# DAY 3 - API INTEGRATION AND DATA MIGRATION General E Commerce

1. Step:

    **.**clone repository using this command

    [https://github.com/bilalmk/hackathon-template02.git](https://github.com/bilalmk/hackathon-template02.git)

**this repository I clone out of my hackathon I clone this repo for send data in my sanity project**

**2. Step:**

    **Install all Dependencies**

    **npm install**

**. Add Configure Environment Variables in the .env file.**

**I had already implemented Sanity in my project, so I used the same environment variables to ensure that the data migrates to the same Sanity project.**

**. Import Data in sanity**

**Run the following command for migrate data in sanity**
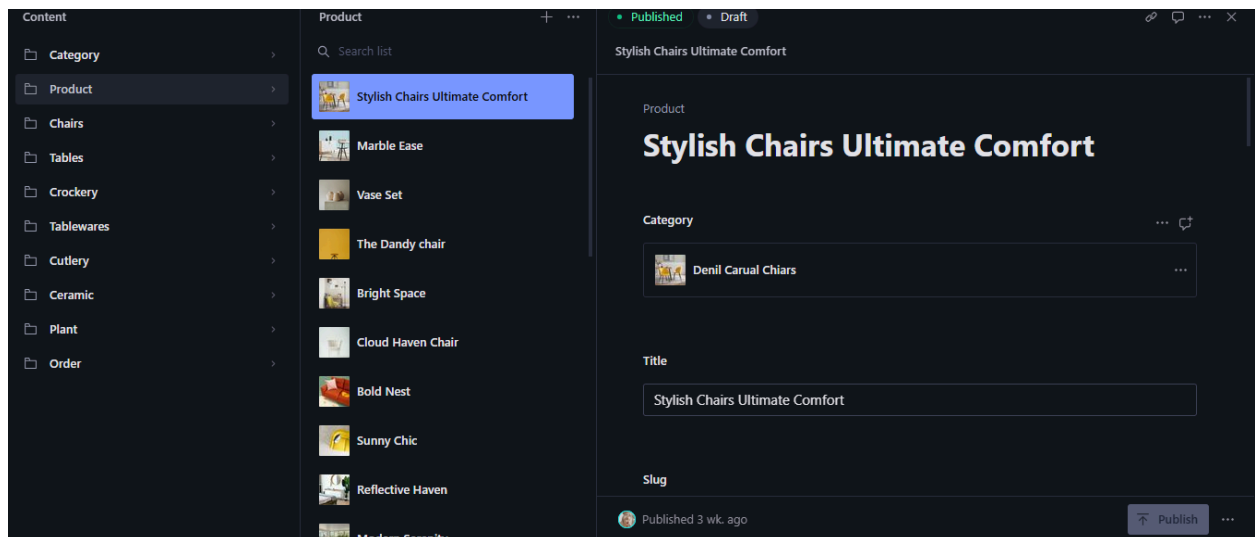
npm run import-data

After running this command, my API data successfully migrated to my Sanity project. One more thing, I created the schema that the instructor provided on GitHub. I copied that schema and

pasted it into my hackathon project, where I had previously created some schemas. For the product schema, I created a separate file in Sanity

**.sanity/schemaTypes/product.ts**



**.Look here my data successfully migrate**

After completing the migration, I manually added some products along with new categories and tags to the schema. These additions were essential as they were missing from the initial API data. Using these custom categories and tags, I successfully fetched and displayed the data on the frontend.
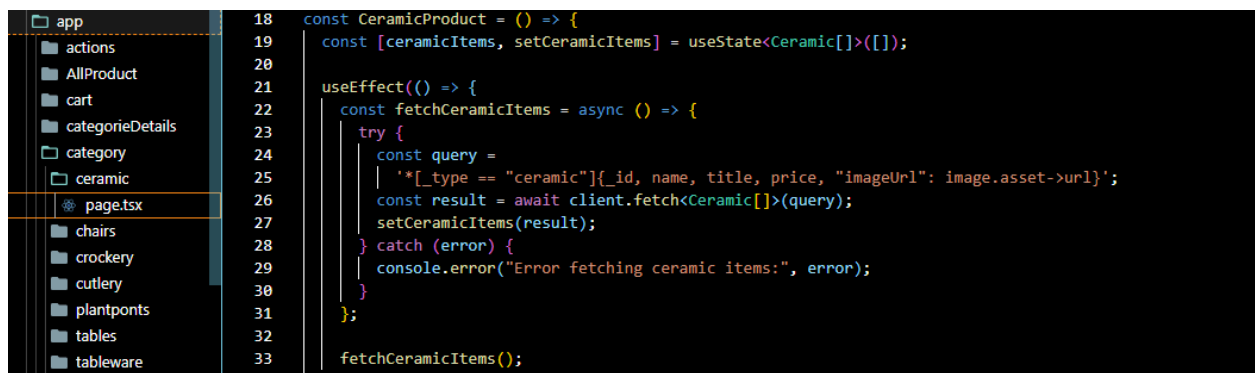
```javascript
import { defineType, defineField } from "sanity";

export const Ceramic = defineType({
  name: "ceramic",
  title: "Ceramic",
  type: "document",
  fields: [
    defineField({
      name: "name",
      title: "Name",
      type: "string",
      validation: (rule) => rule.required(),
    }),
    defineField({
      name: "title",
      title: "Title",
      type: "string",
      validation: (rule) => rule.required(),
    }),
    defineField({
      name: "price",
      title: "Price",
      type: "number",
      validation: (rule) =>
        rule.min(0).warning("Price should not be negative"),
    }),
    defineField({
      name: "inventory",
      title: "Inventory",
      type: "number",
      validation: (rule) => rule.required().min(0).warning("Inventory must be a non-negative number"),
    }),
```

