

# Day 3 - API Integration Report – Ecommerce and Rental Commerce

## Steps for Day 3:

### 1. Understand the Provided API:

I first customize my API by creating another next js project just for API customization and my API URL end point is /furniture-api

Customizable API Link: <https://template-6-data.vercel.app/api/furniture-api>

API was not according to my day 2 schema and according to my website UI. I added images in API, add rental and stock option also. From This I learn how to customize API manually

### 2. Adjustments Made to Schemas.

Compare your existing Sanity CMS schema (created on Day 2) with the API data structure.

I validate and adjust my schema according to the planning schema that I have done on day 2 documentation.

### Addition:

API Field:

- Change title (related to real product)
- Change tags
- Make and Change the description short.
- Available for Rental
- Stock
- 4 images
- Rental Price Per Day

Schema Type and Field:

- Document type name change --- it was product --- I convert it into furniture.
- I added slug for customize URL.
- I added hotspot in image.
- In schema the image was one. I added Image Array.
- Added isStock Field which is Boolean.
- Added Stock Quantity field
- Added isAvailableForRental Field which is Boolean
- Added RentalPricePerDay Field.

### 3. Data Migration Options:

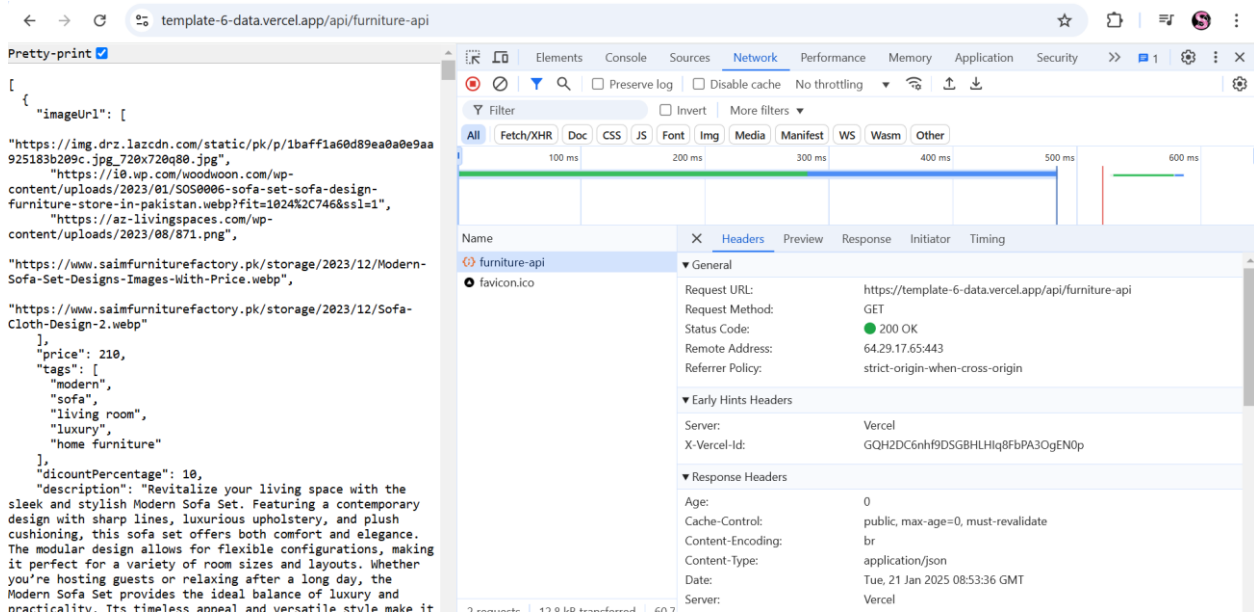
Manual Import:

After Customizing the Provided API , I use Manual Import as mentioned above I customized the Provided API then I have to move towards Manual Import and I do Manual Import. And I migrate the data to Sanity through migrate.mjs file.

Migration steps and tools used:

I used migration script to migrate data from API to Sanity.

