

# Day-3 API Integration Report - Shop.co

## API Intergration Process :

Fetch Data from API

Code Snippet:

```
async function importProducts() {
  try {
    const response = await fetch('https://template1-neon-nu.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);
  }
}
```

## Adjustments In Schemas:

add “isNew” in schema .

## Final Schema :

```
import { defineType } from "sanity"

export default defineType({
  name: 'products',
  title: 'Products',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Name',
      type: 'string',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
      options: {
        list: [
          {title: 'T-Shirt', value: 'tshirt'},
          {title: 'Short', value: 'short'},
          {title: 'Jeans', value: 'jeans'},
          {title: 'Hoodie', value: 'hoodie'},
          {title: 'Skirt', value: 'skirt'},
        ],
      },
    },
    {
      name: 'discountPercent',
      title: 'Discount Percent',
      type: 'number',
    },
    {
      name: 'isNew',
      type: 'boolean',
      title: 'New',
    },
    {
      name: 'colors',
      title: 'Colors',
      type: 'array',
      of: [
        {type: 'string'}
      ],
    },
    {
      name: 'sizes',
      title: 'Sizes',
      type: 'array',
      of: [
        {type: 'string'}
      ],
    },
  ],
})
```

# Migration steps:

## Environment Variable Setup:

1- Install Sanity and its dependencies. Generate an API token in Sanity with the "Editor" role, and create a .env file to securely store the projectId, token, and production values.

2- To handle images, update the next.config.mjs file by adding the image domain "cdn.sanity.io".

```
const nextConfig = {
  reactStrictMode: true,
  images: {
    domains: ["cdn.sanity.io"],
  },
};

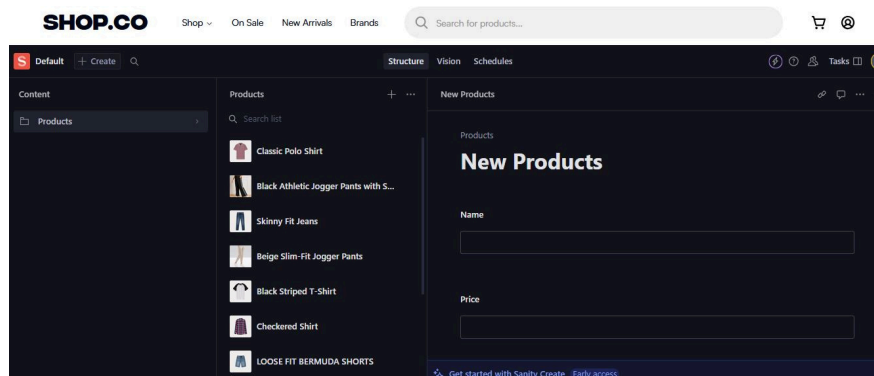
export default nextConfig;
```

3- Add the required import command in the package.json file.

```
"import-data": "node scripts/importData.mjs"
```

4- Execute the command npm run import-data.

5- All products will be fetched from the API and stored in Sanity.



# API Call Output:

console.log(products):

```
next-server (v14.2.18)
},
{
  _id: 'pNdFo0jzUh1upuRBLXqu56',
  color: null,
  size: null,
  category: 'jeans',
  discountPercent: 0,
  image_url: 'https://cdn.sanity.io/images/59nmwrlv/production/1780596a854c7fafc6b9e10558b3da006ad6a39b-195x258.jpg',
  name: 'Black Athletic Jogger Pants with Side Stripes',
  description: 'These sporty black jogger pants offer a perfect blend of comfort and style. Featuring a relaxed fit with an elastic waistband and adjustable drawstring, they are ideal for both casual wear and light athletic activities. The distinctive white side stripes add a stylish, modern touch that enhances their sporty appeal. With deep pockets for convenience, these pants are versatile enough to pair with a variety of tops, from t-shirts to hoodies. Comfortable and easy to wear, these jogger pants are a great choice for a laid-back, active look.',
  price: 180,
  isNew: true
},
{
  description: 'These classic black jeans offer a versatile and timeless addition to any wardrobe. Featuring a straight t-leg cut, they provide a comfortable, flattering fit for various body types. Crafted from durable denim, these jeans are both stylish and practical, perfect for both casual outings and semi-formal occasions. The clean, dark black color adds a sleek and sophisticated touch, making them easy to pair with any top, from t-shirts to shirts or hoodies. With their classic design and premium fit, these black jeans are a must-have staple for any wardrobe.',
  discountPercent: 16,
  _id: 'qQpkQmFIZ2DSQkhvXKuSRk',
  name: 'Classic Black Straight-Leg Jeans',
  price: 170,
  isNew: false,
  color: null,
  size: null,
  category: 'jeans',
  image_url: 'https://cdn.sanity.io/images/59nmwrlv/production/b2264d466c3c96f8ae3f1192a2f73ac313221576-183x275.jpg'
},
{
  discountPercent: 0,
  image_url: 'https://cdn.sanity.io/images/59nmwrlv/production/18be6c1b158aacd1475323d7544d2bf5640a5b4d-295x298.png',
}
```

# Code Snippets:

## API Integration :

```
async function importProducts() {  
  try {  
    const response = await fetch('https://template1-neon-nu.vercel.app/api/products');  
  
    if (!response.ok) {  
      throw new Error(`HTTP error! Status: ${response.status}`);  
    }  
  
    const products = await response.json();  
  }  
}
```

## Migration Script:

```

async function uploadImageToSanity(imageId) {
  if (!response.ok) {
    throw new Error('Failed to fetch image: ${imageId}');
  }

  const buffer = await response.arrayBuffer();
  const bufferImage = Buffer.from(buffer);

  const asset = await client.assets.upload('image', bufferImage, {
    filename: imageId.split('/').pop(),
  });

  console.log('Image uploaded successfully: ${asset._id}');
  return asset._id;
} catch (error) {
  console.error('Failed to upload image:', imageId, error);
  return null;
}
}

async function uploadProduct(product) {
  try {
    const imageId = await uploadImageToSanity(product.imageId);

    if (imageId) {
      const document = {
        _type: 'products',
        name: product.name,
        description: product.description,
        price: product.price,
        image: {
          _type: 'image',
          asset: {
            _ref: imageId,
          },
        },
        category: product.category,
        discountPercent: product.discountPercent,
        isNew: product.isNew,
        colors: product.colors,
        sizes: product.sizes
      };

      const createdProduct = await client.create(document);
      console.log('Product ${product.name} uploaded successfully:', createdProduct);
    } else {
      console.log('Product ${product.name} skipped due to image upload failure.');
```

# Fetch Data Display on Frontend:

\$150

Discount: 10%

Category: hoodie

Available Colors:

Sizes:


style Pair it with jeans, joggers, or shorts to create a standout look!

\$145

Category: tshirt


Available Colors:

Sizes:



Vertical Striped Shirt

This product is a stylish vertical striped shirt, featuring alternating green and darker green stripes for a sophisticated



Checkered Shirt

This men's plaid shirt combines timeless style and comfort, featuring a bold red, navy, and white checkered pattern. With a

## Final Check List

Api Understanding:    ✓

Schema Validation :    ✓

**Data Migration:** ✓

**Api Integration in Next.js:** ✓

**Submission Preparation :** ✓

Wahiba Afzal.