

# Rental E-Comeerce webiste

## Day:3 API Inetgeration & data migartion

Rollno: 405127

### **Data Migration Process:**

- I used the API which was provide me .
- Setup sanity according the API.
- Setup schemas field naming conventions.
- Use the script to migrate data from API to sanity
- Set the sainty API token to create data

Here is script

```

import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

// Load environment variables from .env.local
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '.env.local') });

// Create Sanity client
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2021-08-31'
});

// Tabnine | Edit | Test | Explain | Document
async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop()
    });
    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
}

```

```

async function importData() {
  try {
    console.log('Fetching car data from API...');

    // API endpoint containing car data
    const response = await axios.get('https://sanity-nextjs-application.vercel.app/api/hackathon/template7');
    const cars = response.data;

    console.log(`Fetched ${cars.length} cars`);

    for (const car of cars) {
      console.log(`Processing car: ${car.name}`);

      let imageRef = null;
      if (car.image_url) {
        imageRef = await uploadImageToSanity(car.image_url);
      }

      const sanityCar = {
        _type: 'products',
        id: `product_${car.id}`,
        name: car.name,
        brand: car.brand || null,
        type: car.type,
        fuelCapacity: car.fuel_capacity,
        transmission: car.transmission,
        seatingCapacity: car.seating_capacity,
        pricePerDay: car.price_per_day,
        originalPrice: car.original_price || null,
        tags: car.tags || [],
        image: imageRef ? {
          _type: 'image',
          asset: {
            _type: 'reference',
            ref: imageRef,

```

```

async function importData() {
  const sanityCar = {
    id: `product_${car.id}`,
    name: car.name,
    brand: car.brand || null,
    type: car.type,
    fuelCapacity: car.fuel_capacity,
    transmission: car.transmission,
    seatingCapacity: car.seating_capacity,
    pricePerDay: car.price_per_day,
    originalPrice: car.original_price || null,
    tags: car.tags || [],
    image: imageRef ? {
      _type: 'image',
      asset: {
        _type: 'reference',
        _ref: imageRef,
      },
    } : undefined,
  };

  console.log('Uploading car to Sanity:', sanityCar.name);
  const result = await client.create(sanityCar);
  console.log(`Car uploaded successfully: ${result._id}`);
}

console.log('Data import completed successfully!');
} catch (error) {
  console.error('Error importing data:', error);
}
}

importData();

```

## How to setup sanity?

```

import { defineType, defineField } from "sanity";

export default defineType({
  name: 'products',
  type: 'document',
  title: 'Products',
  fields: [
    defineField({
      name: 'id',
      type: 'string',
      title: 'Car id',
    }),
    defineField({
      name: 'name',
      type: 'string',
      title: 'Car Name',
    }),
    defineField({
      name: 'description',
      type: 'string',
      title: 'Description',
    }),
    defineField({
      name: 'available',
      type: 'boolean',
      title: 'Is car available',
    }),
    defineField({
      name: 'brand',
      type: 'string',
      title: 'Brand',
      description: 'Brand of the car (e.g., Nissan, Tesla, etc.)',
    }),
    defineField({
      name: 'type',
      type: 'string',
      title: 'Car Type',
      description: 'Type of the car (e.g., Sport, Sedan, SUV, etc.)',
    }),
    defineField({
      name: 'fuelCapacity',
      type: 'string',
      title: 'Fuel Capacity',
      description: 'Fuel capacity or battery capacity (e.g., 90L, 100kWh)',
    }),
    defineField({
      name: 'transmission',
      type: 'string',
      title: 'Transmission',
      description: 'Transmission type (e.g., Automatic, Manual, Semi-Automatic)',
    }),
    defineField({
      name: 'seatingCapacity',
      type: 'string',
    })
  ]
});

```

I used this process to migrate in sanity with the help of grooq query data was fetched by and display in project success fully

Popular

[View All](#)

**Rolls-Royce**  
Sedan



70L Manual 4 People

**12,000** PKR/day

[Rent Now](#)

**Koenigsegg**  
Sport



90L Manual 2 People

**12,000** PKR/day

[Rent Now](#)

**Nissan GT-R**  
Sport



80L Manual 2 People

**35,000** PKR/day

[Rent Now](#)

**Nissan GT-R**  
Sport



80L Manual 2 People

**21,000** PKR/day

[Rent Now](#)

AKBAR FAROOQ