

Plane for Day 3 – API Integration and Data Migration...

2. Chart Layout:

Section	Details	Flow/Examples
Objective	API integration with Next.js and data migration into Sanity CMS.	1. API methods. 2. Schema compatibility.
Key Learning Outcomes	1. API Integration. 2. Data Migration. 3. Explore e Commerce platforms. 4. Validate schemas with real-world data.	Example: Fetch data from Shopify or Woo Commerce.
Integration Steps	Step 1: Fetch data from an API. Step 2: Validate it with Sanity schema. Step 3: Migrate into Sanity CMS using APIs.	fetch() + Sanity schema validation
Provided APIs	Templates (0-9) for examples. APIs are read-only for guidance in schema creation and data validation.	Import API templates into your project.
Final Goal	Build a functional backend marketplace with customized data sources and a seamless Next.js-Sanity integration.	Example: A mini marketplace showing imported products.

Key Outcomes:

1. API Integration in Next.js:
 - o Learn how to fetch data from external APIs.
 - o Handle API responses and errors effectively (step-by-step process).
2. Data Migration:
 - o Validate fetched data with Sanity schemas.

- o Migrate external data into Sanity CMS without losing compatibility.
- 3. Explore e Commerce Platforms:
 - o Shopify, Woo Commerce, Salesforce, and others.
- 4. Schema Validation:
 - o Build schemas that match real-world data structures and ensure smooth API integration.

Workflow/Steps for Integration:

1. Step 1: Fetch API Data
 - o Use the given APIs (Templates 0-9).
 - o Write a fetch function in Next.js to retrieve data.
 - o Validate fetched data by testing its format.
2. Step 2: Sanity CMS Schema Setup
 - o Create schemas in Sanity CMS (e.g., Product, User, Order schemas).
 - o Match API data fields with your schema fields.
 - o Example: API data for a product may have:
 - Title Product Name
 - Price Cost
 - Description Details
3. Step 3: Data Migration

- o Import validated API data into Sanity CMS.
- o Ensure data is saved in the correct format within Sanity.
- o Test migrations by fetching and displaying data in Next.js.

4. Step 4: Final Testing

- o Fetch migrated data from Sanity CMS.
- o Display data dynamically in Next.js.
- o Example: Render a product list or marketplace page.

Tools/Resources Required:

- Next.js for Frontend.
- Sanity CMS for Backend.
- Provided APIs for data population.
- Postman (optional) for testing API responses.

API Overview:

- Provided APIs: Templates 0-9.
- These APIs are read-only and serve as guidance for validation and migration.
- Freedom to:
 - 1. Populate Sanity CMS directly.
 - 2. Manually import data.

Conclusion:

By the end of this module, we'll:

- Successfully integrate APIs into your Next.js project.
- Migrate data into Sanity CMS with properly validated schemas.
- Build a functional marketplace backend.