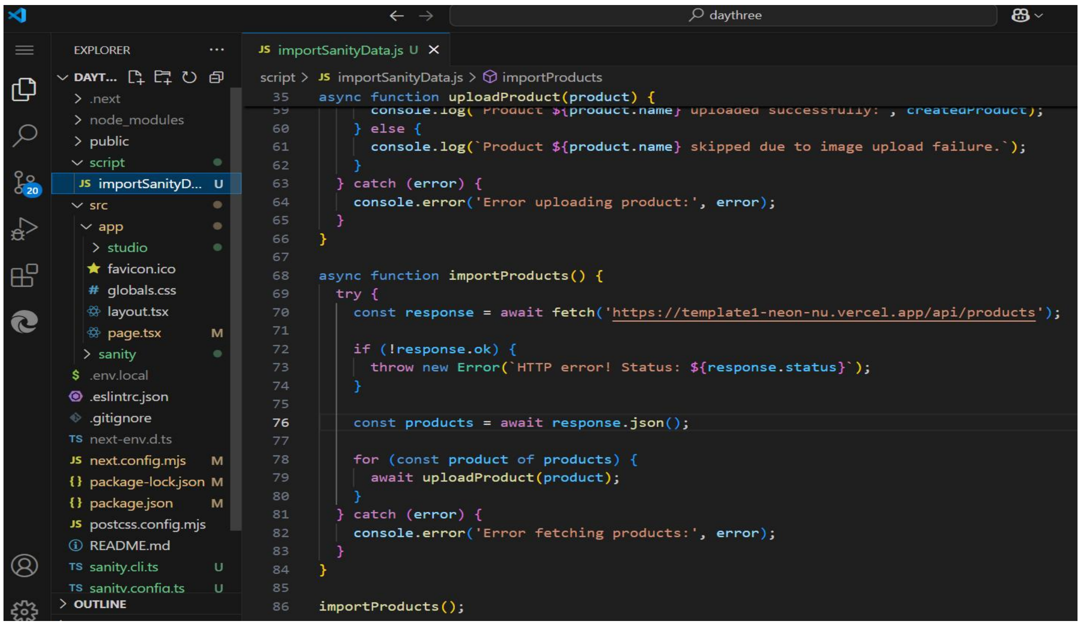


```
import { type SchemaTypeDefinition } from 'sanity'
import {  products } from './product'

export const schema: { types: SchemaTypeDefinition[] } = {
  types: [products],
}
```

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:





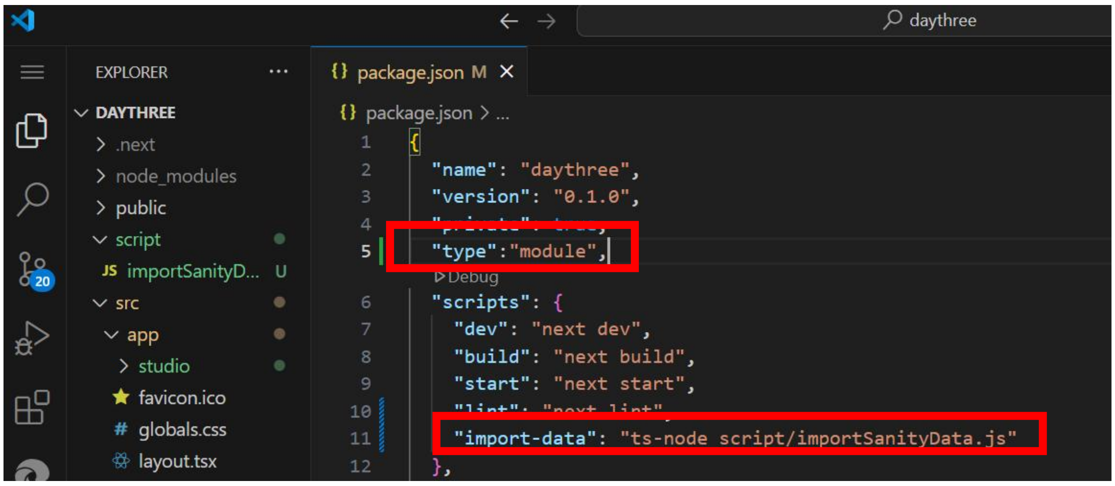
```javascript
async function uploadProduct(product) {
        console.log(`Product ${product.name} uploaded successfully:`, createdProduct);
      } else {
        console.log(`Product ${product.name} skipped due to image upload failure.`);
      }
    } catch (error) {
      console.error('Error uploading product:', error);
    }
}

async function importProducts() {
  try {
    const response = await fetch('https://template1-neon-nu.vercel.app/api/products');

    if (!response.ok) {
      throw new Error(`HTTP error! Status: ${response.status}`);
    }

    const products = await response.json();

    for (const product of products) {
      await uploadProduct(product);
    }
  } catch (error) {
    console.error('Error fetching products:', error);
  }
}

importProducts();
```

