

DAY 3 – API INTEGRATION AND DATA MIGRATION

- The focus for Day 3 is integrating APIs and migrating data into Sanity CMS to build a functional marketplace backend. Students will learn how to utilize APIs for reference, migrate data into Sanity CMS, and ensure compatibility with their templates. This approach replicates real-world practices and prepares students to handle diverse client requirements, including integrating headless APIs or migrating existing data from popular e-commerce platforms.

Document Title: “Day 3- API Integration Report - [Taha Aleem]”

What to Submit: 1

A report documenting:

- ❖ API integration process.
- ❖ Adjustments made to schemas.
- ❖ Migration steps and tools used.

Screenshots of:

- ❖ API calls.
- ❖ Data successfully displayed in the frontend.
- ❖ Populated Sanity CMS fields.
- ❖ Code snippets for API integration and migration script.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TAHA\Desktop\Api\website>code .

C:\Users\TAHA\Desktop\Api\website>npm create sanity@latest

> next-e-commerce-shopco@0.1.0 npx
> create-sanity

You are logged in as tahaaleem80@gmail.com using Google
Fetching existing projects

? Create a new project or select an existing one Create new project
? Your project name: day3api
Your content will be stored in a dataset that can be public or private, depending on
whether you want to query your content with or without authentication.
The default dataset configuration has a public dataset named "production".
? Use the default dataset configuration? Yes
Creating dataset
? Would you like to add configuration files for a Sanity project in this Next.js folder? Yes
? Do you want to use TypeScript? Yes
```

```
Run `npm audit` for details.
```

```
Success! Your Sanity configuration files has been added to this project
```

```
C:\Users\TAHA\Desktop\Api\website>
```

S

Taha Aleem

day3api

30 days left in trial

T

DA

Taha Aleem

day3api

PLAN

Growth Trial

STATUS

Active

PROJECT ID

9lap05fw

Getting started

Overview

Members

Studios

Datasets

Access

Activity

Usage

Plan

API

Settings

Webhooks

CORS origins

Tokens

GROQ-powered webhooks

HTTP callbacks to a given URL triggered by changes in your content lake

[Learn more about webhooks](#)

+ Create webhook

0 of 2 webhooks (2 included in plan)

[Get more webhooks](#)

S

Taha Aleem

day3api

30 days left in trial

T

Getting started

Overview

Members

Studios

Datasets

Access

Activity

Usage

Plan

API

Settings

Webhooks

CORS origins

Tokens

CORS origins

Hosts that can connect to the project API.

+ Add CORS origin

ORIGIN	CREDENTIALS	CREATED	
http://localhost:3000	Allowed	23 minutes	
http://localhost:3333	Allowed	13 hours	

Taha Aleemday3api29 days left in trial

Getting startedOverviewMembersStudiosDatasetsAccessActivityUsagePlanAPISettings

Webhooks

CORS origins

Tokens

Tokens

Tokens are used to authenticate apps and scripts to access project data.

Name

Examples: "Employee import", "Website preview" or "PDF generator".

daythree

Permissions

Choose the access privileges for the token.

Contributor

☐ Read and write access to draft content within all datasets, with no access to project settings. (Tokens: read+write drafts)

☐ Deploy Studio (Token only)

Access to deploy Sanity Studio and GraphQL APIs to our hosted service.

Developer

☒ Read and write access to all datasets, with access to project settings for developers. (Tokens: read+write)

API

website

.next

node_modules

public

src

app

components

lib

sanity

lib

schemaTypes

TS index.tsU

TS product.tsU

TS env.tsU

TS structure.tsU

styles

types

.env.local

.gitignore

components.json

OUTLINE

TIMELINE

website > src > sanity > schemaTypes > TS product.ts > product

1import { defineType } from "sanity"

2

3export const product = defineType({

4 name: 'product',

5 title: 'Product',

6 type: 'document',

7 fields: [

8 {

9 name: 'name',

10 title: 'Name',

11 type: 'string',

12 },

13 {

14 name: 'price',

15 title: 'Price',

16 type: 'number',

17 },

18 {

19 name: 'description',

20 title: 'Description',

21 type: 'text',

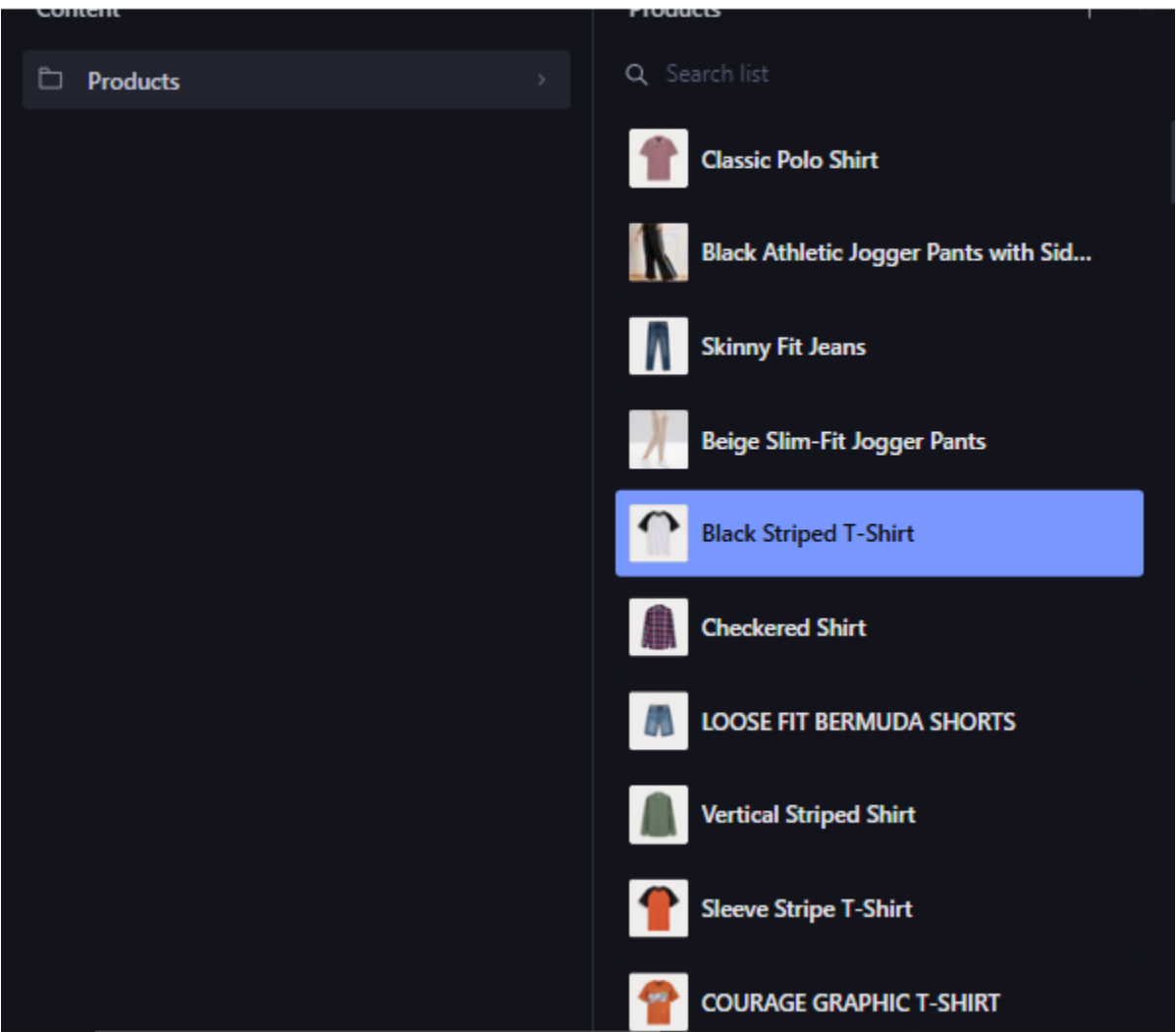
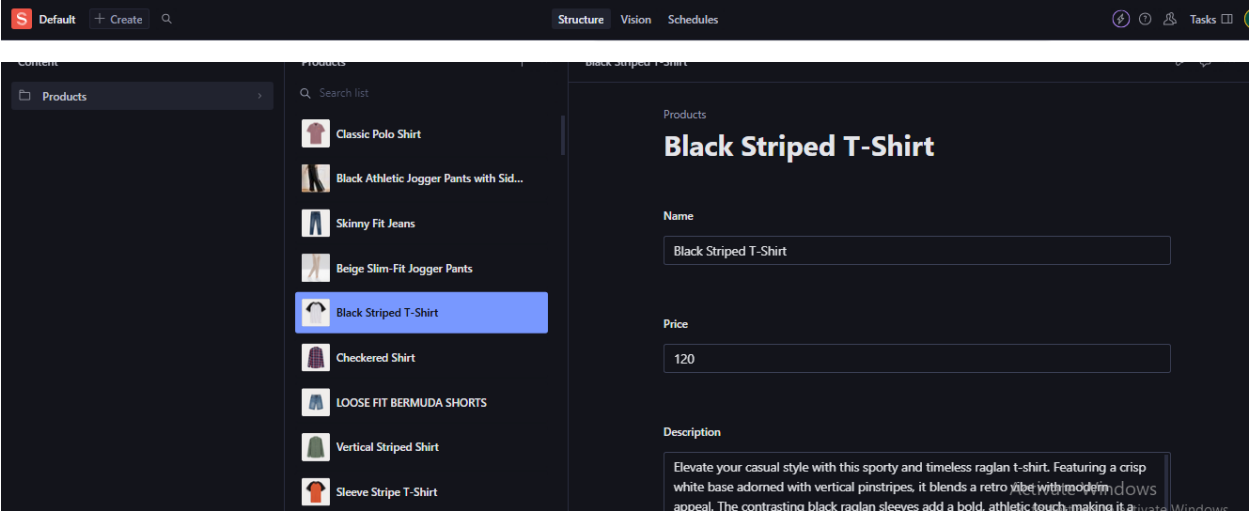
22 },

23 {

24 name: 'image',

25 title: 'Image',

26 type: 'image',



SHOP.CO

Shop

On Sale

New Arrivals

Brands

Search for products...



Sanity Studio interface showing a query for products in the 'production' dataset. The query is `*[type==products]`. The result shows 130 items, with the first item being a 'Classic Black Long Sleeve Button-Down Shirt' with a price of 190. The interface includes tabs for Structure, Vision, and Schedules, and a query URL for copying to the clipboard.

```
script > JS importData.js > client > token
1 import { createClient } from '@sanity/client';
2
3 const client = createClient({
4   projectId: '9lap05fw',
5   dataset: 'production',
6   useCdn: true,
7   apiVersion: '2025-01-13',
8   token: 'sk0J4tweIBo3w5hDntpHBKKhCjJ0cWEivm66fj70C2Wb1YFWPXLf8enSH2q7UokRZU0yCzSp8M1RuUyZrqm9l83IE7n4U21iXSnmuf3tmXdljJpIB06vTuvMwZR7s2'
9 });
10
11 async function uploadImageToSanity(imageUrl) {
12   try {
13     console.log(`Uploading image: ${imageUrl}`);
14
15     const response = await fetch(imageUrl);
16     if (!response.ok) {
17       throw new Error(`Failed to fetch image: ${imageUrl}`);
18     }
19
20     const buffer = await response.arrayBuffer();
21     const bufferImage = Buffer.from(buffer);
22
23     const asset = await client.assets.upload('image', bufferImage, {
24       filename: imageUrl.split('/').pop(),
25     });
26
27     console.log(`Image uploaded successfully: ${asset._id}`);
28     return asset._id;
29   } catch (error) {
30     console.error(error);
31   }
32 }
```

```
WEBSITE
> .next
> node_modules
> public
> script
> JS importData.js
> src
> app
> shop
> studio
> favicon.ico
> layout.tsx

package.json > ...
1 {
2   "name": "next-ecommerce-shopco",
3   "version": "0.1.0",
4   "private": true,
5   "type": "module",
6   "scripts": {
7     "dev": "next dev",
8     "build": "next build",
9     "start": "next start",
10    "lint": "next lint",
11    "import-data": "node script/importData.js"
12  },
13 }
```

