Marketplace Hackathon

AVION

DAY 3 - API INTEGRATION AND DATA MIGRATION

Introduction:

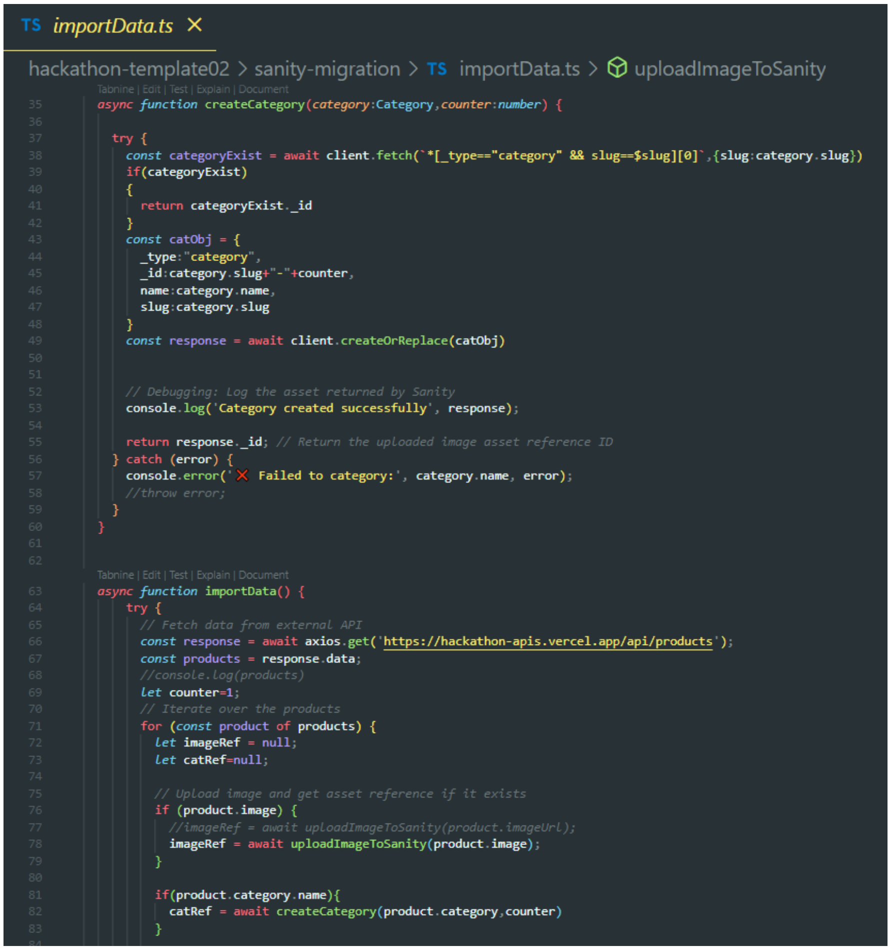
Welcome to our eCommerce store, home decor, featuring a vast selection of crockery, tables, chairs, and furniture. We offer quality craftsmanship and timeless designs to complement every home style. Whether you're refreshing your living room, dining space, or bedroom, our diverse collection has something for every taste and need. Explore our beautiful and functional pieces to create a space that reflects your unique personality and enhances your home's ambiance.

API Overview:

The provided APIs are read-only and are intended to guide schema validation and data migration. APIs to populate their Sanity CMS or import data. Additionally, they have the flexibility to use other APIs or data sources to customize their marketplace. These resources will help in building and structuring the backend effectively.

API:

Template:02: https://hackathon-apis.vercel.app/api/products


TS importData.ts ✕
hackathon-template02 > sanity-migration > TS importData.ts > uploadImageToSanity
Tabnine | Edit | Test | Explain | Document

```
async function createCategory(category:Category,counter:number) {

  try {
    const categoryExist = await client.fetch(`*[_type=="category" && slug==$slug][0]`,{slug:category.slug})
    if(categoryExist)
    {
      return categoryExist._id
    }
    const catObj = {
      _type:"category",
      _id:category.slug+"-"+counter,
      name:category.name,
      slug:category.slug
    }
    const response = await client.createOrReplace(catObj)


    // Debugging: Log the asset returned by Sanity
    console.log('Category created successfully', response);

    return response._id; // Return the uploaded image asset reference ID
  } catch (error) {
    console.error('❌ Failed to category:', category.name, error);
    //throw error;
  }
}

