



DAY 3 - API INTEGRATION AND DATA MIGRATION



Prepared by Ameen Alam

First published: PKT 11:08 AM, January 15, 2025

Document Revision Information

Version	Date	Amendment	Author
1.0	11:08 AM, January 15, 2025	Initial release of Day1	Ameen Alam
1.1	03:00 PM, January 15, 2025	Added Examples in Day1	Ameen Alam
2.0	01:00 PM, January 16, 2025	Day 2 Release	Ameen Alam
2.1	02:00 PM, January 17, 2025	Self-Validation Checklist	Ameen Alam
3.0	04:00 PM, January 17, 2025	Day 3 Release	Ameen Alam
3.1	08:30 PM, January 17, 2025	Minor correction in link	Ameen Alam

Table of Contents

Day 3 - API Integration and Data Migration.....	2
API Overview:	2
Provided APIs:.....	2
Insert API Documentation Here:	2
Using External Data Sources (Optional):	4
Steps for Day 3:	4
1. Understand the Provided API:	4
2. Validate and Adjust Your Schema:	5
3. Data Migration Options:.....	5
4. API Integration in Next.js:.....	5
Error Handling Tips:.....	6
Expected Output:	6
Submission Requirements:.....	6
Best Practices:	7
Day 3 Checklist:	8
Self-Validation Checklist:.....	8
FAQs:.....	8

Day 3 - API Integration and Data Migration

Objective:

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 eCommerce platforms.

Key Learning Outcomes:

1. Understand how to integrate APIs into your Next.js project.
2. Learn to migrate data from APIs into Sanity CMS.
3. Explore how to use existing data from **eCommerce platforms like Shopify, Magento, WooCommerce, WordPress, Salesforce, Custom Backend, Sanity, Mock APIs or others.**
4. Develop skills to handle and validate schemas, ensuring alignment with data sources.

API Overview:

Provided APIs:

Below are API references for templates 0 to 9. These APIs are read only and meant for guidance in schema validation and data migration. Students have the freedom to:

- Use these APIs to populate their Sanity CMS.
- Import data manually.
- Use other APIs or existing data sources to customize their marketplace.

Insert API Documentation Here:

Practice Template – Sir Ali Jawwad

<https://alijawwad001.atlassian.net/wiki/external/OWVkJZTc1NjExMjU1NDljNmFkMGJhZTUxYmVmMWNmMzc>

Template 0 - Sir Ahmed Raza and Bilal Fareed

- https://docs.google.com/document/d/1tAtI-WtSBY7XbgxbvhtJq40d_IsfOBtcivmLLao06g/edit?usp=sharing

Template 1 – Sir Fahad Khan and Sir Ali Aftab

- API Documentation: <https://github.com/developer-hammad-rehman/template1/blob/main/README.md>
- Migration Script: <https://github.com/developer-hammad-rehman/template1/blob/main/importData.js>
- Schema : <https://github.com/developer-hammad-rehman/template1/blob/main/src/sanity/schemaTypes/products.ts>
- GitHub repo: <https://github.com/developer-hammad-rehman/template1/tree/main>

Template 2 - Sir Bilal Muhammad Khan and Sir Aneeq

- API <https://hackathon-apis.vercel.app/api/products>
- Sanity Schemas:
 - <https://github.com/bilalmk/hackathon-template02/blob/main/schema/category.ts>
 - <https://github.com/bilalmk/hackathon-template02/blob/main/schema/product.ts>
- Migration Script: <https://github.com/bilalmk/hackathon-template02>

Template 3 - Sir Okasha

- migration script: <https://github.com/OkashaTanoli/template-03-api/blob/master/scripts/data-migration.mjs>
- api: <https://template-03-api.vercel.app/api/products>
- sanity schema : <https://github.com/OkashaTanoli/template-03-api/blob/master/src/sanity/schemaTypes/products.ts>

Template 4 – Sir Anas and Sir Ubaid Ur Rehman

- API URL: <https://next-ecommerce-template-4.vercel.app/api/product>
- Schema: <https://github.com/anasseth/next-ecommerce-template-4/blob/master/src/sanity/schemaTypes/product.ts>
- Documentation: <https://github.com/anasseth/next-ecommerce-template-4/blob/master/README.md>

Template 5 - Sir Muhammad Bilal

- We will follow Same as Template 6 API docs

Template 6 - Sir Fahad Khan and Sir Hamza Alvi

- API Url : <https://template6-six.vercel.app/api/products>
- Schema: <https://github.com/developer-hammad-rehman/template6/blob/main/src/sanity/schemaTypes/product.ts>

- Migration: <https://github.com/developer-hammad-rehman/template6/blob/main/importData.js>

Template 7 - Sir Asharib Ali and Sir Ahmed Raza Shaikh

- <https://sanity-nextjs-application.vercel.app/api/hackathon/template7>
- Sanity Schema: <https://github.com/AsharibAli/sanity-nextjs/blob/main/sanity/schemaTypes/cars.ts>
- Migration Script: <https://github.com/AsharibAli/sanity-nextjs/blob/main/scripts/importTemplate7Data.mjs>

Template 8 - Sir Hamzah Syed

https://docs.google.com/document/d/1tg3wdRvcGnEcyVRyzVpXzl8tbPClD_Z9mfFrq1vhkns/edit?tab=t.0

Template 9 - Sir Mubashir

- <https://github.com/mubashirimtiaz/sanity-nextjs>

Using External Data Sources (Optional):

Students may also integrate data from popular eCommerce platforms and custom backend using their headless APIs or data export features:

- **Shopify:** Export product data via CSV or access APIs.
- **Magento:** Use REST or GraphQL APIs for product and customer data.
- **WooCommerce:** Access REST APIs or export data manually.
- **WordPress:** Use REST APIs for content or WooCommerce-specific data.
- **Salesforce:** Use APIs for customer relationship and product data.
- **Already Developed Backend or Data Management platform**
- **CMS (Sanity and others)**

Steps for Day 3:

1. Understand the Provided API:

- Review the API documentation for your assigned template.
- Identify key endpoints such as:
 - Product listings (e.g., /products)
 - Categories (e.g., /categories)
 - Order history (if applicable)

2. Validate and Adjust Your Schema:

- Compare your existing Sanity CMS schema (created on Day 2) with the API data structure.
- Adjust field names, types, and relationships to ensure compatibility.
Example:
API Field: product_title
Schema Field: name
Adjust your schema to match or map fields during migration.

