

## Day3-API Integration Report-Brands Bazar

### API Integration process:

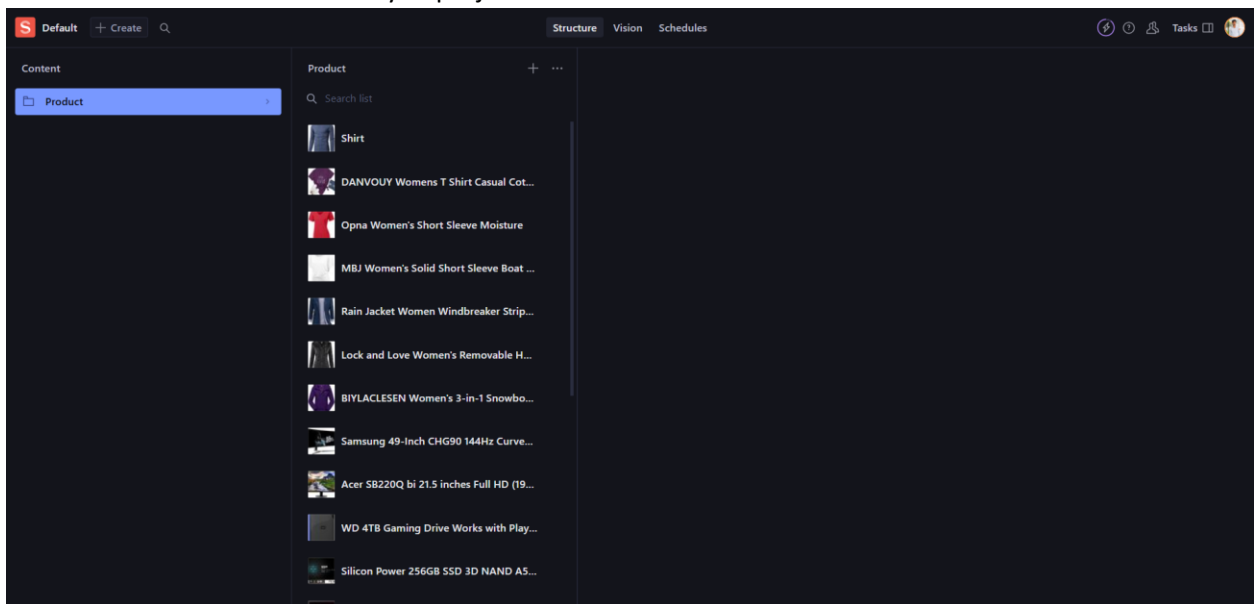
In this process of integrating API in to migrate to Sanity.io following steps were followed:

### Migrating Data to Sanity.io

1. Firstly, created a clone of Sir Ali Jawad's sanity migration repo.  
[GitHub - jawwad-ali/sanity-migration](https://github.com/jawwad-ali/sanity-migration)
2. Now run the command (npm i) to install needed node-modules.
3. Now create a project in Sanity.io.
4. Generate an API token.
5. Create a ".env" file in the root directory and set environment variables like shown in the picture below.

```
1 projectId="Your_Project_ID"
2 dataset="production"
3 token="Your_API_Token"
```

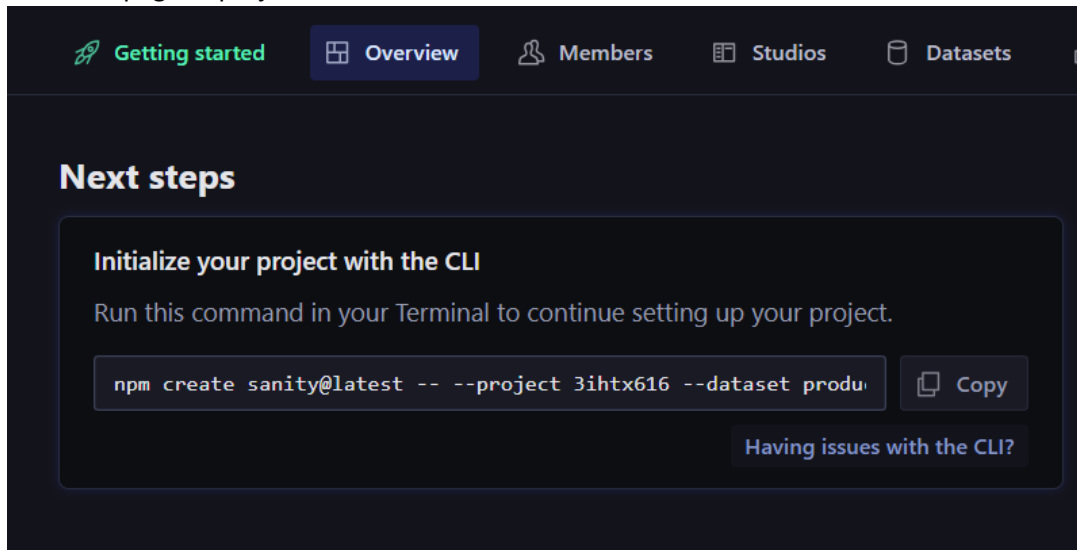
6. Now run the command `tsc && node importData.js` to migrate data from API to Sanity.
7. Now the data should be in Sanity.io project.



