

## **Hackathon Day : 3**

### **Api Integration project**

#### **Gernal E-commerce Marketplace**

On the third day of the Technical Marketplace project, we integrated the API to fetch data. Here's the process for API integration:

**1. Install Sanity in Next.js**

Use `npm` to install Sanity in your Next.js project.

**2. Create a New Project in Sanity**

Set up a new project in Sanity, generating a unique ID and token.

**3. Add Credentials to Environment File**

Add the generated ID and token to the local environment file for secure access.

**4. Create Schema in Sanity**

Define a schema in the `product.ts` file using a template.

**5. Populate Import Data File**

Fill the `importData.mjs` file with the API data to be imported.

**6. Install Axios**

Install Axios in the project for making API requests.

**7. Fetch Data**

Run the command to fetch product data from the API and display it in the project.

**8. Verify Data in Sanity**

Check if the product data has been successfully fetched and displayed in Sanity.

**Mudassir Ali**

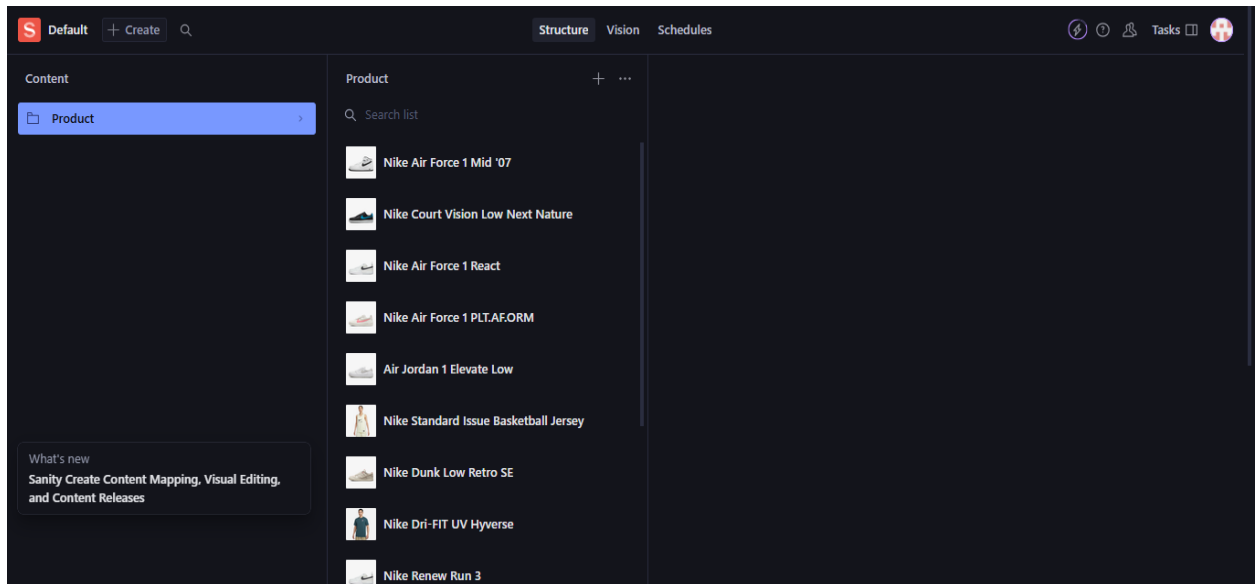
## Fetch Data from Api :-

```
const response = await axios.get('https://template-03-api.vercel.app/api/products');  
const products = response.data.data;
```

## Schema:-

```
export const productSchema = {  
  name: 'product',  
  title: 'Product',  
  type: 'document',  
  fields: [  
    {  
      name: 'productName',  
      title: 'Product Name',  
      type: 'string',  
    },  
    {  
      name: 'category',  
      title: 'Category',  
      type: 'string',  
    },  
    {  
      name: 'price',  
      title: 'Price',  
      type: 'number',  
    },  
    {  
      name: 'inventory',  
      title: 'Inventory',  
      type: 'number',  
    },  
    {  
      name: 'colors',  
      title: 'Colors',  
      type: 'array',  
      of: [{ type: 'string' }],  
    },  
    {  
      name: 'status',  
      title: 'Status',  
      type: 'string',  
    },  
    {  
      name: 'image',  
      title: 'Image',  
      type: 'image', // Using Sanity's image type for image field  
      options: {  
        hotspot: true,  
      },  
    },  
    {  
      name: 'description',  
      title: 'Description',  
      type: 'text',  
    },  
  ],  
}
```

## Sanity fields:-



## Migration Script :-

```
async function uploadImageToSanity(imageUrl) {
  try {
    console.log('uploading image: ${imageUrl}');
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop()
    });
    console.log('Image uploaded successfully: ${asset._id}');
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
}

// CodeLens: Refactor | Explain | Generate | JSDoc | X
async function importData() {
  try {
    console.log('migrating data please wait...');
    // API endpoint containing car data
    const response = await axios.get('https://template-83-api.vercel.app/api/products');
    const products = response.data.data;
    console.log('products ==> ', products);

    for (const product of products) {
      let imageUrl = null;
      if (product.image) {
        imageUrl = await uploadImageToSanity(product.image);
      }

      const sanityProduct = {
        _type: 'product',
        productName: product.productName,
        category: product.category,
        price: product.price,
        inventory: product.inventory,
        colors: product.colors || [], // Optional, as per your schema
        status: product.status,
        description: product.description,
        image: imageUrl ? {
          type: 'image',
          asset: {
            _type: 'reference',
            _ref: imageUrl,
          },
        } : undefined,
      };
      await client.create(sanityProduct);
    }

    console.log('Data migrated successfully!');
  } catch (error) {
    console.error('Error in migrating data ==> ', error);
  }
}

importData();
```

Api Integration	Schema validation	Data Migration	Api Integration (Next.js)	Submission Prepration
Done	Done	Done	Done	Done

**Mudassir Ali**