Day 3 - API Integration and Data Migration

Objective

The focus of Day 3 is to integrate APIs and migrate data into Sanity CMS to build a functional marketplace backend. By the end of this session, you will:

What is an API integration? Types, benefits and alternatives

- Understand API integration in Next.js projects.
- Learn to migrate data from APIs into Sanity CMS.
- Use data from eCommerce platforms such as Shopify, Magento, WooCommerce, WordPress, Salesforce, or mock APIs.
- Develop skills to validate schemas, ensuring compatibility with data sources.

This approach mirrors real-world scenarios, preparing you to handle diverse client requirements, including headless API integrations and data migrations.

Key Learning Outcomes

1. Integrate APIs into your Next.js project.
2. Migrate data from APIs into Sanity CMS.
3. Use existing eCommerce platform data to populate your backend.
4. Validate schemas to align with data sources.

Template 6 API Overview

Provided API for Template 6

- **API URL**: https://template6-six.vercel.app/api/products
- **Sanity Schema**: Product Schema
- **Migration Script**: Data Migration Script

Steps for Day 3

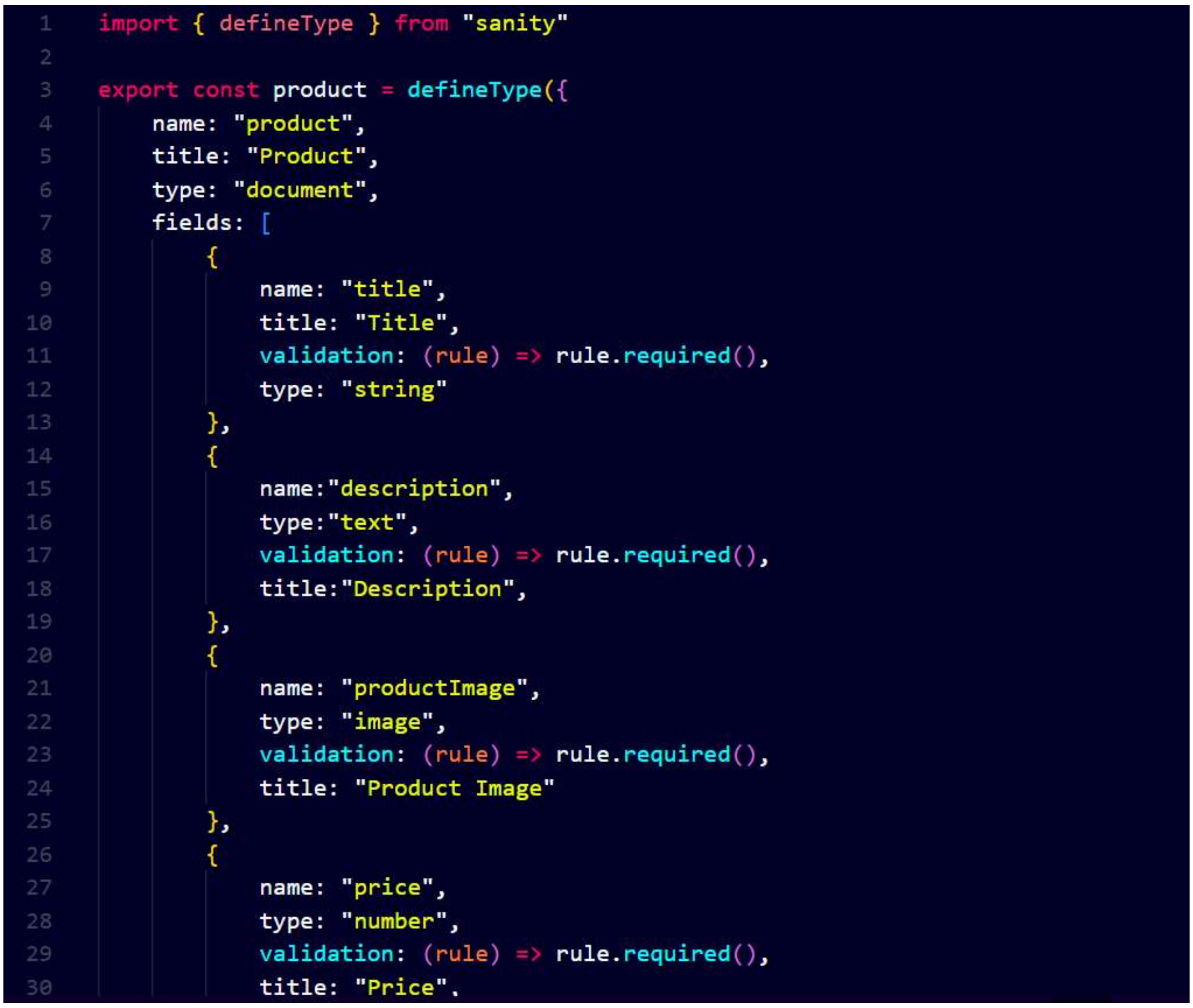
1. Understand the Provided API

- **Review API Documentation**: Study the API endpoints provided for Template 6.
- **Key Endpoints**:
  - `/products`: Provides product listings.
  - Other endpoints (if applicable).

2. Validate and Adjust Your Schema

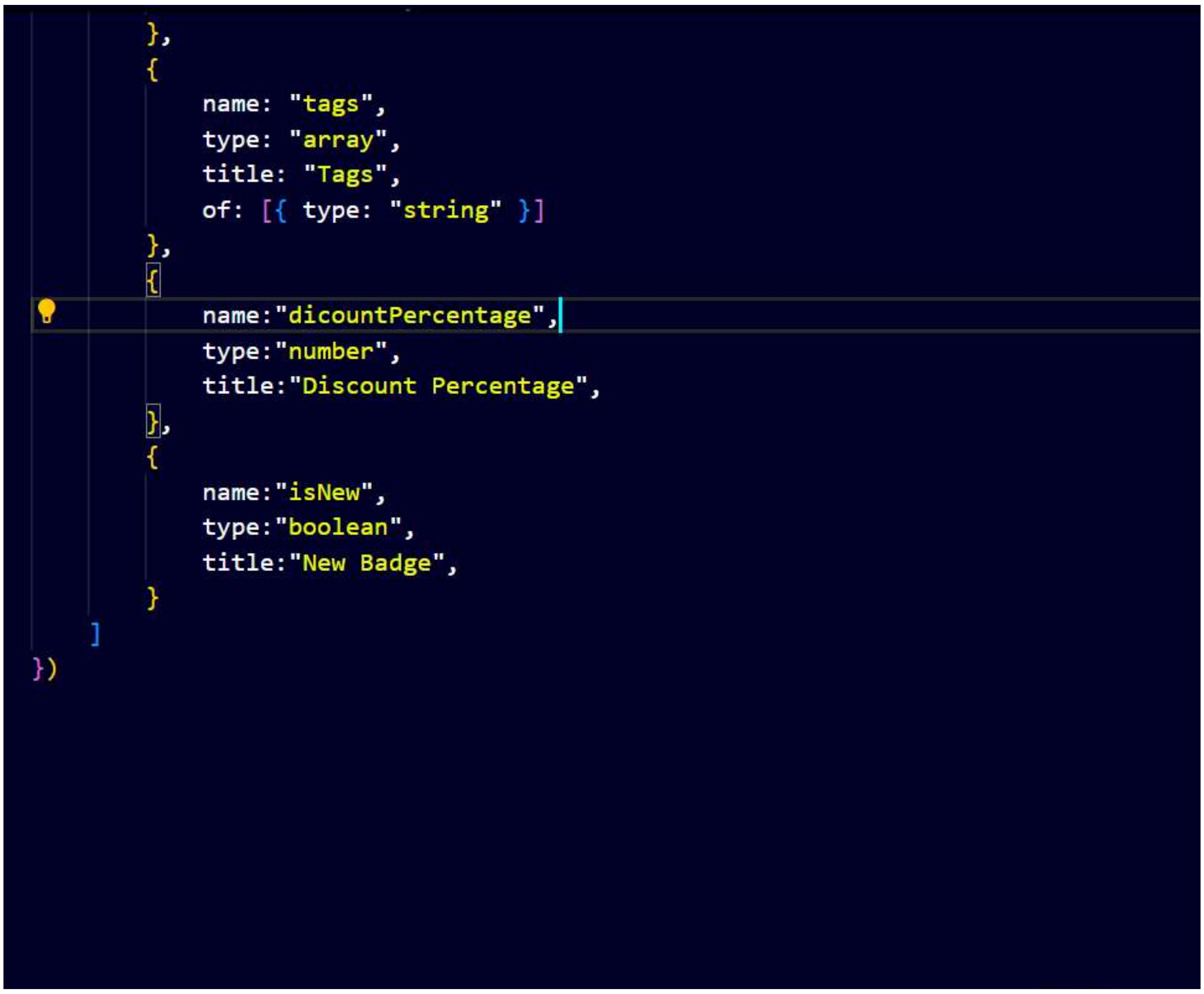
- Compare your existing Sanity CMS schema with the API data structure.
- Adjust field names, data types, and relationships in your schema to ensure compatibility.