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

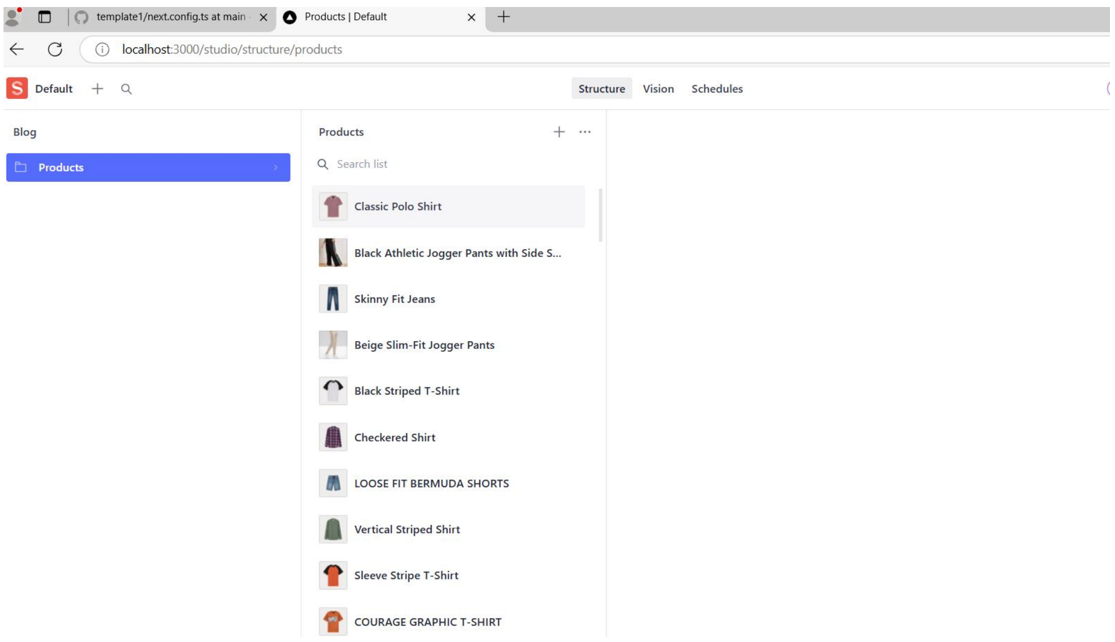


```
{
  "name": "daythree",
  "version": "0.1.0",
  "private": true,
  "type":"module",
  ▷Debug
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "import-data": "ts-node script/importSanityData.js"
  },
```

