Day 3 - API Integration and Data Migration

API Overview:

API (Application Programming Interface) allows different systems to communicate and exchange data. On Day 3, the focus is on integrating external APIs into your application and migrating data efficiently.

Provided APIs:

- List of APIs provided for integration during Day 3.

Insert API Documentation Here:

Here, you will document all the APIs provided for your integration, including endpoints, request parameters, responses, authentication methods, etc.

Using External Data Sources (Optional):

Incorporating external data sources like databases or third-party APIs is optional but can enhance the functionality of your application. Ensure the integration is smooth and data is fetched accurately.

Steps for Day 3:

1. Understand the Provided API:

- Review API documentation.
- Identify the endpoints, required parameters, and response formats.
- Determine the type of authentication (if any) and set it up.

2. Validate and Adjust Your Schema:

- Analyze your existing data schema.
- Modify your schema (e.g., database structure or API models) to match the API responses.
- Ensure data consistency and integrity across your system.

3. Data Migration Options:

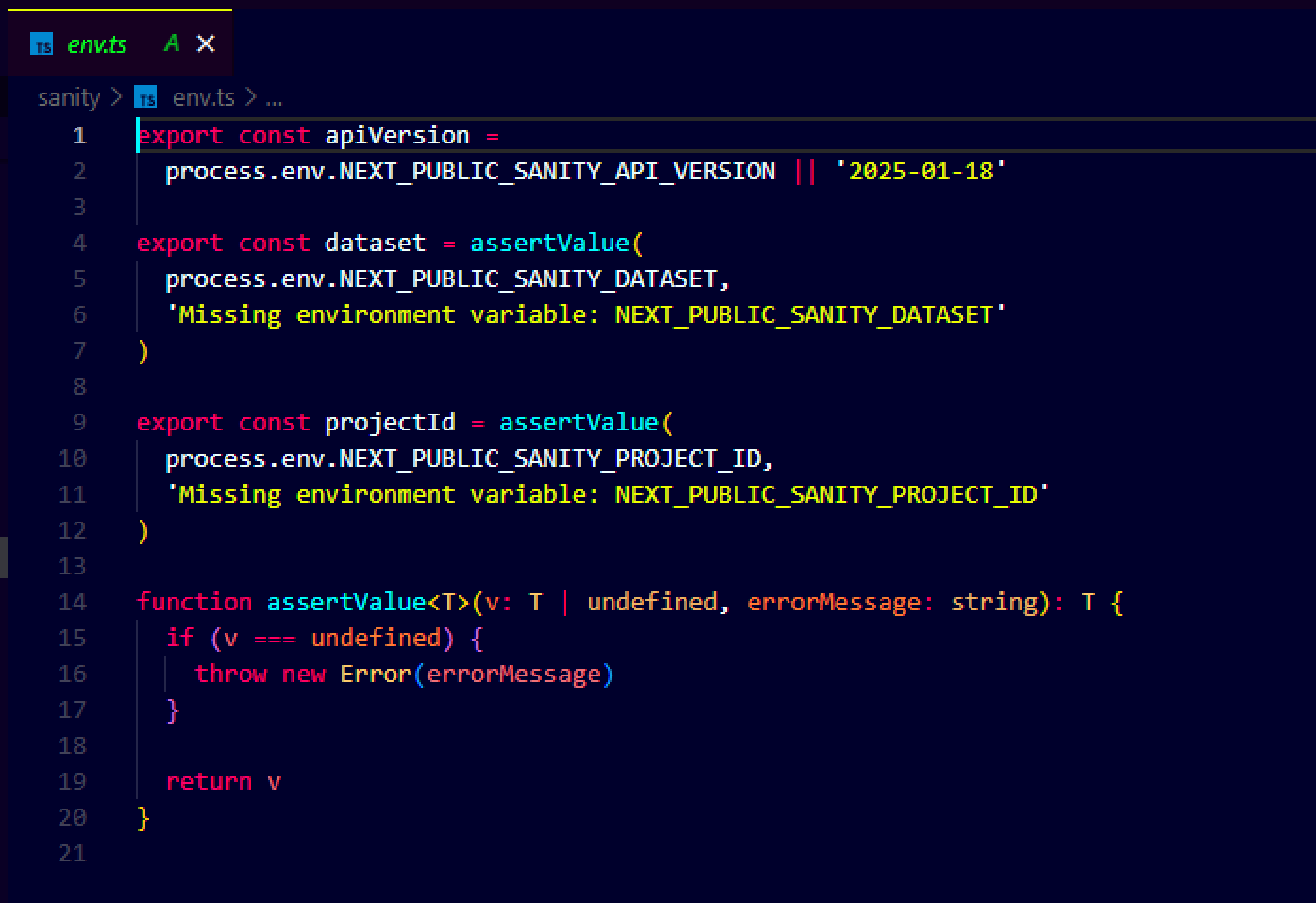
- **Data Import/Export:** Choose between CSV, JSON, or XML formats for transferring data.
- **Bulk Migration:** For large datasets, ensure batch processing is handled efficiently.
- **Real-time Migration:** Consider using webhooks or API callbacks for real-time data migration.