Example Schema Adjustment:



```
import { defineType } from "sanity"

export const product = defineType({
    name: "product",
    title: "Product",
    type: "document",
    fields: [
        {
            name: "title",
            title: "Title",
            validation: (rule) => rule.required(),
            type: "string"
        },
        {
            name:"description",
            type:"text",
            validation: (rule) => rule.required(),
            title:"Description",
        },
        {
            name: "productImage",
            type: "image",
            validation: (rule) => rule.required(),
            title: "Product Image"
        },
        {
            name: "price",
            type: "number",
            validation: (rule) => rule.required(),
            title: "Price",
```



```
        },
        {
            name: "tags",
            type: "array",
            title: "Tags",
            of: [{ type: "string" }]
        },
        {
            name:"dicountPercentage",
            type:"number",
            title:"Discount Percentage",
        },
        {
            name:"isNew",
            type:"boolean",
            title:"New Badge",
        }
    ]
})
```

3. Data Migration Options

A. Using the Provided API

1. Write a script to fetch data from the API.
2. Transform the data to align with your Sanity schema.
3. Import the data into Sanity CMS using a migration script.

B. Manual Import

1. Export data from the API as JSON or CSV.
2. Use Sanity's built-in import tools to upload the data.

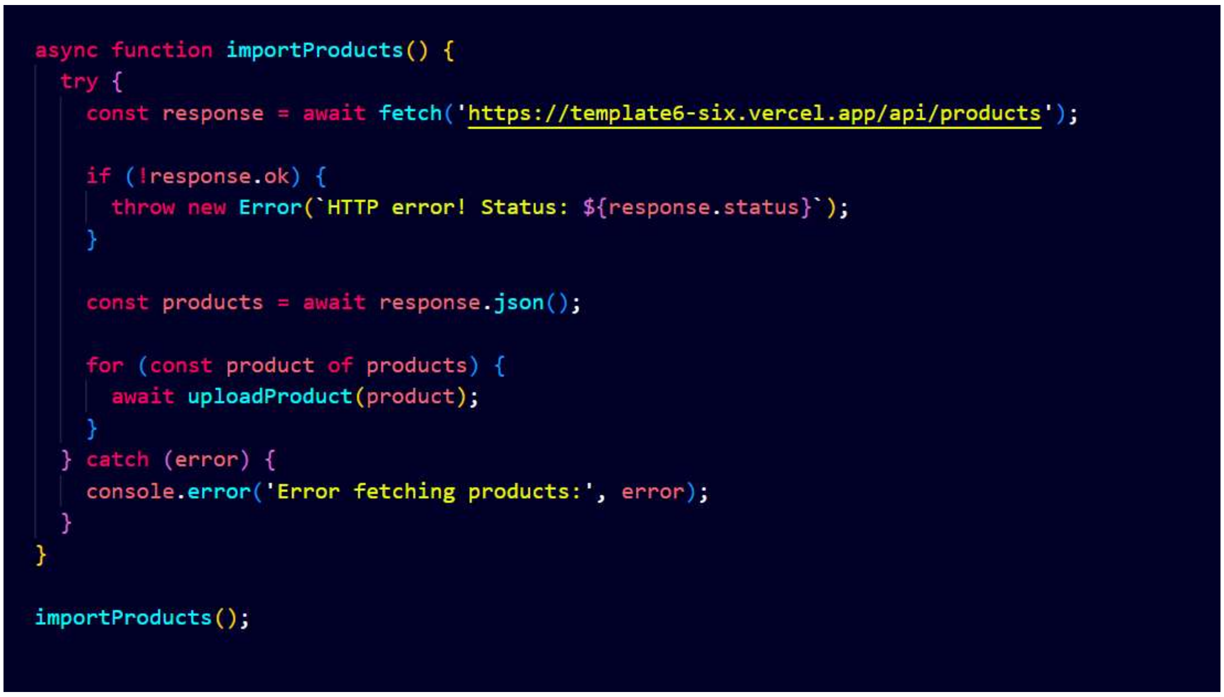
C. Using External Platforms

Fetch data from platforms like Shopify, WooCommerce, or others. Map fields to your Sanity schema and migrate using similar scripts.

4. API Integration in Next.js

1. **Create Utility Functions**: Write reusable functions to fetch data from the API.
2. **Render Data in Components**: Use the fetched data to populate UI components.
3. **Test API Integration**: Use tools like Postman or browser developer tools to verify API responses.

Sample Utility Function:



```
async function importProducts() {
  try {
    const response = await fetch('https://template6-six.vercel.app/api/products');

    if (!response.ok) {
      throw new Error(`HTTP error! Status: ${response.status}`);
    }

    const products = await response.json();

    for (const product of products) {
      await uploadProduct(product);
    }
  } catch (error) {
