

DAY-3

API Integration & Data Migration

01

**Understanding The
Provided API**

02

**Validate And
Adjusted Schema**

04

**Data Migration
Option**

05

**API Integration in
nextjs**

07

CHECK LIST



01 Understanding The Provided API

In step Understanding the provided API we made two API one for product & and one for Category. I was given Template 4 so i adjusted the API according to figma file provided earlier.

The API provided didn't have all info i need so i created two API using route handling.

You can see json data

<https://my-appv1.vercel.app/api/product>

<https://my-appv1.vercel.app/api/category>

```
}
async function importData() {
  try {
    console.log('Fetching products from API...')
    const response = await axios.get('https://my-appv1.vercel.app/api/product', {
      headers: {
        'Access-Control-Allow-Origin': '*',
        'Access-Control-Allow-Method': 'GET',
        'Access-Control-Allow-Headers': 'Content-Type'

        // Adjust as needed for your server
      },
    })
  }
}
```

```
{
  "id": "1",
  "name": "Tribù Elio Chair",
  "imagePath": "https://res.cloudinary.com/tribuelio/image/upload/v1641111111/tribuelio-chair-1.jpg",
  "price": "1200",
  "description": "A sleek outdoor chair with a modern design, perfect for your patio or garden.",
  "discountPercentage": 10,
  "isFeatured": true,
  "isTrending": true,
  "isLatestProduct": false,
  "stockLevel": 25,
  "category": "Chair",
  "rating": 4.5,
  "color": [
    "Natural Wood",
    "Grey",
    "Black"
  ],
  "additionalInfo": "Crafted with premium materials for durability and comfort."
},
```

How to implement it

First i use chatgpt by giving a schema from day2 and instruct it to create 20 dataset for product. and then use route handling method to convert into json. and then deploy to vercel. After deploying it, i fetch the given above URL in importProduct.mjs we use mjs because we are using module export. after that i run through some which will be describe below

Note:use vercel link to import data and schema should be made before importing uploading data on sanity

02 Validate and Adjust Your schema

In second step : i use category schema and the product schema made changes in product schema according to figma file and provided category as foreign key in Schema for product.

```
export default {
  name: "product",
  title: "product",
  type: "document",
  fields: [
    {
      name: "name",
      title: "Name",
      type: "string",
      description: "Name of the product",
    },
    {
      name: 'image',
      title: 'Image URL',
      type: 'image',
    },
    {
      name: "price",
      title: "price",
      type: "string",
      description: "Price of the product",
    },
    {
      name: "description",
      title: "description",
      type: "text",
      description: "A detailed description of the product",
    },
    {
      name: "discountPercentage",
      title: "discount Percentage",
      type: "number",
      description: "Discount percentage applied to the product",
    },
    {
      name: "isFeatured",
      title: "Is Featured",
      type: "boolean",
      description: "Whether the product is a featured item",
    },
  ],
}
```

```
export default {
  name: 'category1',
  title: 'category1',
  type: 'document',
  fields: [
    {
      name: 'id',
      title: 'ID',
      type: 'number',
      validation: (Rule:any) =>
        Rule.required(),
    },
    {
      name: 'name',
      title: 'name',
      type: 'string',
      validation: (Rule:any) =>
        Rule.required(),
    },
    {
      name: 'image',
      title: 'Image URL',
      type: 'image',
    },
  ],
};
```

03

```
name: "isTrending",
title: "Is Trending",
type: "boolean",
description: "Whether the product is trending",
},
{
name: "isLatestProduct",
title: "Is Latest Product",
type: "boolean",
description: "Whether the product is a latest addition",
},
{
name: "stockLevel",
title: "Stock Level",
type: "number",
description: "Available stock of the product",
},
{
name: 'category',
type: 'reference', // Reference field
to: [{ type: 'category' }], // Referencing the "category" schema
title: 'Category',
},
{
name: "rating",
title: "Rating",
type: "number",
description: "Rating of the product (out of 5)",
validation: (Rule:any) => Rule.min(0).max(5),
},
{
name: "color",
title: "Color",
type: "array",
of: [{ type: "string" }],
description: "Available colors for the product",
},
{
name: "additionalInfo",
title: "Additional Info",
type: "text",
description: "Additional information about the product",
},
],
};
```