```

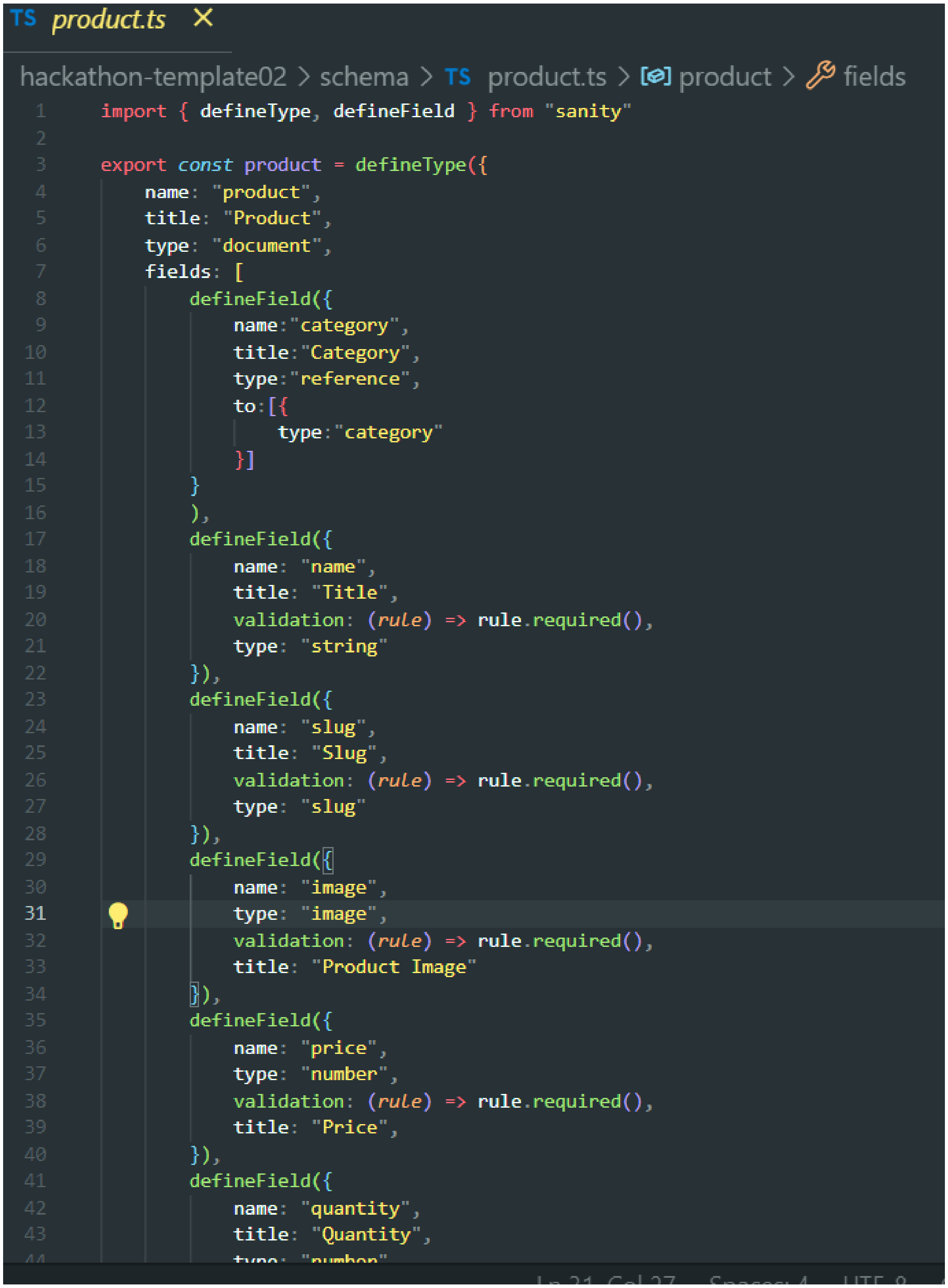
```
async function importData() {
    try {
      // Fetch data from external API
      const response = await axios.get('https://hackathon-apis.vercel.app/api/products');
      const products = response.data;
      //console.log(products)
      let counter=1;
      // Iterate over the products
      for (const product of products) {
        let imageRef = null;
        let catRef=null;

        // Upload image and get asset reference if it exists
        if (product.image) {
          //imageRef = await uploadImageToSanity(product.imageUrl);
          imageRef = await uploadImageToSanity(product.image);
        }

        if(product.category.name){
          catRef = await createCategory(product.category,counter)
        }
```