## API Calls:



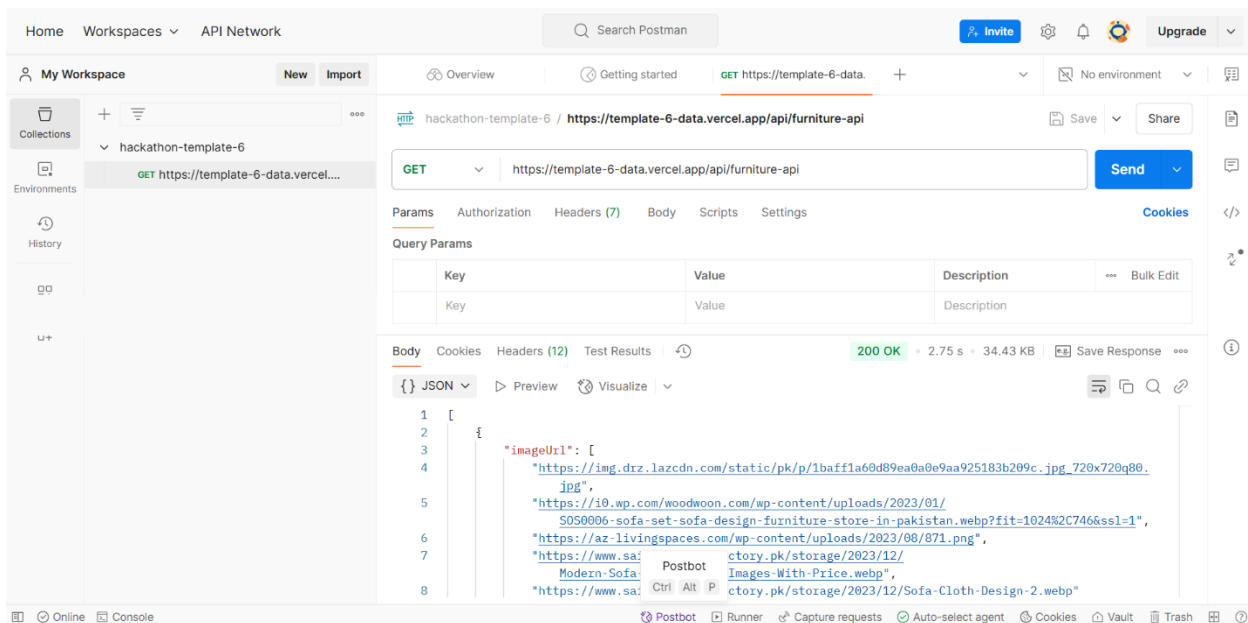
## 4. API Integration in Next.js:

Step 1: Create Utility Functions --- Created

Step 2: Render Data in Components --- Successfully fetched show in the last.

### Step 3: Test API Integration

- I tested my api on postman.
- API data fetched in postman.
- status 200 ok.



