

Technical Documentation Hacthon 2025

Day 3 – API Integration Report- General Ecommerce Site

API Integration Process

1. Sanity Client Creation (Key Variables: PROJECT_ID, DATASET & API_TOKEN)
2. Environmental Setup: Installed necessary dependencies such as @sanity/client, axios.
3. Allow Token permission from Sanity,
4. Data Fetching
5. Error Handling

Fields Added:

1. Product Colors
2. Product Sizes
3. Product Status

Migration Steps and Tools Used:

1. Verify and tested Smooth working of APIs with Postman and ThunderClient.

Product Scheme

```
export default ({
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'title',
      type: 'string',
      title: 'Title',
    },
    {
      name: 'price',
      type: 'number',
      title: 'Price',
    },
    {
      name: 'quantity',
      type: 'number',
      title: 'Quantity',
    },
    {
      name: 'colors',
      title: 'Colors',
      type: 'array',
      of: [{ type: 'string' }],
    },
    {
      name: 'size',
      title: 'Size',
      type: 'array',
      of: [{ type: 'string' }],
    },
    {
      name: 'image',
      type: 'image',
      title: 'Image',
      options: {
        hotspot: true,
      },
    },
    {
      name: 'status',
      type: 'string',
      title: 'Status',
      options: {
        list: [
          { title: 'Available', value: 'Available' },
          { title: 'Sold Out', value: 'SoldOut' },
        ],
      },
    },
  ],
})
```

API Calls

```
import type { NextApiRequest, NextApiResponse } from "next";
import sanityClient from "../../sanity/lib/client"

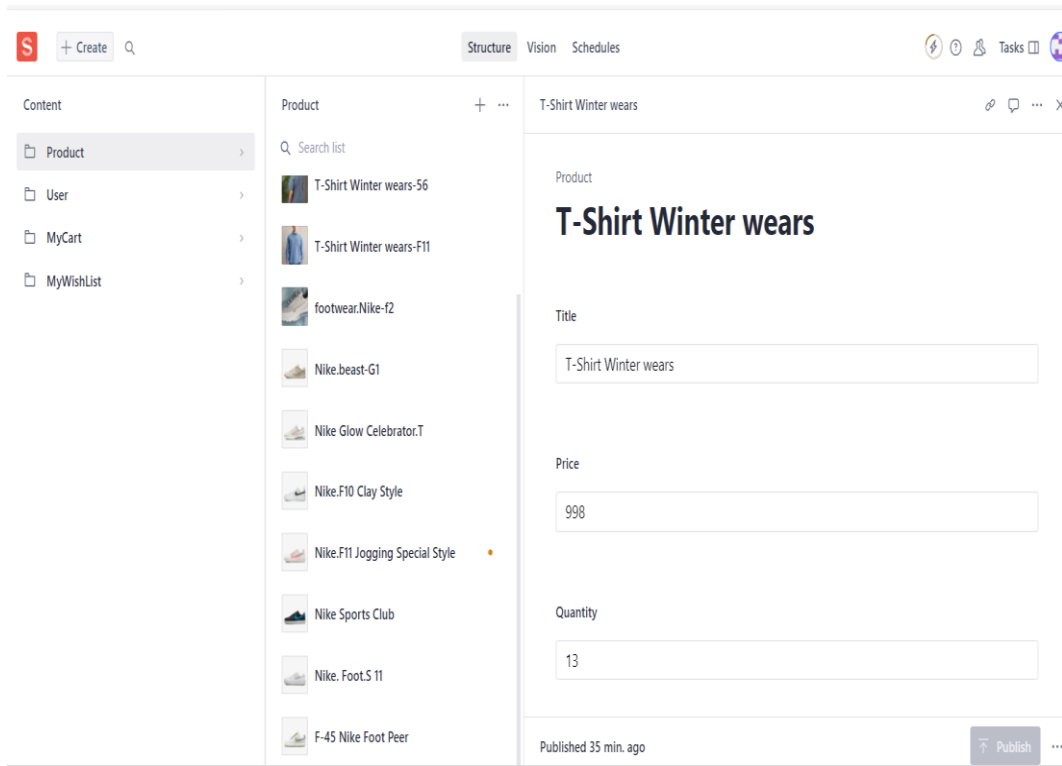
type Data = {
  name: string;
};

export default async function handler( req: NextApiRequest, res: NextApiResponse<Data>,)
{
  if(req.method === "GET"){
    const category = req.query.category; // Extract category from query
    const query = `*[_type == "product" ${category ? `&& category == $category` : ''}]{
      _id,
      title,
      price,
      quantity,
      category,
      size,
      colors,
      status,
      "imageUrl": image.asset->url,
      slug
    }`;

    const items = await sanityClient.fetch(query, category ? { category } : {});

    res.status(200).json(items);
  }
}
```

Sanity CMS Fields



Data Displayed at Frontend