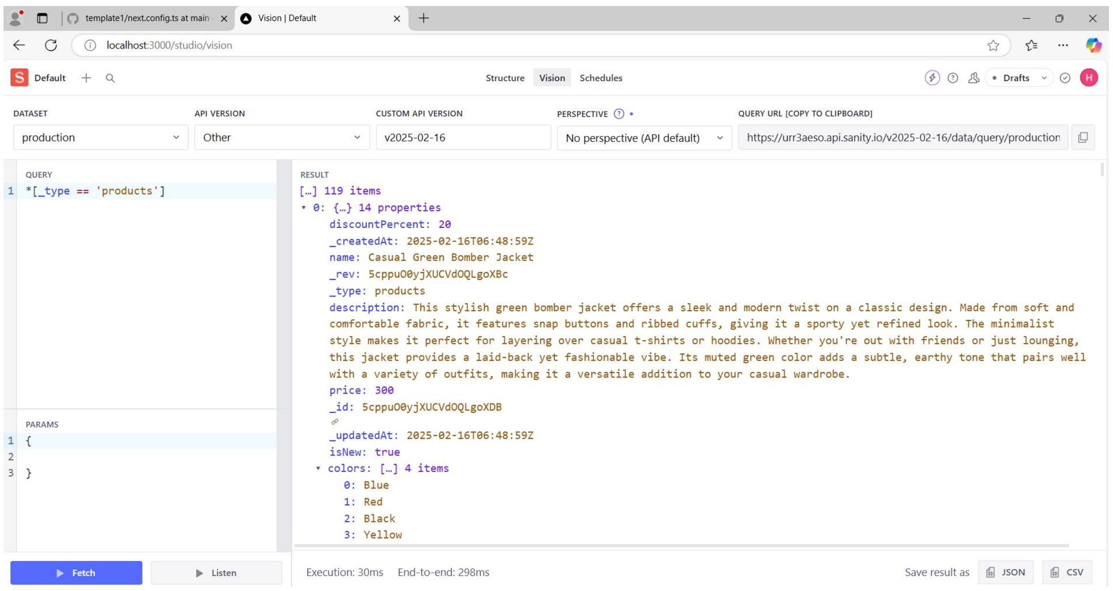
Now run the import script using:

```
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 my Sanity dataset.



Classic Polo Shirt
Black Athletic Jogger Pants with Side S...
Skinny Fit Jeans
Beige Slim-Fit Jogger Pants
Black Striped T-Shirt
Checkered Shirt
LOOSE FIT BERMUDA SHORTS
Vertical Striped Shirt
Sleeve Stripe T-Shirt