## Code snippets for API integration and migration scripts.

```
// Import environment variables from .env.local
import dotenv from "dotenv";
dotenv.config();

// Import the Sanity client to interact with the Sanity backend
import { createClient } from "@sanity/client";

// Load required environment variables
const {
  NEXT_PUBLIC_SANITY_PROJECT_ID, // Sanity project ID
  NEXT_PUBLIC_SANITY_DATASET, // Sanity dataset (e.g., "production")
  NEXT_PUBLIC_SANITY_TOKEN, // Sanity API token
  BASE_URL = "https://template-6-data.vercel.app", // API base URL for products and categories
} = process.env;

// Check if the required environment variables are provided
if (!NEXT_PUBLIC_SANITY_PROJECT_ID || !NEXT_PUBLIC_SANITY_TOKEN) {
  console.error(
    "Missing required environment variables. Please check your .env.local file."
  );
  process.exit(1); // Stop execution if variables are missing
}

// Create a Sanity client instance to interact with the target Sanity dataset
const targetClient = createClient({
  projectId: NEXT_PUBLIC_SANITY_PROJECT_ID, // Your Sanity project ID
  dataset: NEXT_PUBLIC_SANITY_DATASET || "production", // Default to "production" if not set
  useCdn: false, // Disable CDN for real-time updates
  apiVersion: "2025-01-18", // Sanity API version
  token: NEXT_PUBLIC_SANITY_TOKEN, // API token for authentication
});

// Function to upload an image to Sanity
async function uploadImageToSanity(imageUrl) {
  try {
    // Fetch the image from the provided URL
    const response = await fetch(imageUrl);
    if (!response.ok) throw new Error(`Failed to fetch image: ${imageUrl}`);

    // Convert the image to a buffer (binary format)
    const buffer = await response.arrayBuffer();

    // Upload the image to Sanity and get its asset ID
    const uploadedAsset = await targetClient.assets.upload(
      "image",
      Buffer.from(buffer),
      {
        filename: imageUrl.split("/").pop(), // Use the file name from the URL
      }
    );

    return uploadedAsset._id; // Return the asset ID
  } catch (error) {
    console.error("Error uploading image:", error.message);
    return null; // Return null if the upload fails
  }
}

// Function to upload multiple images to Sanity
async function uploadImagesToSanity(imageUrls) {
  const uploadedImages = await Promise.all(
    imageUrls.map((url) => uploadImageToSanity(url))
  );

  return uploadedImages
    .filter((imageId) => imageId) // Filter out failed uploads
    .map((imageId) => ({
      _type: "image",
      asset: {
        _ref: imageId,
      },
    }));
}

// Main function to migrate data from REST API to Sanity
async function migrateData() {
  console.log("Starting data migration...");

  try {
    // Fetch products from the REST API
    const productsResponse = await fetch(`${BASE_URL}/api/furniture-api`);
    if (!productsResponse.ok) throw new Error("Failed to fetch products.");
    const productsData = await productsResponse.json(); // Parse response to JSON

    // Migrate products
    for (const product of productsData) {
      console.log(`Migrating product: ${product.title}`);
      const productImages = await uploadImagesToSanity(product.imageUrl);

      // Prepare the new product object
      const newProduct = {
        _type: "furniture",
        title: product.title,
        description: product.description,
        productImage: productImages,
        price: product.price,
        discountPercentage: product.discountPercentage,
        isNew: product.isNew,
        isStock: product.isStock,
        stock: product.stock,
        availableForRental: product.availableForRental,
        rentalPricePerDay: product.rentalPricePerDay,
        tags: product.tags,
      };

      // Save the product to Sanity
      const result = await targetClient.create(newProduct);
      console.log(`Migrated product: ${product.title} (ID: ${result._id})`);
    }

    console.log("Data migration completed successfully!");
  } catch (error) {
    console.error("Error during migration:", error.message);
    process.exit(1); // Stop execution if an error occurs
  }
}

// Start the migration process
migrateData();
```

Data Migrated to Sanity:

Default

+ Create

Structure

Vision

Schedules

production

Other

v2025-01-18

raw

https://udtrx020.api.sanity.io/v2025-01-18/data/query/production?query=

QUERY

1

\*[\_type == "furniture"]

2

PARAMS

1

{

2

3

}

RESULT

[...] 24 items

0: {...} 16 properties

slug: {...} 2 properties

current: modern-side-table

\_type: slug

isStock: true

\_rev: 4QxLptHrsfZdPI0faGYsyz

dicountPercentage: 10

productImage: [...] 5 items

0: {...} 3 properties

\_type: image

\_key: 50f1526dc5d215cacbce5d70452dc485

asset: {...} 1 property

1: {...} 3 properties

\_type: image

\_key: 2a329d4828b08ea7f8706eca9760b855

asset: {...} 1 property

2: {...} 3 properties

\_type: image

\_key: 2507d36fdd01aecb05bc80f16cca43cc

asset: {...} 1 property

3: {...} 3 properties


\_type: image

\_key: 5ba6d34a8686be81c3d11f8c79a54fd1

asset: {...} 1 property

Product Fetch And Display In UI Dynamically:


Our Products



Modern Side Table

162


Stock: 20



Modern TV Unit

160

Stock: 25




King-Size Bed

90

Available for Rent


Stock: 5



Wooden Desk

340

Out of Stock




Glass Top Coffee Table

200

Available for Rent

Stock: 40




Luxury Dining Table Set

250

Available for Rent

Stock: 30




Mid-Century Nightstand

140

Available for Rent

Stock: 75



3-Seater Sofa Set

224

Available for Rent

Stock: 15

Show More

Furniro

Home

Shop


Blog


Contact

Home

Shop

Mid-Century Nightstand





Mid-Century Nightstand

\$140

5 Customer Review

Bring timeless elegance to your bedroom with the Mid-Century Nightstand. Inspired by classic mid-century design, this piece features clean lines, tapered legs, and a rich wood finish that exudes sophistication. The nightstand is equipped with a spacious drawer and an open shelf, providing ample storage for books, accessories, and bedtime essentials. Its compact size and versatile design make it a perfect addition to any bedroom, seamlessly blending with modern, vintage, or eclectic decor styles. Durable and stylish, the Mid-Century Nightstand is both a functional and aesthetic enhancement to your sleeping space.