04 Data Migration Option

in this step we are going to create a script folder in main folder and then write a script and import[product or category].mjs and then to use and run script we are going to write node "pathname of script" and then on command prompt run npm run [importdata]

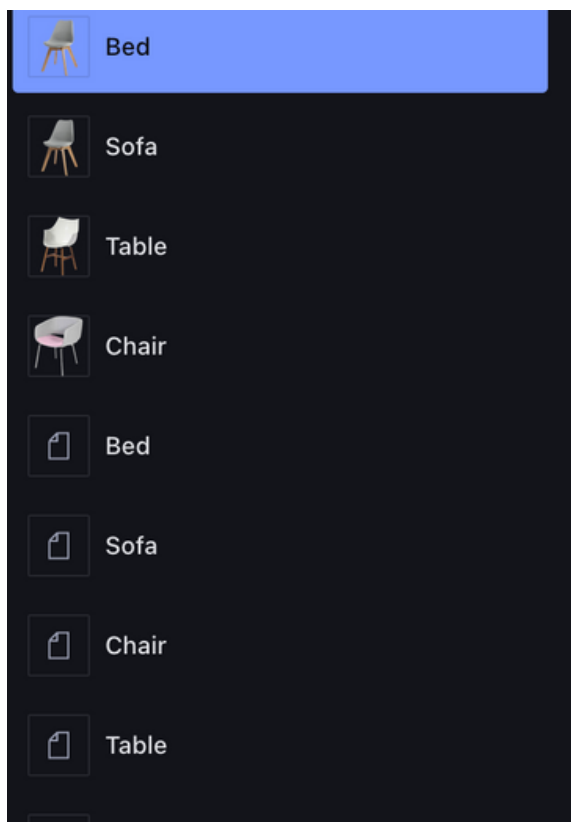
```
at process.processTicksAndRejections (node:internal/p...
)
[maeydahmasroor@Maeydahs-MacBook-Pro ecommerce_sanity % npm
> ecommerce_sanity@0.1.0 import-c
> node script/importCategory.mjs

Fetching products from API...
Fetched 4 category
Processing product: undefined
Uploading product to Sanity: undefined
Product uploaded successfully: qQpkQmFIZ2DSQkhvXJ4shL
Processing product: undefined
Uploading product to Sanity: undefined
Product uploaded successfully: pNdFo0jzUh1upuRBLVT6gF
Processing product: undefined
Uploading product to Sanity: undefined
Product uploaded successfully: rnb16y4e0BKBgYGF3IhVzV
Processing product: undefined
Uploading product to Sanity: undefined
Product uploaded successfully: qQpkQmFIZ2DSQkhvXJ4w1l
Data import completed successfully!
maeydahmasroor@Maeydahs-MacBook-Pro ecommerce_sanity %

loading product to Sanity: undefined
Product uploaded successfully: BQj9k9W1Mf2GlcUg
Data import completed successfully!
maeydahmasroor@Maeydahs-MacBook-Pro my-app1 % np

my-app1@0.1.0 import-category
node script/importCategory.mjs

Fetching products from API...
Fetched 4 category
Processing product: Chair
Uploading product to Sanity: Chair
Product uploaded successfully: 0e0ztuJY6NWdyblBE
Processing product: Table
Uploading product to Sanity: Table
Product uploaded successfully: YjFIo1g1LQZHJZg27
Processing product: Sofa
Uploading product to Sanity: Sofa
Product uploaded successfully: 0e0ztuJY6NWdyblBE
Processing product: Bed
Uploading product to Sanity: Bed
Product uploaded successfully: BQj9k9W1Mf2GlcUg
Data import completed successfully!
```



Some of the info was not properly upload so i use manual import then.

