

Day 3 - API Integration Report – [E-Furniture Store]

1.

```
D:\8-dec\my-app>cmd
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.

D:\8-dec\my-app>npm create sanity@latest -- --project rgfbd8fd --dataset production --template clean

> my-app@0.1.0 npx
> create-sanity --project rgfbd8fd --dataset production --template clean

✓ You are logged in as amberkhanum33@gmail.com using Google
✓ Fetching existing projects

? Would you like to add configuration files for a Sanity project in this Next.js folder? Yes
⚠
⚠
⚠ It looks like you are using Next.js 15 and React 19
⚠ Please read our compatibility guide.
⚠ https://www.sanity.io/help/react-19
⚠
? Do you want to use TypeScript? Yes
? Would you like an embedded Sanity Studio? Yes
? What route do you want to use for the Studio? /studio
? Select project template to use Clean project with no predefined schema types
? Would you like to add the project ID and dataset to your .env.local file? Yes
Added http://localhost:3000 to CORS origins
Running 'npm install --legacy-peer-deps --save @sanity/vision@3 sanity@3 @sanity/image-url@1 styled-components@6'
```

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

```
D:\8-dec\my-app>cmd
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.
```

2.

S

Amber Khanum

Product Database

30 days left in trial

Getting started

Overview

Members

Studios

Datasets

Access

Activity

Usage

Plan

API

Settings

Webhooks

CORS origins

Tokens

Tokens

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

NAME	PERMISSIONS	CREATED
E-commerce	Editor	20 hours

Copy the token below – this is your only chance to do so!

skgQarLNrneV27tUngvQeKDBwI4Kbv5zRUznCzOyfHvBocmuFUCgTdXdY5URCt1v11cxPFj01an38haIW87Qfyv6KYHyqiTfe4QVs7eMxjfrqXiUJWfW9oxjf1yGns2CkkGGuOzwLypj8jEiFqnUN7dapiOQLNV00Z3IopgrbVjLGHISsZMN

+ Add API token

S

Amber Khanum

Product Database

30 days left in trial

Getting started

Overview

Members

Studios

Datasets

Access

Activity

Usage

Plan

API

Settings

Webhooks

CORS origins

Tokens

Product Database

PLAN
Growth Trial

STATUS
Active

PROJECT ID
rgfbd8fd

GROQ-powered webhooks

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

Learn more about webhooks

0 of 2 webhooks
(2 included in plan)

Get more webhooks

Create webhook

There are no GROQ-powered webhooks in this project

3.

```
1 export default {
2   name: 'product',
3   type: 'document',
4   title: 'Product',
5   fields: [
6     {
7       name: 'name',
8       type: 'string',
9       title: 'Name',
10    },
11    {
12      name: 'image',
13      type: 'image',
14      title: 'Image',
15      options: {
16        hotspot: true, // Enable image cropping
17      },
18    },
19    {
20      name: 'price',
21      type: 'number',
22      title: 'Price',
23    },
24    {
25      name: 'description',
26      type: 'text',
27      title: 'Description',
28    },
29    {
30      name: 'discountPercentage',
31      type: 'number',
32      title: 'Discount Percentage',
33    },
34    {
35      name: 'isFeaturedProduct',
36      type: 'boolean',
37      title: 'Featured Product',
38    },
39  ],
40 }
```

```
1 import { createClient } from '@sanity/client';
2 import fetch from 'node-fetch';
3
4 // Initialize Sanity client
5 const client = createClient({
6   projectId: 'rgfbd9fd',
7   dataset: 'production',
8   useCdn: false, // Set to true if you want faster reads
9   apiVersion: '2025-01-13',
10   token: 'skgQar1NrneV27tUngvQekD6W4tkbV5ZRUZnCzoyfhvBocmuFUCgtdXdy5URctiv1cxPFj01an38h1l87Qfyv6KvHyq1tfe4Qv57efxcJfrqX1UJWf69oxjflyGns2CkkgGu0ZwlypJ8JEf',
11 });
12
13 // Function to upload an image to Sanity
14 async function uploadImageToSanity(imageUrl) {
15   try {
16     console.log('Uploading image: ${imageUrl}');
17
18     const response = await fetch(imageUrl);
19     if (!response.ok) {
20       throw new Error('Failed to fetch image: ${imageUrl}');
21     }
22
23     const buffer = await response.arrayBuffer();
24     const bufferImage = Buffer.from(buffer);
25
26     const asset = await client.assets.upload('image', bufferImage, {
27       filename: imageUrl.split('/').pop(),
28     });
29
30     console.log('Image uploaded successfully: ${asset._id}');
31     return asset._id;
32   } catch (error) {
33     console.error('Failed to upload image:', imageUrl, error);
34     return null;
35   }
36 }
37
38 // Function to upload a single product to Sanity
39 async function uploadProduct(product) {
```

4.

The image displays two screenshots of the Sanity Studio interface. The top screenshot shows the 'Luxury Flower Bed' product detail view. The bottom screenshot shows the query editor with a query and its result.

Top Screenshot: Product Detail View

The interface shows a sidebar with a list of products. The 'Luxury Flower Bed' is selected and highlighted. The main content area displays the product details:

- Product:** Luxury Flower Bed
- Name:** Luxury Flower Bed
- Image:** A photograph of a bedroom with a large bed, a nightstand, and a lamp.
- Price:** (Field is empty)
- Published:** 17 hr. ago
- Buttons:** Publish, ...

Bottom Screenshot: Query Editor

The interface shows the query editor with the following details:

- DATASET:** production
- API VERSION:** Other
- CUSTOM API VERSION:** v2025-01-27
- PERSPECTIVE:** raw
- QUERY URL (COPY TO CLIPBOARD):** [https://rghbd8fd.api.sanity.io/v2025-01-27/data/query/production?query=*\[type==product\]](https://rghbd8fd.api.sanity.io/v2025-01-27/data/query/production?query=*[type==product])
- QUERY:**

```
1 * [type==product]
```
- PARAMS:**

```
1 {  
2     
3 }
```
- RESULT:**

```
[_] 52 items  
  0: {_  
    _createdAt: 2025-01-27T07:29:40Z  
    members: [_] 1 item  
      0: pp1rVw00U  
      _rev: i5ry3u0sKkTEgpoPKfsy4R  
      _type: system.group  
      _id: .groups.administrator  
      _updatedAt: 2025-01-27T08:00:26Z  
    grants: [_] 2 items  
      0: {_  
        filter: _id in path("****")  
        permissions: [_] 4 items  
      1: {_  
        permissions: [_] 4 items  
        filter: (_id in path("****")) && _id in path("drafts.****")  
      1: {_  
        _updatedAt: 2025-01-27T08:00:26Z  
        grants: [] 0 items  
        _createdAt: 2025-01-27T07:29:40Z  
        members: [] 0 items  
        _rev: i5ry3u0sKkTEgpoPKfsy4R  
        _type: system.group
```
- Execution:** 13ms End-to-end: 319ms
- Buttons:** Fetch, Listen, Save result as, JSON, CSV