

My Furniture Mart

API integration process :

I have integrated products data from api , first I made my schema in sanity folder than import it in index.ts. After it I have used migration script by creating folder scripts in it file named as importSanityData.mjs , in this folder I have fetch the data of all products by its Url , then checks in sanity studio , all done.

Schema adjustments :

Yes I had made some adjustments to schema as I have added some more fields and set the image field as according to my migration script.

Migration steps and tools used:

As I had mentioned in api integration process , it would be same for migration steps , first I had created schema then created a migration script to import all the products data to sanity studio.

Schema (product.ts) :

```
// eslint-disable-next-line import/no-anonymous-default-export
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'id',
      title: 'ID',
      type: 'string',
    },
    {
      name: 'name',
      title: 'Name',
      type: 'string',
    },
    {
      name: 'image', // Changed field name to 'image'
      title: 'Image',
      type: 'image', // Changed type to 'image'
      options: {
        hotspot: true, // Enables image cropping in Sanity Studio
      },
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
    {
      name: 'discountPercentage',
      title: 'Discount Percentage',
      type: 'number',
    },
    {
      name: 'isFeaturedProduct',
      title: 'Is Featured Product',
    },
  ],
}
```

```

      type: 'boolean',
    },
    {
      name: 'stockLevel',
      title: 'Stock Level',
      type: 'number',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
    },
  ],
};

```

Migration script (import data) :

```

import fetch from 'node-fetch';

import { createClient } from "@sanity/client";
import path from 'path';

const client = createClient({
  projectId: '', // Replace with your Sanity project ID
  dataset: 'production', // Replace with your dataset name
  useCdn: false, // Disable CDN for real-time updates
  apiVersion: '',
  token: '', // Replace with your Sanity API token (must have write permissions)
});

// Fetch product data from your API
async function fetchProductData() {
  try {
    const response = await fetch('https://template-0-beta.vercel.app/api/product');
    if (!response.ok) throw new Error(`API Error: ${response.statusText}`);

    return await response.json();
  } catch (error) {
    console.error('Error fetching product data:', error);
    process.exit(1);
  }
}

```

```

// Function to upload images to Sanity
async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);

    const imageResponse = await fetch(imageUrl);
    if (!imageResponse.ok) throw new Error(`Failed to fetch image: ${imageUrl}`);

    const imageBuffer = await imageResponse.arrayBuffer();
    const fileName = path.basename(imageUrl);

    const assetDocument = await client.assets.upload('image',
Buffer.from(imageBuffer), {
      filename: fileName,
    });

    return assetDocument._id; // Return the image reference ID
  } catch (error) {
    console.error('Error uploading image:', error);
    return null;
  }
}

// Delete all existing product documents
async function deleteExistingProducts() {
  try {
    await client.delete({ query: '*[_type == "product"]' });
    console.log('Deleted existing products.');
  } catch (error) {
    console.error('Error deleting existing products:', error);
    process.exit(1);
  }
}

// Import new product data into Sanity
async function importData() {
  try {
    const products = await fetchProductData();
    await deleteExistingProducts();

    const mutations = [];

    for (const product of products) {
      let imageRef = null;
      if (product.imagePath) {