### Adjusting Schemas

**Note:** I am using a previous UI from the GIAIC UI/UX Hackathon so that more time can be spent to build functionality. How-ever I have made some changes in the UI according to project goals.

To make and adjust schemas we link our sanity to our Next.js UI by running the command on overview page of project.



Now make a file of `product.ts` in the schema folder and make schemas according to your data.

src > sanity > schemaTypes > product.ts > product > fields

```
1  import { Rule } from "sanity";
2
3
4  const product = {
5    name: "product",
6    title: "Product",
7    type: "document",
8    fields: [
9      {
10       name: "name",
11       title: "Name",
12       validation: (rule: Rule) => rule.required(),
13       type: "string",
14     },
15     {
16       name: "slug",
17       title: "Slug",
18       validation: (rule: Rule) => rule.required(),
19       type: "slug",
20       options: {
21         source: "name",
22         maxLength: 96,
23       },
24     },
25     {
26       name: "price",
27       title: "Price",
28       validation: (rule: Rule) => rule.required(),
29       type: "number",
30     },
31     {
32       name: "image",
33       title: "Image",
34       validation: (rule: Rule) => rule.required(),
35       type: "image",
36       options: {
37         hotspot: true,
38       },
39     },
40     {
41       name: "description",
42       title: "Description",
43       validation: (rule: Rule) => rule.required(),
44       type: "text",
45     },
46     {
47       name: "tags",
48       type: "array",
49       title: "Tags",
50       of: [{ type: "string" }],
51     },
52     {
53       name: "isNew",
54       type: "boolean",
55       title: "New Badge",
56     },
57   ],
58 };
59
60 export default product;
61
```

Product

# New Product

Name



Slug

Generate

Price

Image



Drag or paste image here



Upload



Select

Description

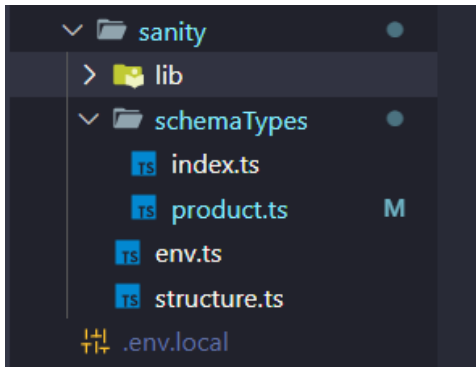
Tags

No items

+ Add item



New Badge



After adjusting the schemas according to requirements we are ready to move ahead and display the data on frontend.

## Displaying Content to Frontend

I am using Sanity.io's GROQ to fetch data to my frontend.

### Steps:

1. Navigate to vision in sanity.
2. According to my listing requirement I am using this GROQ query:

3. In my code I have used async function to fetch data from sanity.

```
const getFtProducts = async () => {  
  const product = await client.fetch(`  
    *[_type == "product"]{  
      _id,  
      name,  
      decsription,  
      price,  
  
      "image_url": image.asset->url,  
    }  
  `);  
  return product;  
};
```

4. Now using useEffect and useState to handle and update the UI with the API fetched data as shown in the code below.

```
const [products, setProducts] = useState([]);  
  
useEffect(() => {  
  const fetchProducts = async () => {  
    const productsData = await getFtProducts();  
    setProducts(productsData);  
  };  
  
  fetchProducts();  
}, []);
```

5. I have used array map method for displaying data on the UI.

```

201 {products.map(
202   (product: {
203     id: string;
204     name: string;
205     price: number;
206     image_url: string;
207   }) => (
208     <div className="w-[241px] h-[639px] " key={product.id}>
209       <Image
210         className="h-80 "
211         alt={product.name}
212         src={product.image_url}
213         width={239}
214         height={420}
215       />
216       <div className="w-[239px] h-[188px] flex flex-col pt-1 items-center gap-2 ">
217         <h1 className="font-bold text-[16px] text-[#252B42] leading-[24px]">
218           {truncateText(product.name, 20)}
219         </h1>
220         <p className="text-[14px] leading-[24px] font-bold text-[#737373]">
221           English Department
222         </p>
223         <div className="flex gap-1 items-center">
224           <p className="font-bold text-[16px] text-[#23856D]">
225             {product.price}
226           </p>
227         </div>
228         <div className="py-3 flex gap-2">
229           <div className="rounded-full w-[16px] h-[16px] bg-[#23A6F0]"></div>
230           <div className="rounded-full w-[16px] h-[16px] bg-[#23856D]"></div>
231           <div className="rounded-full w-[16px] h-[16px] bg-[#E77C40]"></div>
232           <div className="rounded-full w-[16px] h-[16px] bg-[#252B42]"></div>
233         </div>
234       </div>
235     </div>
236   )
237 )}
238

```

## Conclusion:

The conclusion of this process turns out to be that our UI now updates according to the backend (Sanity.io). The products are now displayed dynamically and will update when the backend (Sanity Project) updates.





Shirt

English Department

697



Fjallraven - Foldsac...

English Department

109.95



SanDisk SSD PLUS 1TB...

English Department

109



Silicon Power 256GB ...

English Department

109



WD 4TB Gaming Drive ...

English Department



Acer SB220Q bi 21.5 ...

English Department



Samsung 49-Inch CHG9...

English Department



BIYLACLESEN Women's ...

English Department