Schema:

"Product" document for Sanity CMS with essential fields like category, name, slug, image, price, quantity, and tags. It also includes additional fields such as a description, features, and product dimensions (height, width, depth). Each field is validated to ensure that the data is complete, required, and structured correctly for use in an eCommerce environment. This setup supports organizing and managing detailed product data efficiently in your marketplace.



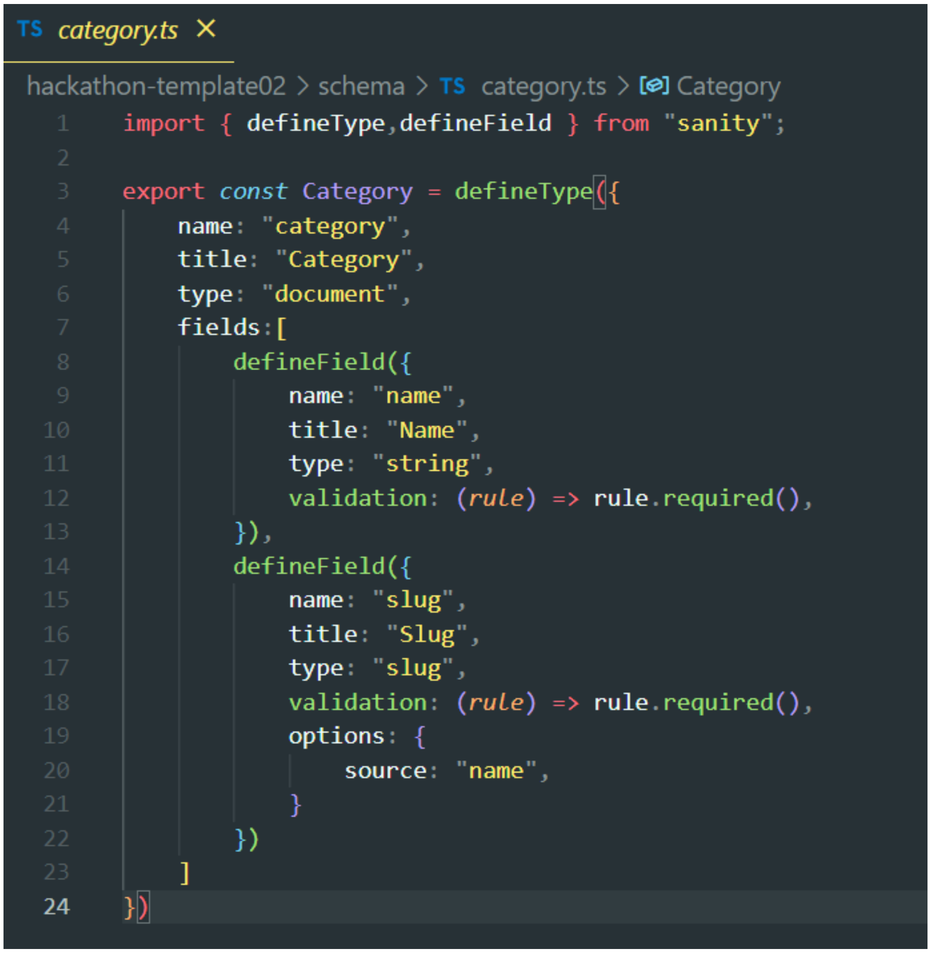
TS product.ts ×
hackathon-template02 > schema > TS product.ts > product > fields
Ln 31, Col 27 Spaces: 4 UTF-8

```ts
import { defineType, defineField } from "sanity"

export const product = defineType({
    name: "product",
    title: "Product",
    type: "document",
    fields: [
        defineField({
            name:"category",
            title:"Category",
            type:"reference",
            to:[{
                type:"category"
            }]
        }
        ),
        defineField({
            name: "name",
            title: "Title",
            validation: (rule) => rule.required(),
            type: "string"
        }),
        defineField({
            name: "slug",
            title: "Slug",
            validation: (rule) => rule.required(),
            type: "slug"
        }),
        defineField({
            name: "image",
            type: "image",
            validation: (rule) => rule.required(),
            title: "Product Image"
        }),
        defineField({
            name: "price",
            type: "number",
            validation: (rule) => rule.required(),
            title: "Price",
        }),
        defineField({
            name: "quantity",
            title: "Quantity",
            type: "number"
```

