

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.
 - **Sanity Document Creation:**
 - Transformed fetched data into Sanity-compatible document structures.
 - Uploaded each document using **client.create()**.
3. **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.

Screenshot of:

API calls (Chef):

```
[
  {
    position: 'Head Chef',
    experience: 12,
    specialty: 'Italian Cuisine',
    image: { _type: 'image', asset: [Object] },
    description: 'Expert in crafting authentic Italian dishes and pastries.',
    available: true,
    name: 'Tahmina Rumi'
  },
  {
    description: 'Renowned for creating perfectly grilled meats and vegetables.',
    available: true,
    name: 'M. Mohammad',
    position: 'Grill Master',
    experience: 10,
    specialty: 'Grilled Dishes',
    image: { asset: [Object], _type: 'image' }
  },
  {
    image: { _type: 'image', asset: [Object] },
    description: 'Expert in international cuisines and menu planning.',
    available: true,
    name: 'Bisnu Devgon',
    position: 'Executive Chef',
    experience: 20,
    specialty: 'Global Cuisine'
  },
  {
    position: 'Chef de Cuisine',
    experience: 18,
    specialty: 'Seafood Specialties',
    image: { _type: 'image', asset: [Object] },
    description: 'Master of crafting exquisite seafood dishes with unique flavors.',
    available: true,
    name: 'William Rumi'
  },
  {
    available: true,
    name: 'Jorina Begum',
    position: 'Sous Chef',
    experience: 8,
    specialty: 'Pastry and Desserts',
    image: { _type: 'image', asset: [Object] },
    description: 'Specializes in creative pastries and dessert innovations.'
  },
  {
    position: 'Pastry Chef',
    experience: 15,
    specialty: 'Dessert Innovations',
    image: { _type: 'image', asset: [Object] },
    description: 'Known for creating innovative and delicious pastries and desserts.',
    available: true,
    name: 'Aminul Haque'
  }
]
```

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:



Chicken Chup

\$12

[View Product](#)



Fresh Lime

\$38

[View Product](#)



Chocolate Muffin

\$28

[View Product](#)



Country Burger

\$45

[View Product](#)



Burger

\$21

[View Product](#)



Pizza

\$43

[View Product](#)

[Activate Wire](#)
[Go to Settings to](#)

Chef data:



Tahmina Rumi

Expert in crafting authentic Italian dishes and pastries.
Italian Cuisine
Head Chef
Available
12



M. Mohammad

Renowned for creating perfectly grilled meats and...
Grilled Dishes
Grill Master
Available
10



Bisnu Devgon

Expert in international cuisines and menu planning.
Global Cuisine
Executive Chef
Available
20



William Rumi

Master of crafting exquisite seafood dishes with unique...
Seafood Specialties
Chef de Cuisine
Available
18



Jorina Begum

Specializes in creative pastries and dessert innovations.
Pastry and Desserts
Sous Chef
Available
8