05 API Integration in nextjs

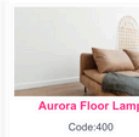
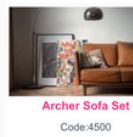
In this step we are going to make utility function like query to fetch data

```
async function fetchData() {
  const query = `
    *[_type == "product" && isLatestProduct == true] {
      _id,
      name,
      price,
      description,
      discountPercentage,
      stockLevel,
      "categoryName": category->title, // Fetch the title of the category
      rating,
      color[],
      additionalInfo,
      image
    }
  `;
}
```

```
const ProductCard = ({product}) => {
  return (
    <div key={product.id}>
      <div className="bg-white shadow-md rounded-lg p-4">
        <div className="hidden group-hover:block h-[29px] flex items-center gap-2">
          <Image src={Heart} alt="c" width={15} height={15} />
          <Image src={CART} alt="c" width={15} height={15} />
          <Image src={magnify} alt="c" width={15} height={15} />
        </div>
      </div>
      <div className="w-[270px] h-[236px] bg-gray-300 flex flex-col align-center justify-center">
        <Image
          src={urlFor(product.image).url()} // Use category image
          alt={product.name || "Category Image"}
          width={200}
          height={200}
        />
      </div>
      <p>No image available</p> // Fallback for missing image
    </div>
  );
};

const ProductDetails = ({product}) => {
  return (
    <div className="hidden group-hover:block h-[29px] flex items-center">
      <a href="#">View Details</a>
    </div>
    <div className="bg-blue-500 hover:bg-blue-600 text-white p-4">
      <h3 className="text-lg font-semibold mb-2 text-white">Product Details</h3>
      <p className="text-gray-600 mb-2 text-center">Product Name: {product.name}</p>
      <p className="text-gray-600 mb-2 text-center">Product Price: {product.price}</p>
    </div>
  );
};
```

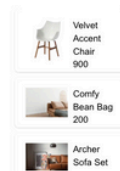
Trending Products



20% off in All Products
[Shop Now](#)



23% off in All Products
[View Collection](#)



Top Categories



GET <http://localhost:3000/api/category>

Params Authorization Headers (7) Body Scripts Tests Settings

Body Cookies Headers (6) Test Results [↻](#)

{ } JSON [Preview](#) [Visualize](#) [⌵](#)

```
1  [
2    {
3      "id": "1",
4      "name": "Tribù Elio Chair",
5      "imagePath": "https://res.cloudinary.com/dilkessdw/image/upload/v1734612322/ElioChair.png",
6      "price": "1200",
7      "description": "A sleek outdoor chair with natural wooden elements and a modern design.",
8      "discountPercentage": 10,
9      "isFeatured": true,
10     "isTrending": true,
11     "isLatestProduct": false,
12     "stockLevel": 25,
13     "category": "Chair",
14     "rating": 4.5,
15     "color": [
16       "Natural Wood",
17       "Grey",
18       "Black"
19     ],
20     "additionalInfo": "Crafted with premium outdoor materials to withstand the elements.",
21   },
22   {
23     "id": "2",
24     "name": "Archer Sofa Set",
25     "imagePath": "https://res.cloudinary.com/dilkessdw/image/upload/v1734612322/ArcherSofaSet.png",
26     "price": "4500",
27     "description": "Luxurious 3-seater sofa with plush cushions and a sturdy frame.",
28     "discountPercentage": 15,
29     "isFeatured": true,
```

Day 3 Checklist: Self-Validation Checklist:

API UNDERSTANDING	YES	
SCHEMA VALIDATION	YES	
DATA MIGRATION	YES(DID BUT I ADDED SOME INFO MANUALLY)	
API INTEGRATION	YES(BUT DYAMIC DATA NOT BEING ON VERCEL)	
SUBMISSION PREPARATION	YES	