**API Integration Process**

**API Selection:**

The API integration process involved using a mock API endpoint to fetch product data

Instead of using the provided API, the custom mock API was selected to fetch the data. The

decision was made due to the flexibility and control over the mock API data, making it easier

to test various scenarios.

**Data Fetching:**

The data fetching process is performed using the **Axios** library to send GET requests to the

mock API. The retrieved product data is then processed before being integrated into the

Sanity CMS.

**Sanity Integration:**

The integration with **Sanity CMS** was achieved through:

**Sanity Client Setup:** The project utilizes a custom Sanity client that includes the necessary

credentials (project ID, dataset, and token).

**Image Handling:** A custom function was created to handle image uploads, fetching the image

from the URL, converting it to a buffer, and uploading it to Sanity as an asset.

**Product Data Upload:** Each product's data was mapped into a schema and uploaded using

the Sanity client. References to assets and inventories were also handled by fetching or

creating them as required.

**Key Steps:**

 **Fetching Data** from the mock API.

 **Uploading Images** to Sanity.

 **Creating Inventory References** for each product.

 **Inserting Product Data** into Sanity using the client.

**Migration Steps and Tools Used**

 **Migration Overview:**

The migration process aimed at importing external data (from the mock API) into the Sanity

CMS while maintaining data integrity and ensuring the correct schema references were

established.

 **Steps Involved:**

1. **Sanity Project Setup:** A Sanity project was created, and the necessary credentials (project ID,

dataset, and token) were obtained and integrated into the client setup.

2. **Axios for Data Fetching:** The Axios library was used to fetch data from the mock API. A

function was created to handle the fetching of data and error handling.

3. **Schema Creation and Adjustment:** The schemas were first created in Sanity, including

product, inventory, order, and shipment schemas. Adjustments to references between products,

inventories, and images were made to ensure proper relationships.

4. **Data Import:** The data from the mock API was iterated over, and for each product:

o Images were uploaded to Sanity using the custom image upload function.

o Inventory references were either fetched or created.

o Product data was mapped to the schema and uploaded to Sanity using the

client.createOrReplace() method.

5. **Final Verification:** After the migration, the data was verified in Sanity for consistency, and

the correct references were confirmed.

 **Tools and Technologies Used:**

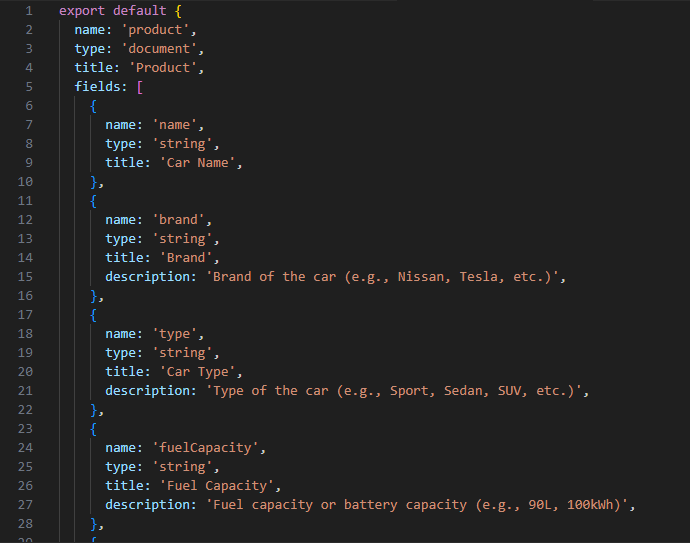
 **Axios:** For fetching data from the mock API.

 **Sanity Client:** For interacting with Sanity CMS, including fetching and uploading data.

 **Sanity Migration Script:** Custom migration script used for uploading data and images to

Sanity, handling inventory references, and creating/updating product entries.

 **Node.js:** The entire migration process was handled using Node.js scripts.



A screen shot of a computer program

Description automatically generated



A screenshot of a car sale

Description automatically generated