COURAGE GRAPHIC T-SHIRT



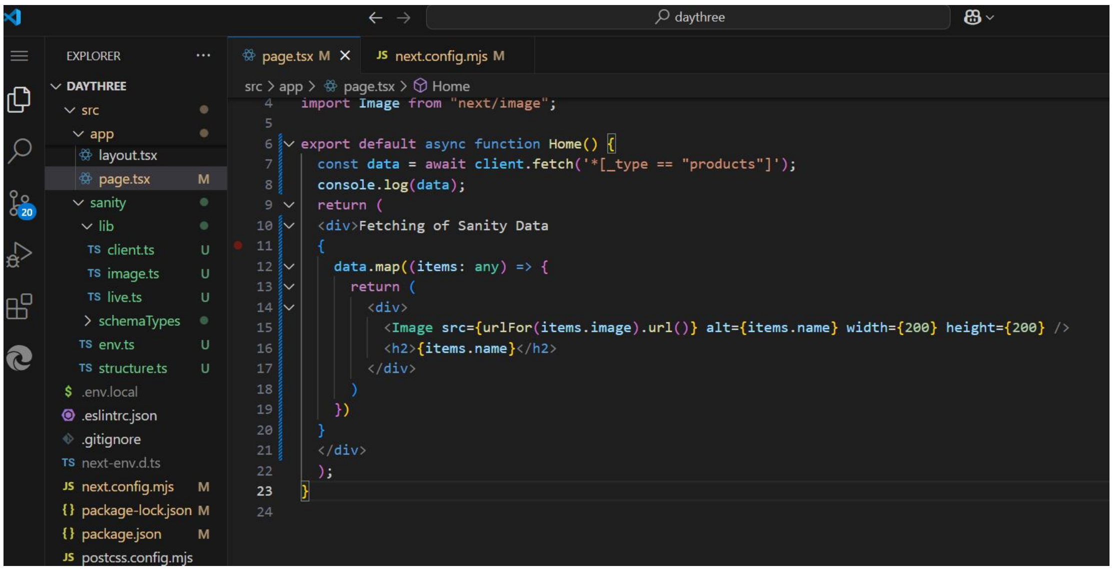
DATASET: production
API VERSION: Other
CUSTOM API VERSION: v2025-02-16
PERSPECTIVE: No perspective (API default)
QUERY URL [COPY TO CLIPBOARD]: https://urr3aeso.api.sanity.io/v2025-02-16/data/query/production

QUERY
```
*[_type == 'products']
```

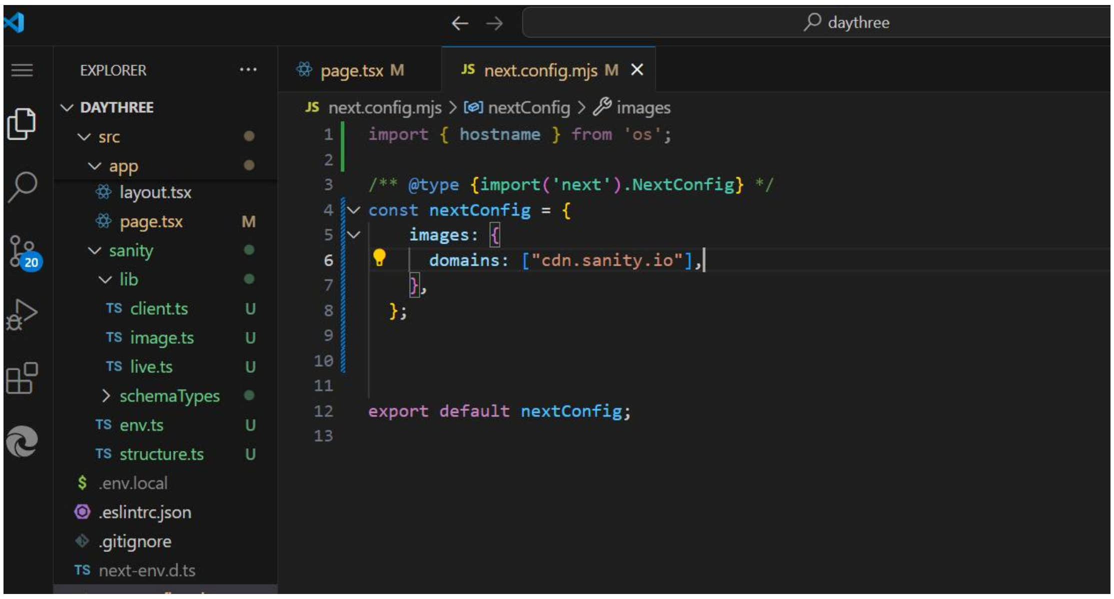
PARAMS
```
{