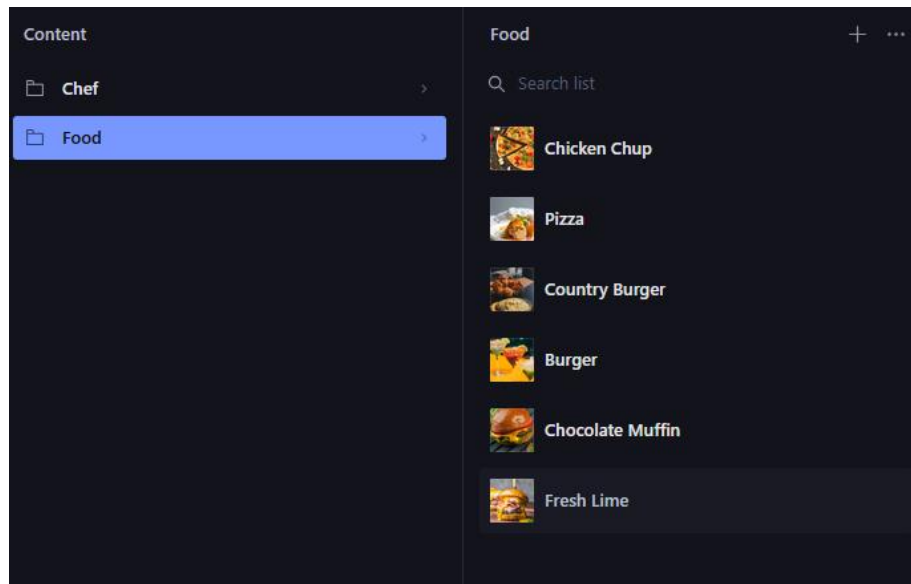


Munna Kathy

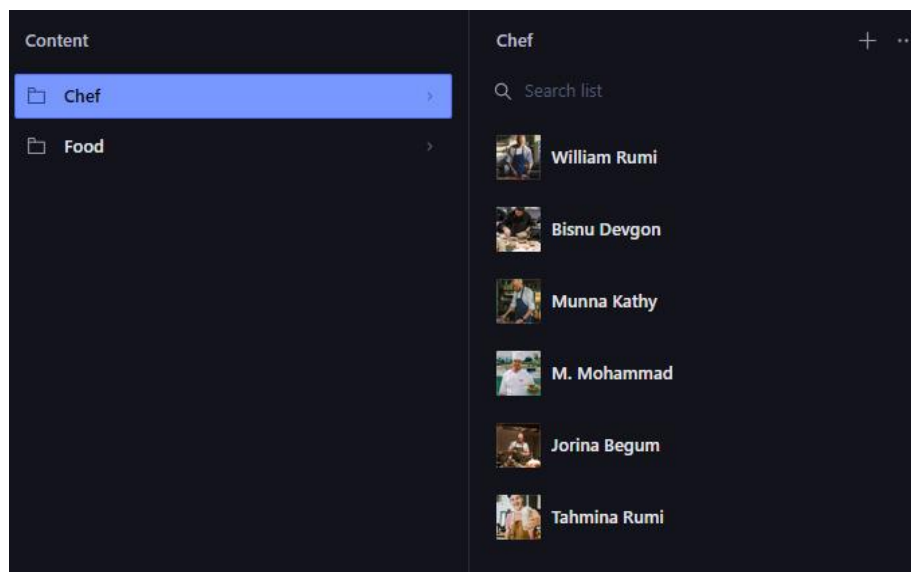
Pioneer in Asian fusion dishes blending traditional...
Asian Fusion
Culinary Instructor
Available
15

Populated sanity CMS fields:

Food :



Chef:



Code Snippets for API integration and migration scripts:

```
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.SANITY_TOKEN,
  apiVersion: '2021-08-31',
});

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;
  }
}

async function importData() {
  try {
    console.log('Fetching food, chef data from API...');
    const $Promise = [];
    $Promise.push(
      axios.get('https://sanity-nextjs-rouge.vercel.app/api/foods')
    );
    $Promise.push(
      axios.get('https://sanity-nextjs-rouge.vercel.app/api/chefs')
    );
    const [foodsResponse, chefsResponse] = await Promise.all($Promise);
    const foods = foodsResponse.data;
    const chefs = chefsResponse.data;
    for (const food of foods) {
      console.log(`Processing food: ${food.name}`);
      let imageRef = null;
      if (food.image) {
        imageRef = await uploadImageToSanity(food.image);
      }
      const sanityFood = {
        _type: 'food',
        name: food.name,
        category: food.category || null,
        price: food.price,
      };
    }
  }
}
```

```

category: food.category || null,
price: food.price,
originalPrice: food.originalPrice || null,
tags: food.tags || [],
description: food.description || '',
available: food.available !== undefined ? food.available : true,
image: imageRef
? {
  _type: 'image',
  asset: {
    _type: 'reference',
    _ref: imageRef,
  },
}
: undefined,
});
console.log('Uploading food to Sanity:', sanityFood.name);
const result = await client.create(sanityFood);
console.log('Food uploaded successfully: ${result._id}');
for (const chef of chefs) {
  console.log('Processing chef: ${chef.name}');
  let imageRef = null;
  if (chef.image) {
    imageRef = await uploadImageToSanity(chef.image);
  }
  const sanityChef = {
    _type: 'chef',
    name: chef.name,
    position: chef.position || null,
    experience: chef.experience || 0,
    specialty: chef.specialty || '',
    description: chef.description || '',
    available: chef.available !== undefined ? chef.available : true,
    image: imageRef
    ? {
      _type: 'image',
      asset: {
        _type: 'reference',
        _ref: imageRef,
      },
    }
    : undefined,
  });
  console.log('Uploading chef to Sanity:', sanityChef.name);
  const result = await client.create(sanityChef);
  console.log('Chef uploaded successfully: ${result._id}');
}
console.log('Data import completed successfully!');
} catch (error) {
  console.error('Error importing data:', error);
}
}
importData();

```

Final check:

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