    console.error('Error fetching products:', error);
  }
}

importProducts();
```

Error Handling Tips

- **Log Errors**: Maintain a centralized log for debugging.
- **User-Friendly UI**: Display meaningful error messages.
- **Fallback Data**: Use skeleton loaders or default data.

Expected Output

1. Sanity CMS populated with imported data from:
    - Provided API.

- External sources (if applicable).
- Manually uploaded data.

2. Functional API integration in Next.js:
   - Product listings displayed on the frontend.
   - Categories and other relevant data rendered dynamically.

Submission Requirements

Document Title:

"Day 3 - API Integration Report - [Sanitay and Electric Store]"

What to Submit:

1. A report detailing:
   - API integration process.
   - Adjustments made to schemas.
   - Migration steps and tools used.
2. Screenshots of:
   - API calls.
   - Data successfully displayed on the frontend.
   - Populated Sanity CMS fields.
3. Code snippets for API integration and migration scripts.

Best Practices

1. **Secure Sensitive Data**:
   - Use `.env` files to store API keys.
2. **Clean Code**:
   - Use descriptive variable names.
   - Modularize reusable functions.
   - Add comments to explain complex logic.
3. **Data Validation**:
   - Validate data types and constraints using schemas.
   - Log discrepancies for debugging.
4. **Thorough Documentation**:
   - Include screenshots, scripts, and testing notes.
   - Maintain a changelog for schema adjustments.
5. **Version Control**:

  - Commit changes frequently.
  - Tag significant milestones.
6. **Comprehensive Testing**:
   - Handle edge cases (e.g., empty responses).
   - Use tools like Postman for endpoint validation.
7. **Peer Review**:
   - Share your code for feedback.

FAQs

1. **Can we use other APIs or data sources?**
   - Yes, you are free to use APIs or data sources that align with your marketplace's requirements. The provided APIs are for practice and reference.
2. **How do we handle schema mismatches?**
   - Identify differences in field names or data types and adjust your schema accordingly.
3. **What if I'm new to API integration?**
   - Start with simple API calls using tools like Postman. Use provided scripts as a guide.
4. **Can I manually add data to Sanity CMS?**
   - Yes, manual addition is an option for smaller datasets or learning purposes.

Day 3 Checklist

- **API Understanding**: [Yes]
- **Schema Validation**: [Yes]
- **Data Migration**: [Yes]
- **API Integration in Next.js**: [Yes]