4. API Integration in Next.js:

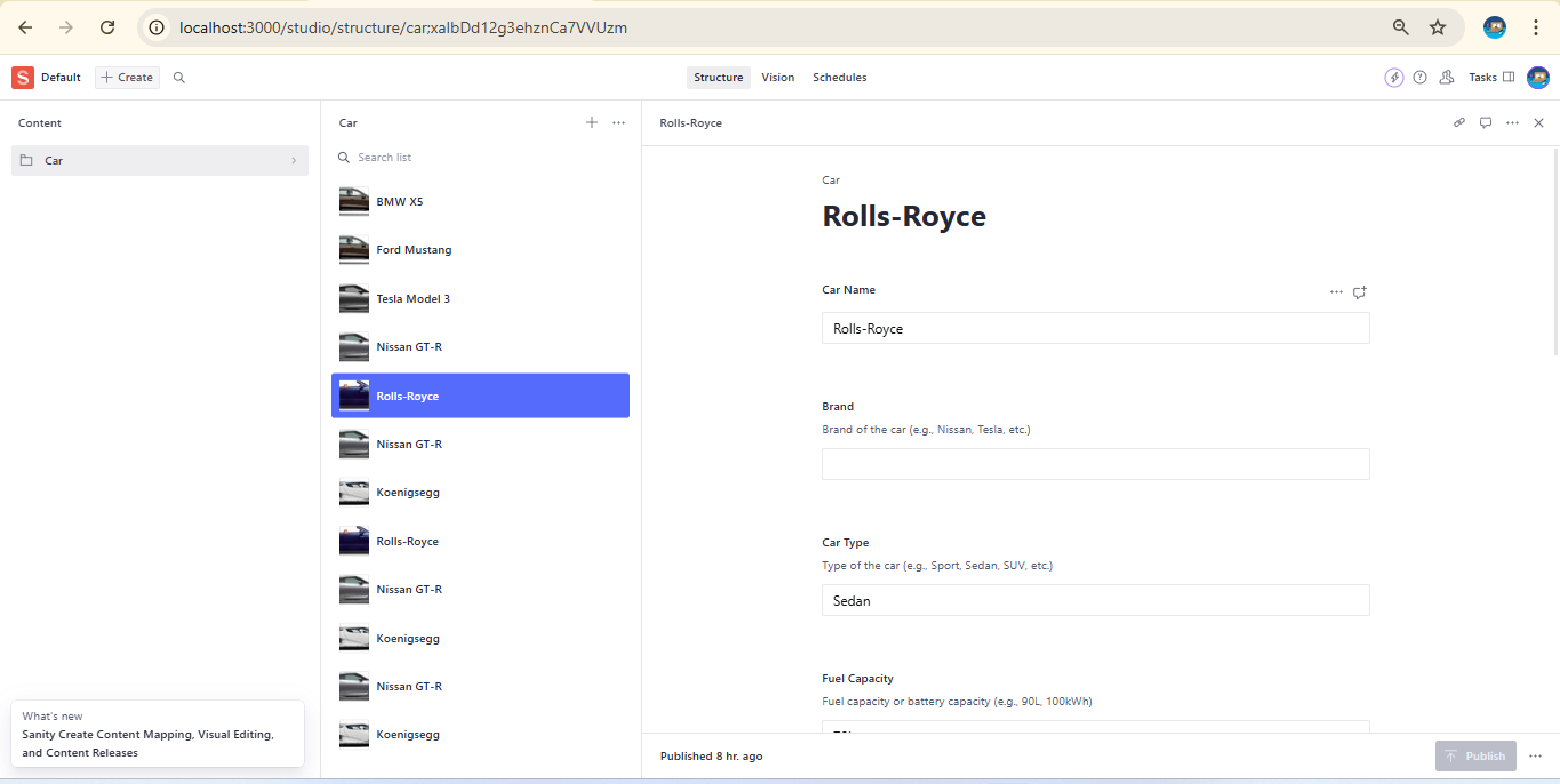
- Install necessary packages (like Axios, fetch, etc.) for API calls.
- Use `getServerSideProps` or `getStaticProps` for server-side fetching.
- Handle asynchronous data fetching in React components.
- Map API data to your component states and render them dynamically.

Error Handling Tips:

- Always check for error responses (4xx or 5xx status codes).
- Implement try-catch blocks in async functions to handle unexpected failures.
- Provide clear error messages to users and log errors for debugging.



```typescript
export const apiVersion =
  process.env.NEXT_PUBLIC_SANITY_API_VERSION || '2025-01-18'

export const dataset = assertValue(
  process.env.NEXT_PUBLIC_SANITY_DATASET,
  'Missing environment variable: NEXT_PUBLIC_SANITY_DATASET'
)

export const projectId = assertValue(
  process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  'Missing environment variable: NEXT_PUBLIC_SANITY_PROJECT_ID'
)

function assertValue<T>(v: T | undefined, errorMessage: string): T {
  if (v === undefined) {
    throw new Error(errorMessage)
  }

  return v
}

```



localhost:3000/studio/structure/car;xalbDd12g3ehznCa7VVUzm

Default | + Create | Structure | Vision | Schedules | Tasks

Content

Car

Car

Search list

- BMW X5
- Ford Mustang
- Tesla Model 3
- Nissan GT-R
- Rolls-Royce
- Nissan GT-R
- Koenigsegg
- Rolls-Royce
- Nissan GT-R
- Koenigsegg
- Nissan GT-R
- Koenigsegg

Rolls-Royce

Car

Rolls-Royce

Car Name

Rolls-Royce

Brand

Brand of the car (e.g., Nissan, Tesla, etc.)

Car Type

Type of the car (e.g., Sport, Sedan, SUV, etc.)

Sedan

Fuel Capacity

Fuel capacity or battery capacity (e.g., 90L, 100kWh)

Published 8 hr. ago | Publish

What's new
Sanity Create Content Mapping, Visual Editing, and Content Releases