

```
1 export default {
2   name: 'product',
3   title: 'Product',
4   type: 'document',
5   fields: [
6     {
7       name: 'id',
8       title: 'ID',
9       type: 'string',
10    },
11    {
12      name: 'name',
13      title: 'Name',
14      type: 'string',
15    },
16    {
17      name: 'image',
18      title: 'Image',
19      type: 'image',
20    },
21    {
22      name: 'imagePath',
23      title: 'Image Path',
24      type: 'url',
25    },
26    {
27      name: 'price',
28      title: 'Price',
29      type: 'number',
30    },
31    {
32      name: 'description',
33      title: 'Description',
34      type: 'text',
35    },
36    {
37      name: 'discountPercentage',
38      title: 'Discount Percentage',
39      type: 'number',
40    },
41    {
42      name: 'isFeaturedProduct',
43      title: 'Is Featured Product',
44      type: 'boolean',
45    },
46    {
47      name: 'stockLevel',
48      title: 'Stock Level',
49      type: 'number',
50    },
51    {
52      name: 'category',
53      title: 'Category',
54      type: 'string',
55    },
56  ],
57 };
58
```

```

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 // Load environment variables from .env.local
8 const __filename = fileURLToPath(import.meta.url);
9 const __dirname = path.dirname(__filename);
10 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12 // Create Sanity client
13 const client = createClient({
14   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
15   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
16   useCdn: false,
17   token: process.env.SANITY_API_TOKEN,
18   apiVersion: '2021-08-31',
19 });
20
21 async function uploadImageToSanity(imageUrl) {
22   try {
23     console.log('Uploading image: ${imageUrl}');
24     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
25     const buffer = Buffer.from(response.data);
26     const asset = await client.assets.upload('image', buffer, {
27       filename: imageUrl.split('/').pop(),
28     });
29     console.log('Image uploaded successfully: ${asset._id}');
30     return asset._id;
31   } catch (error) {
32     console.error('Failed to upload image:', imageUrl, error.message);
33     return null;
34   }
35 }
36
37 async function importData() {
38   try {
39     console.log('Migrating data, please wait...');
40
41     // Fetch products from the API
42     const response = await axios.get('https://template-0-beta.vercel.app/api/product');
43     const products = response.data;
44
45     console.log('Products fetched:', products);
46
47     for (const product of products) {
48       let imageRef = null;
49
50       if (product.imagePath) {
51         imageRef = await uploadImageToSanity(product.imagePath);
52       }
53
54       const sanityProduct = {
55         _type: 'product',
56         _id: product.id,
57         name: product.name,
58         category: product.category,
59         description: product.description,
60         discountPercentage: product.discountPercentage,
61         isFeaturedProduct: product.isFeaturedProduct,
62         stockLevel: product.stockLevel,
63         price: parseFloat(product.price),
64         image: imageRef
65         ? {
66           _type: 'image',
67           asset: {
68             _type: 'reference',
69             _ref: imageRef,
70           },
71         }
72         : undefined,
73         imagePath: product.imagePath, // Store original image URL
74       };
75
76       await client.create(sanityProduct);
77       console.log('Product created in Sanity: ${sanityProduct._id}');
78     }
79
80     console.log('Data migrated successfully!');
81   } catch (error) {
82     console.error('Error in migrating data:', error.message);
83   }
84 }
85
86 importData();

```

S

Default

+ Create

Q

StructureVisionSchedules


TasksH


Content


Product


Product


Search list


Luxury Flower Bed


Blue Bed


Red Bed


White Bed


Solid Bed


Matilda Velvet Bed


Diondre Chair


Leisure Sofa Chair Set


Liberty Center

Sleek Modern Table

Replica Table

Alpha Table

Cozy Sofa

High Quality Modern Sofa

Windows Taskbar

Search: Type here to search

Taskbar Icons: File Explorer, Edge, Office, Chrome, etc.

System Tray: S&P 500 +1.00%, 5:22 PM, 1/20/2023

Default

+ Create

StructureVisionSchedules

Tasks

H

DATASET

production

API VERSION

Other

CUSTOM API VERSION

v2025-01-19

PERSPECTIVE

raw

QUERY URL [COPY TO CLIPBOARD]

https://949228j7.api.sanity.io/v2025-01-19/data/query/production?que

QUERY

```
1 *[_type == "product"]{
2   category,
3   id,
4   price,
5   description,
6   stockLevel,
7   imagePath,
8   discountPercentage,
9   isFeaturedProduct,
10  name,
11  "image": image.asset._ref
12 }
```

PARAMS

```
1 {
2
3 }
```

RESULT

```
[...] 63 items
  0: {...} 10 properties
    category: Chair
    id: null
    stockLevel: 25
    imagePath: https://plus.unsplash.com/premium_photo-1668073439372-2ceafa1222b7?q=80&w=1887&auto=format&fit=crop&iixlib=rb-4.0.3&iixid=M3wxMjA3fDB8MHxwaG90by1wYVdlfHx8fGVufDB8fHx8fA%3D%3D
    isFeaturedProduct: true
    name: Chair Wibe
    image: image-8e3afa71668987a0516404c4a818d9538a8da023-1887x2831-jpg
    price: 1200
    description: A sleek outdoor chair with natural wooden elements and a modern look.
    discountPercentage: 10
  1: {...} 10 properties
    image: image-6043c0f8a743bd0c02f98edaba88057f853a1a8c-2070x1380-jpg
    category: Table
    price: 900
    description: A sturdy oak chair with a sleek and minimalist design.
    stockLevel: 18
    imagePath: https://plus.unsplash.com/premium_photo-1681412205470-77848a519359?q=80&w=2070&auto=format&fit=crop&iixlib=rb-4.0.3&iixid=M3wxMjA3fDB8MHxwaG90by1wYVdlfHx8fGVufDB8fHx8fA%3D%3D
    discountPercentage: 10
    isFeaturedProduct: false
    name: Alpha Table
    id: null
  2: {...} 10 properties
    category: Table
```

Fetch

Listen

Execution: 13ms End-to-end: 472ms

Save result as

JSON

CSV