Category Schema:

This schema defines a "Category" document for Sanity CMS, including essential fields like "name" and "slug". The "name" field is a required string, while the "slug" is automatically generated based on the category name and also marked as required. This structure allows for easy categorization of products within your marketplace, ensuring each category has a unique identifier (slug) for URL generation and consistency across the system.

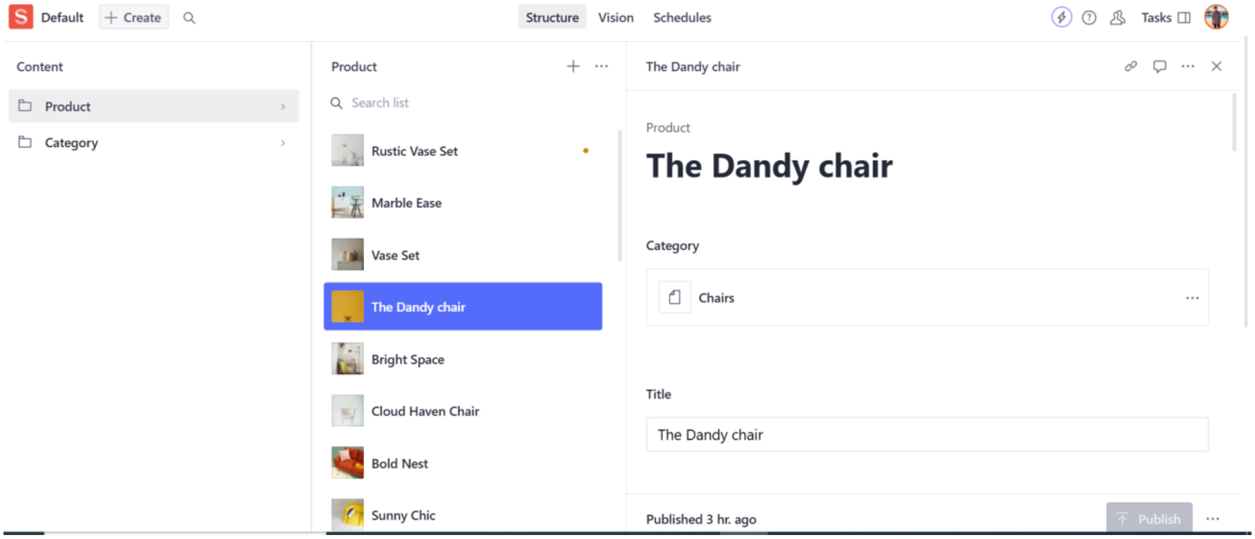


TS category.ts ×
hackathon-template02 > schema > TS category.ts > [⊘] Category

```
import { defineType,defineField } from "sanity";

export const Category = defineType({
    name: "category",
    title: "Category",
    type: "document",
    fields:[
        defineField({
            name: "name",
            title: "Name",
            type: "string",
            validation: (rule) => rule.required(),
        }),
        defineField({
            name: "slug",
            title: "Slug",
            type: "slug",
            validation: (rule) => rule.required(),
            options: {
                source: "name",
            }
        })
    ]
})
```

The image depicts a content management system (CMS) interface likely used for managing product information.



Product

The Dandy chair

Category

Chairs

Title

The Dandy chair

Published 3 hr. ago