**I manually added products along with custom categories and tags that were missing from the API. This allowed me to fetch and display the relevant data on the frontend effectively.**
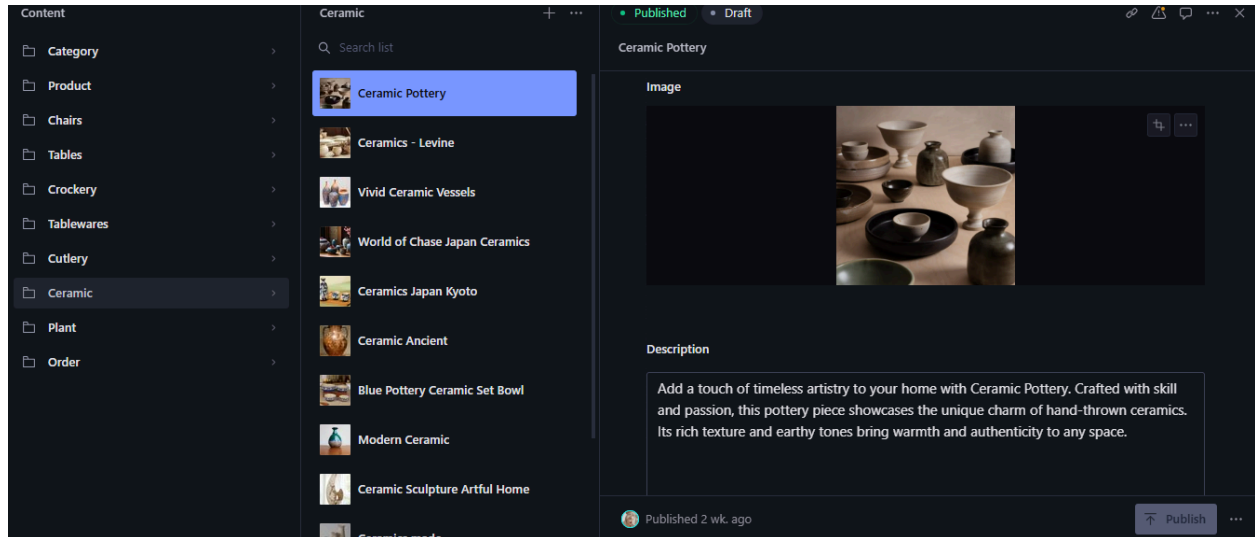


**See here I fetch data from sanity using groq query for display frontend**

```
18  const CeramicProduct = () => {
19    const [ceramicItems, setCeramicItems] = useState<Ceramic[]>([]);
20
21    useEffect(() => {
22      const fetchCeramicItems = async () => {
23        try {
24          const query =
25            '*[_type == "ceramic"]{_id, name, title, price, "imageUrl": image.asset->url}';
26          const result = await client.fetch<Ceramic[]>(query);
27          setCeramicItems(result);
28        } catch (error) {
29          console.error("Error fetching ceramic items:", error);
30        }
31      };
32
33      fetchCeramicItems();
```

**Display UI:**

Home     Ceramics     Tables     Chairs     Crockery     Tableware     Cutlery     Plant Pots

# Elegant Ceramic Collection

**Laima Ceramics**

£80

★ ★ ★ ★ ★

No reviews yet.

♡

**Modern Ceramic**

£30

★ ★ ★ ★ ★

No reviews yet.

♡

**Ceramic Ancient**

£90

★ ★ ★ ★ ★

No reviews yet.

♡

**Contemporary Ceramic**

£20

★ ★ ★ ★ ★

No reviews yet.

♡

```
16  }
17
18  const Cards = () => {
19    const [products, setProducts] = useState<Product[]>([]);
20
21    useEffect(() => {
22      const getProducts = async () => {
23        const allProducts = await fetchAllProducts();
24        setProducts(allProducts);
25      };
26
27      getProducts();
28    }, []);
29
30    return (
31      <div className="px-4 md:px-8 lg:px-36 h-full">
32        <h1 className="text-[#2A254B] font-clash text-2xl md:text-3xl lg:text-4xl font-medium leading-normal mt-16 mb-8">
33          New Products
34        </h1>
35
36        <div className="overflow-hidden mt-6">
37          <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
38            {products.map((product) => (
39              <div key={product._id} className="w-full bg-white p-4 rounded-lg shadow-xl transition-transform transform hove
40                <Link href={`/products/${product._id}`} passHref>
41                  <div className="relative">
42                    <img
43                      src={urlFor(product.image)}
44                      alt={product.name}
45                      className="w-full h-[230px] object-cover rounded-lg transition-all duration-300"
46                    />
47                    <div className="absolute top-0 left-0 w-full h-full bg-black opacity-0 hover:opacity-20 rounded-lg trans
```

# New Products


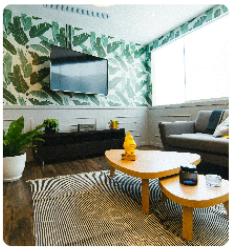
**Stylish Chairs Ultimate C...**

£80

★ ★ ★ ★ ★

No reviews yet.

♡



**The Poplar suede sofa**

£980

★ ★ ★ ★ ★

No reviews yet.

♡



**Tropical Vibe**

£550

★ ★ ★ ★ ★

No reviews yet.

♡



**Sleek Living**

£300

★ ★ ★ ★ ★

No reviews yet.

♡