```

```

    imageRef = await uploadImageToSanity(product.imagePath);
  }

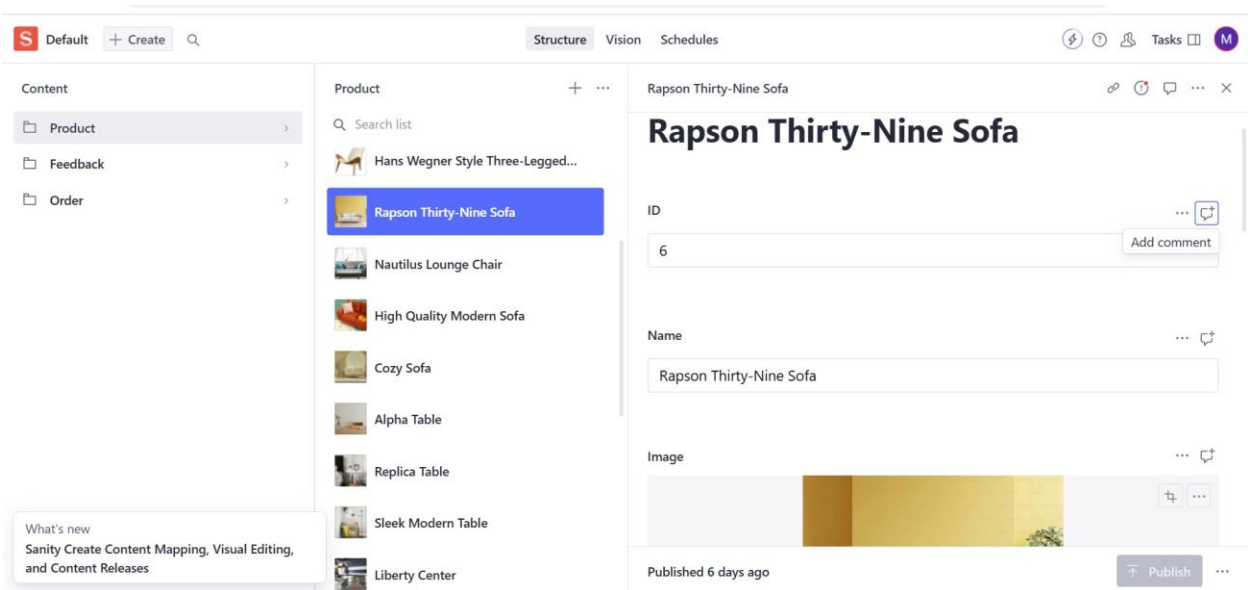
  mutations.push({
    create: {
      _type: 'product',
      id: product.id,
      name: product.name,
      price: product.price,
      description: product.description,
      discountPercentage: product.discountPercentage,
      isFeaturedProduct: product.isFeaturedProduct,
      stockLevel: product.stockLevel,
      category: product.category,
      image: imageRef
      ? {
        _type: "image",
        asset: {
          _type: "reference",
          _ref: imageRef,
        },
      }
      : null,
    },
  });
}

const result = await client.mutate(mutations);
console.log('Successfully imported products:', result);
} catch (error) {
  console.error('Error importing data:', error);
  process.exit(1);
}
}

importData();

```

Sanity studio :



Data displayed on the frontend :

```
import { client } from "@sanity/lib/client";
import Image from "next/image";

// Define a Product type based on your query
interface Product {
  _id: string;
  name: string;
  description: string;
  price: number;
  category: string;
  stockLevel: number;
  isFeaturedProduct: boolean;
  discountPercentage: number;
  image?: {
    asset: {
      url: string;
    };
  };
};

export default async function Featured() {
  // Fetch data inside the component
```

```

const Query = `*[_type == "product"] {
  _id,
  name,
  description,
  price,
  category,
  stockLevel,
  isFeaturedProduct,
  discountPercentage,
  image {
    asset -> { url }
  }
}`;

const data: Product[] = await client.fetch(Query); // Type the data as an array
of Product

return (
  <div className="p-10 my-10">
    <h1 className="text-center font-bold text-3xl mb-8 text-blue-
950">Products</h1>

    <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-
cols-4 gap-6">
      {data.map((product) => (
        <div
          key={product._id}
          className="rounded-lg border p-4 flex flex-col items-center space-y-4
shadow-md hover:shadow-lg transition duration-300"
        >

          {product.image?.asset?.url ? (
            <div className="relative w-full h-48">
              <Image
                src={product.image.asset.url}
                alt={product.name}
                layout="fill"
                style={{ objectFit: 'cover' }} // Update this line
                className="rounded-t-lg"
              />
            </div>
          ) : (
            <p className="text-gray-500">No Image Available</p>
          )}
        </div>
      )}
    </div>
  </div>
);

```

```

<div className="text-center space-y-2">
  <h1 className="text-lg font-bold text-gray-900">{product.name}</h1>
  <p className="text-gray-700">{product.description}</p>
  <p className="text-green-600 font-semibold text-
lg">${product.price}</p>
  <h6 className="text-xl text-zinc-500 font-medium">Discount :
${product.discountPercentage}</h6>
  <h2 className="text-gray-600 text-lg">{product.category}</h2>
  <p className="text-gray-700">Stock Level: {product.stockLevel}</p>
  <p className="text-gray-700">
    Featured: {product.isFeaturedProduct ? "Yes" : "No"}
  </p>
</div>
</div>
  )}
</div>
</div>
);
}

```

Products Displayed :