3. Data Migration Options:

To streamline your migration process, here are three methods you can use. Ensure you validate all data and document your migration steps.

Using the Provided API:

Write scripts to fetch and transform data from the API (You can find the right script in the above-mentioned template, and you can use any template data depending on your flexibility.).

Manual Import:

- Export data from the API or another source as JSON or CSV.
- Use Sanity's built-in import tools to upload the data.
- This method is ideal for small datasets or when learning basic import methods.

Using External Platform APIs:

- Fetch data from platforms like Shopify or WooCommerce.
- Map fields to your Sanity schema.
- Follow a similar script-based migration process, ensuring data transformation aligns with schema requirements.

Key Best Practices:

- Always back up your Sanity project before importing large datasets.
 - Test the imported data for accuracy and alignment with your schema.
 - Include validation checks in your migration scripts to handle errors gracefully.
- Students have multiple options to migrate data:

4. API Integration in Next.js:

This step involves integrating APIs into your frontend project. Ensure robust error handling and modular coding for scalability.

- **Step 1: Create Utility Functions**
- **Step 2: Render Data in Components**
- **Step 3: Test API Integration**
 - Use tools like Postman or your browser's developer tools to test API endpoints.
 - Log responses to ensure data consistency and identify any issues.

Error Handling Tips:

- Log errors in a centralized file for debugging.
- Display user-friendly error messages in your UI.
- Use fallback data or skeleton loaders for a better user experience.
- Create API utility functions in your project to fetch data from the provided API or your Sanity CMS.
- Test the integration by rendering API data on your frontend.

Expected Output:

1. Sanity CMS populated with imported data from:
 - Provided API.
 - External sources.
 - Manually uploaded data.
2. Functional API integration in Next.js displaying:
 - Product listings.
 - Categories.
 - Other relevant data.

Submission Requirements:

- **Document Title:** "Day 3 - API Integration Report - [Your Marketplace Name]"
- **What to Submit:**
 - 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 scripts.

Best Practices:

1. Use .env files for storing sensitive data like API keys securely.
2. Follow clean coding practices:
 - Use descriptive variable names.
 - Modularize functions for reusability.
 - Add comments to explain complex logic.
3. Validate all data during migration:
 - Use schemas to check field types and constraints.
 - Log discrepancies for further investigation.
4. Document every step thoroughly:
 - Include screenshots, scripts, and testing notes.
 - Maintain a changelog for any schema adjustments.
5. Use version control:
 - Commit changes frequently with meaningful messages.
 - Tag significant milestones in your repository.
6. Test API integration thoroughly:
 - Handle edge cases like empty responses or invalid data.
 - Use tools like Postman for endpoint validation.
7. Peer review:
 - Share your code and documentation with peers for feedback.
 - Incorporate suggested improvements to enhance quality.

Day 3 Checklist:

Self-Validation Checklist:

API Understanding:

- ✓ or ✗

Schema Validation:

- ✓ or ✗

Data Migration:

- ✓ or ✗

API Integration in Next.js:

- ✓ or ✗

Submission Preparation:

- ✓ or ✗

FAQs:

1. Can we use other APIs or data sources instead of the provided API?

Yes, you are free to use any API or existing data source that aligns with your marketplace's requirements. The provided APIs are for reference and practice.

2. How do we handle schema mismatches between the API and our Sanity CMS?

Identify the differences in field names or data types and adjust your schema in Sanity CMS to match or map the data correctly during migration.

3. What if we don't have prior experience with API integration?

Start with simple API calls using tools like Postman to understand the responses. Follow the example scripts provided in this document for guidance.

4. Can we manually add data to Sanity instead of using migration scripts?

Yes, you can manually add data to Sanity CMS if you find it easier or if the data size is small. However, learning migration scripting will help you handle larger datasets in the future.

5. Do we need to create new APIs for this hackathon?

No, you only need to integrate and use existing APIs. However, you may customize your frontend logic to process the API responses effectively.

6. Is API integration mandatory for all templates?

Yes, API integration is required to fetch or migrate data. The method (manual or automated) is your choice.

7. What should we submit as part of Day 3?

Your submission should include:

- A report documenting API integration and data migration steps.
- Screenshots of data displayed on your frontend and populated in Sanity CMS.
- Code snippets of your integration and migration scripts.

8. How can we debug API errors?

Use browser developer tools or tools like Postman to test API endpoints. Add error handling in your scripts to log specific issues and adjust requests accordingly.

9. Can we use different data schemas for different templates?

Yes, your schemas should reflect the unique requirements of your assigned template. Ensure they align with your chosen data source or API.

10. What skills will we gain from this exercise?

You will learn:

- API integration techniques.
- Data migration best practices.
- Schema validation and customization.
- Practical skills for building scalable marketplaces using headless CMS systems like Sanity.