Available for Rent




Add To Cart

Compare

SKU: S3001

Tags: Mid-century nightstand

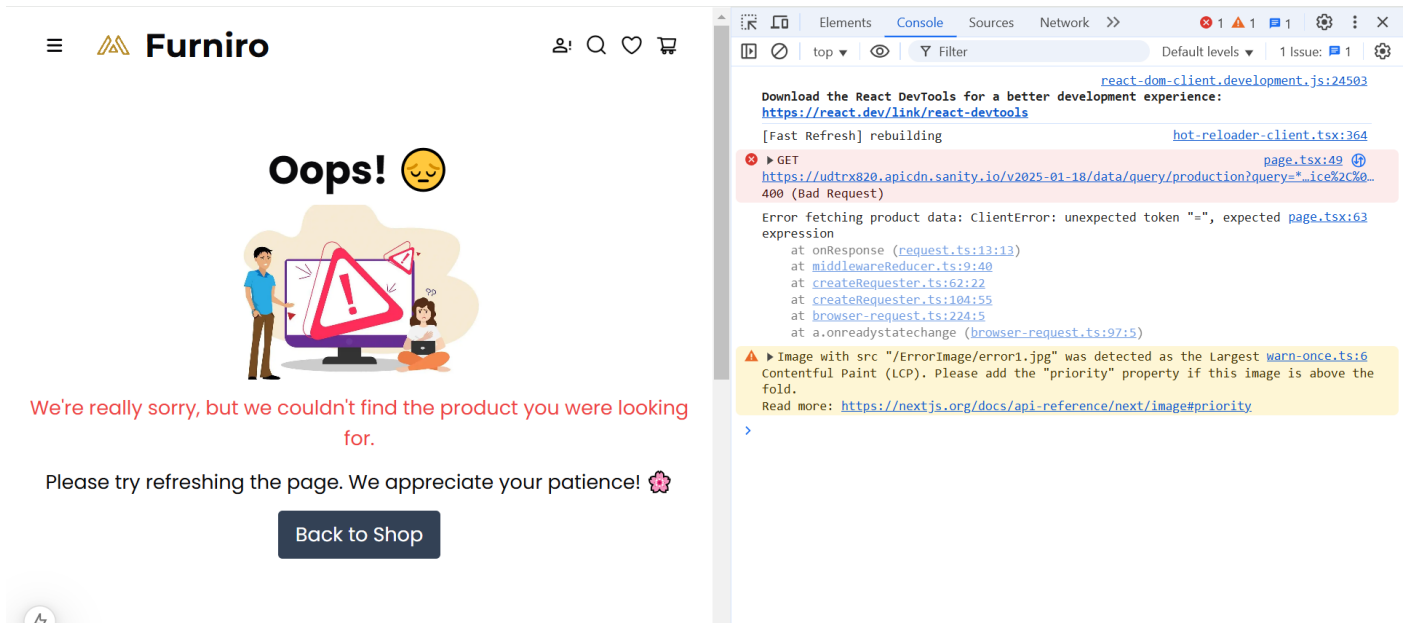
Category: Nightstand, Bedroom, Storage

Share:   

Log Error By Adding ( = )

```
useEffect(() => {
  const fetchSingleProductData = async () => {
    try {
      if (slug) {
        const groqQuery = `*_type === "furniture"{
          title,tags,
          isNew,
          availableForRental,
          stock,
          description,
          dicountPercentage,
          "imageUrls": productImage[].asset->url,
          slug,
          isStock,
          price,
          rentalPricePerDay
        }`;
        const fetchedData = await client.fetch(groq`${groqQuery}`);
        // console.log(fetch);
        // return fetchSingleProductData;
        const findProduct = fetchedData.find(
```

Error Page in loading Product Page:



Error Status:

