


Day 3 – API Integration Report for Cloth E-commerce market place

Prepared by Areeba

This report document the API integration process for cloths E-commerce market place including schema, API testing, Frontend integration, database migration. Moreover screenshot of API call frontend data and sanity CMS are provided

Areeba Ansari



DONALD-BLOME

PLAN

Free

STATUS

Active

PROJECT ID

b7ak1q7v

Getting started

Overview

Members

Studios

Datasets

Access

Usage

Plan

A

Webhooks

CORS origins

Tokens

GROQ-powered webhooks

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

[Learn more about webhooks](#)

What's new

Sanity Create Content Mapping, Visual Editing, and Content Releases

EXPLORER

MY-APP

src

app

layout.tsx

page.tsx

sanity

lib

schemaTypes

authorType.ts

blockContentType.ts

categoryType.ts

index.ts

postType.ts

product.ts

env.ts

page.tsx ...\products

\$.env.local

page.tsx ...\app

JS next.config.cjs

```
1 NEXT_PUBLIC_SANITY_PROJECT_ID="b7ak1q7v"
2 NEXT_PUBLIC_SANITY_DATASET="production"
3 SANITY_API_TOKEN="skfAZNS3UeQwj1QkAC8TvoHE9q8pWm0saOMHncHtMWCwa6FnCE03rKDAs6xKN1rmJHpU18BAXkOgu513xP15GjDR0g6oI4kCL77mB
711u5nDwXvp745gFDaTRhz4HjX12pxJrtQguScvQz8yOU0bhiKnL01K9qZbvVc1Y4Cd0WtHTX2aku1"
```

EXPLORER ... TS product.ts X page.tsx ...\app JS next.config.cjs

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

```
1 import { defineType } from "sanity"
2
3 export const product = defineType({
4   name: 'products',
5   title: 'products',
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 }
```

EXPLORER ... JS importData.js X page.tsx ...\app JS next.config.cjs

script > JS importData.js > uploadImageToSanity

```
1 import { createClient } from '@sanity/client';
2
3 const client = createClient({
4   projectId: 'b7ak1q7v',
5   dataset: 'production',
6   useCdn: true,
7   apiVersion: '2025-01-13',
8   token:
9     'skfAZNS3UeQwj1QkAC8TvoHE9q8pWmOsaOMHncHtMWcWa6FnCE03rKDas6xKN1rmJHpU18BAXk0gu513xP1SG
10     FDaTRhz4HjX12pxJrtQgu5cvQz8y0U0bhiKnL01K9qZbvVclY4Cd0WtHTX2aku1',
11 });
12
13 async function uploadImageToSanity(imageUrl) {
14   try {
15     console.log(`Uploading image: ${imageUrl}`);
16
17     const response = await fetch(imageUrl);
18     if (!response.ok) {
19       throw new Error(`Failed to fetch image: ${imageUrl}`);
20     }
21
22     const buffer = await response.arrayBuffer();
23     const bufferImage = Buffer.from(buffer);
24
25     const asset = await client.assets.upload('image', bufferImage, {
26       filename: imageUrl,
27     });
28
29     return asset.url;
30   } catch (error) {
31     console.error(error);
32     return null;
33   }
34 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

MY-APP

src

sanity

schemaTypes

authorType.ts

blockContentType.ts

categoryType.ts

index.ts

postType.ts

product.ts

env.ts

structure.ts

.env.local

.eslintrc.json

.gitignore

next-env.d.ts

next.config.cjs

next.config.ts

package-lock.json

package.json

postcss.config.mjs

README.md

sanity.cli.ts

package.json

```
1 {
2   "name": "my-app",
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    "postinstall": "npm rebuild sharp",
12    "import-data": "node scripts/importData.js"
13  },
14  "dependencies": {
15    "@sanity/client": "^6.25.0",
16    "@sanity/icons": "^3.5.7",
17    "@sanity/image-url": "^1.1.0",
18    "@sanity/vision": "^3.70.0",
19    "axios": "^1.7.9",
20    "dotenv": "^16.4.7"
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

GET /Casual 200 in 52ms

GET /Casual 200 in 36ms

GET /Casual 200 in 39ms