}
```

RESULT
```
[…] 119 items
▾ 0: {…} 14 properties
    discountPercent: 20
    _createdAt: 2025-02-16T06:48:59Z
    name: Casual Green Bomber Jacket
    _rev: 5cppuO0yjXUCVdOQLgoXBc
    _type: products
    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
    _id: 5cppuO0yjXUCVdOQLgoXDB
    _updatedAt: 2025-02-16T06:48:59Z
    isNew: true
  ▾ colors: […] 4 items
      0: Blue
      1: Red
      2: Black
      3: Yellow
```

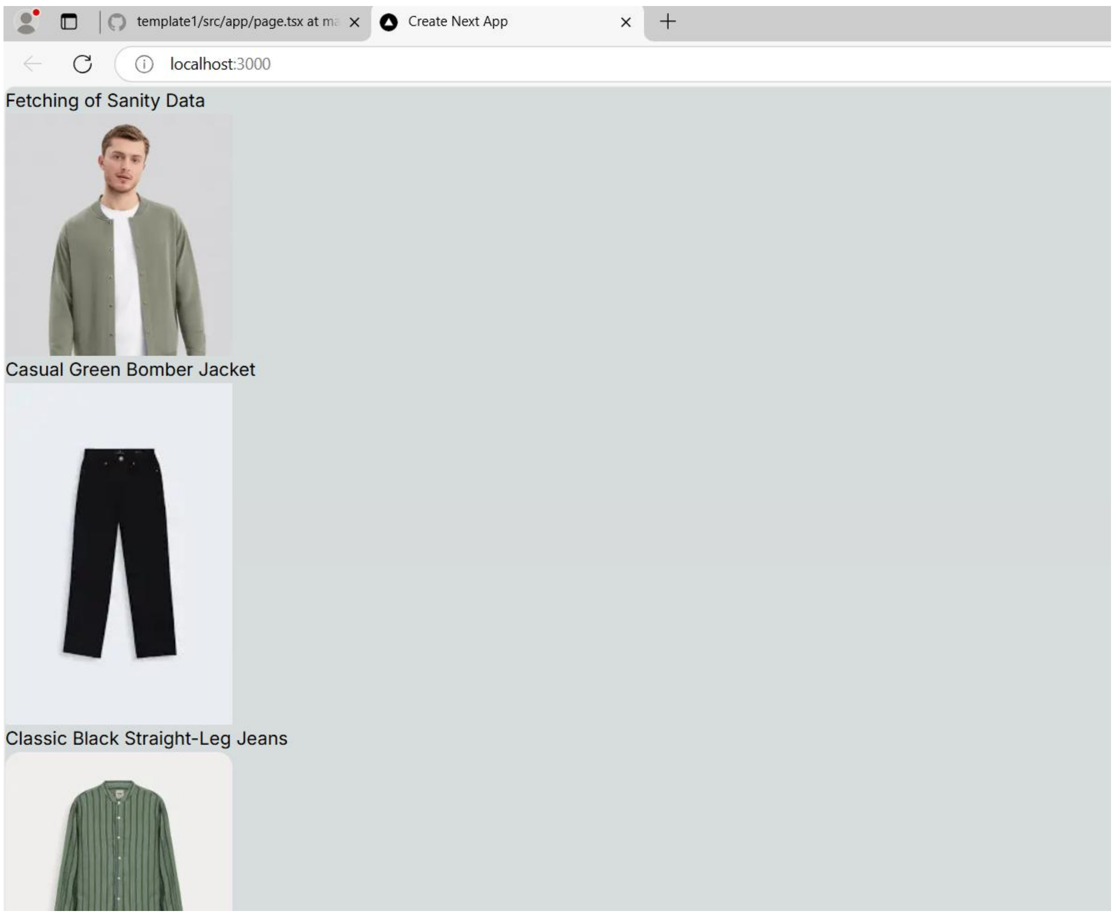
Execution: 30ms End-to-end: 298ms



```
import Image from "next/image";

export default async function Home() {
  const data = await client.fetch('*[_type == "products"]');
  console.log(data);
  return (
  <div>Fetching of Sanity Data
  {
    data.map((items: any) => {
      return (
        <div>
          <Image src={urlFor(items.image).url()} alt={items.name} width={200} height={200} />
          <h2>{items.name}</h2>
        </div>
      )
    })
  }
  </div>
  );
}
```



```
import { hostname } from 'os';

/** @type {import('next').NextConfig} */
const nextConfig = {
    images: {
      domains: ["cdn.sanity.io"],
    },
  };


export default nextConfig;
```



Fetching of Sanity Data

Casual Green Bomber Jacket

Classic Black Straight-Leg Jeans