

# Day 3 - API Integration Report – AS foods

## API Integration Process

1. **Overview:** The integration connects an external API providing **foods** and **chefs** data to a Sanity CMS project.

2. **Steps Taken:**

**Environment Setup:**

- Used **dotenv** to load environment variables from
- **.env.local**. Key variables included:
- NEXT\_PUBLIC\_SANITY\_PROJECT\_ID
- NEXT\_PUBLIC\_SANITY\_DATASET
- SANITY\_TOKEN

**Sanity Client Creation:**

- Used **@sanity/client** to establish a connection to the Sanity project. Configured the client with the project ID, dataset, API version, and
- authentication token.

**Data Fetching:**

- Made concurrent API calls using **axios** to fetch food and chef data. Endpoints accessed:

`https://sanity-nextjs-rouge.vercel.app/api/foods`

`https://sanity-nextjs-rouge.vercel.app/api/chefs`

**Data Processing:**

- Iterated through the fetched data.  
Uploaded images to Sanity's asset library using the
- `client.assets.upload()` method.

3. **Sanity Document Creation:**

- Transformed fetched data into Sanity-compatible document structures. Uploaded each document using **client.create()**.

**Error Handling:**

- Implemented try-catch blocks for API calls and Sanity operations.
- Logged errors for debugging.

## Adjustments Made to Schemas:

1. **Chef Schema:**

- Added the following fields:  
**Position:** String field to specify the chef's role.  
**Experience:** Number field for years of experience. **Specialty:** String field for culinary specialization. **Available:** Boolean field for availability status.
- Enabled **hotspot** option for the image field to allow flexible cropping.

## 2. Food Schema:

- Key fields included:
- **Category:** String field for classification (e.g., Burgers, Drinks). **Price:** Number field for current price. **Original Price:** Optional number field for discounted price. **Tags:** Array field with a tag layout for categorization.
- **Available:** Boolean field for stock status.
- Enabled **hotspot** for the image field to support adaptable displays.

## Migration Steps and Tools Used

### 1. Migration Steps:

#### Preparation:

- Analyzed API data structure and matched it with existing Sanity schemas. Created Sanity schemas for **food** and **chef** with necessary fields.
- **Data Import Script:**
  - Built **import-data.mjs** script to automate data fetching, processing, and uploading.
  - Utilized **axios** for API interaction and **@sanity/client** for CMS operations.
- **Image Handling:**
  - Downloaded images using **axios** with **responseType: 'arraybuffer'**. Uploaded images to Sanity, storing asset references in document fields.
- **Document Creation:**
  - Mapped API data fields to Sanity schema fields.
  - Ensured fallback values for optional or missing fields.

### 2. Tools Used:

- **Node.js Modules:**
  - **axios** for HTTP requests. **dotenv** for environment variable management.
  - **@sanity/client** for CMS interactions.
- **Sanity Features:**
  - Asset management for image uploads.
  - API versioning for consistent schema support.
- **Utilities:**
  - **fileURLToPath** and **path** for resolving file paths.


## API calls (Foods):

```
[
  {
    name: 'Chicken Chup',
    category: 'Appetizer',
    price: 12,
    originalPrice: 15,
    image: { _type: 'image', asset: [Object] },
    description: 'Crispy fried chicken bites served with dipping sauce.',
    available: true,
    tags: [ 'Sell', 'Crispy' ]
  },
  {
    name: 'Fresh Lime',
    category: 'Drink',
    price: 38,
    originalPrice: 45,
    image: { _type: 'image', asset: [Object] },
    description: 'Refreshing fresh lime drink made with natural ingredients.',
    available: true,
    tags: [ 'Healthy', 'Popular' ]
  },
  {
    available: true,
    tags: [ 'Sell', 'Sweet' ],
    name: 'Chocolate Muffin',
    category: 'Dessert',
    price: 28,
    originalPrice: 30,
    image: { _type: 'image', asset: [Object] },
    description: 'Soft and rich chocolate muffin topped with chocolate chips.'
  },
  {
    name: 'Country Burger',
    category: 'Sandwich',
    price: 45,
    originalPrice: 50,
    image: { _type: 'image', asset: [Object] },
    description: 'Classic country-style burger served with fries.',
    available: true,
    tags: [ 'Recommended' ]
  },
  {
    image: { _type: 'image', asset: [Object] },
    description: 'Juicy beef burger with fresh lettuce, tomatoes, and cheese.'
    available: true,
    tags: [ 'Popular' ],
    name: 'Burger',
    category: 'Sandwich',
    price: 21,
    originalPrice: 45
  },
]
```


Data successfully displayed in the frontend:

Food data:


Sort By : Newest Show : Default




**Cheese Butter**  
\$10.00




**Drink**  
\$23.00 ~~\$45.00~~




**Chocolate Muffin**  
\$45.00




**Pizza**  
\$42.00



**Sandwiches**  
\$18.00 ~~\$25.00~~

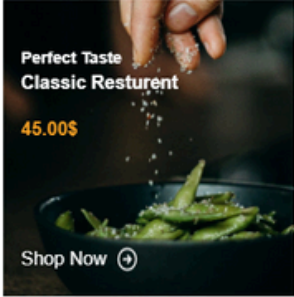



**Chicken Chup**  
\$25.00

Search Product 

**Category**

- ☐ Sandwiches
- ☐ Burger
- ☐ Chiken Chup
- ☐ Drink
- ☐ Pizza
- ☐ Thi
- ☐ Non Veg
- ☐ Uncategorized



**Perfect Taste  
Classic Resturent**  
45.00\$  
Shop Now 



**Filter By Price**

From 0\$ to \$8000 Filter

Populated sanity CMS fields:







Food :

**Content**

-  Chef
-  **Food**

**Food** + ...

Search list

-  **Chicken Chup**
-  **Pizza**
-  **Country Burger**
-  **Burger**
-  **Chocolate Muffin**
-  **Fresh Lime**

## Final check:

API Understanding	Schema Validation	Data Migration	API Integration in Next.js	
✓	✓	✓		