

Hackathon 3 - Day 3 Task

Sanity API Data Fetch and Import

1. Sanity Client Configuration

This section initializes the Sanity client using environment variables.

The following parameters are configured:

- `projectId`: Project ID from Sanity dashboard.
- `dataset`: Dataset name (e.g., 'production').
- `token`: API token for authentication.
- `apiVersion`: Specifies the version of the Sanity API to use.
- `useCdn`: False to ensure fresh data is fetched.

```
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2025-01-15',
  useCdn: false,
});
```

2. Image Upload Function

The `uploadImageToSanity` function fetches an image from a URL and uploads it to Sanity as an asset.

- Uses `axios` to fetch the image data.
- Converts the data to a buffer for uploading.
- Returns the asset ID after a successful upload.

```
async function uploadImageToSanity(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(),
  });
  return asset._id;
}
```

Hackathon 3 - Day 3 Task

Sanity API Data Fetch and Import

1. Sanity Client Configuration

This section initializes the Sanity client using environment variables.

The following parameters are configured:

- `projectId`: Project ID from Sanity dashboard.
- `dataset`: Dataset name (e.g., 'production').
- `token`: API token for authentication.

- `apiVersion`: Specifies the version of the Sanity API to use.
- `useCdn`: False to ensure fresh data is fetched.

```
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2025-01-15',
  useCdn: false,
});
```

2. Image Upload Function

The `uploadImageToSanity` function fetches an image from a URL and uploads it to Sanity as an asset.

- Uses `axios` to fetch the image data.
- Converts the data to a buffer for uploading.
- Returns the asset ID after a successful upload.

```
async function uploadImageToSanity(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(),
  });
  return asset._id;
}
```

3. Data Import Function

The `importData` function fetches product data from an API and imports it into Sanity.

- Fetches product details using `axios`.
- Processes each product to upload its image (if available) and create a document in Sanity.
- Logs progress and handles errors.

```
async function importData() {
  const response = await
  axios.get("https://next-ecommerce-template-4.vercel.app/api/product");
  const products = response.data.products;
```

```

for (const item of products) {
  let imageRef = null;
  if (item.imagePath) {
    imageRef = await uploadImageToSanity(item.imagePath);
  }
  const sanityItem = {
    _type: 'product',
    name: item.name,
    category: item.category || null,

    price: item.price, description: item.description || "",
    discountPercentage: item.discountPercentage || 0,
    stockLevel: item.stockLevel || 0,
    isFeaturedProduct: item.isFeaturedProduct,
    image: imageRef ? { _type: 'image', asset: { _type: 'reference', _ref: imageRef } } :
    undefined,
  };
  const result = await client.create(sanityItem);
}
}

```

4. Environment Variables

Environment variables used for configuration:

- `NEXT_PUBLIC_SANITY_PROJECT_ID`: Project ID from Sanity dashboard.
- `NEXT_PUBLIC_SANITY_DATASET`: Dataset name (e.g., 'production').
- `SANITY_API_TOKEN`: API token with necessary permissions.

NEXT_PUBLIC_SANITY_PROJECT_ID="6an6k5ne"

NEXT_PUBLIC_SANITY_DATASET="production"

SANITY_API_TOKEN="skgFV6V3kmdaBnllaHRAI83CazLkEQxXa3XEN2E0JLihLeJtDLOGQQ98CMY2l38X8RQow6

lxz0Rk7fZDpHeVOnW84uhUu9lyt0E8R9MAJOjfD5utcsxbYIG7Pxc8oRVGgTC9ECGUWt5fAakZRhEW7VzxeMx46n

2lACi8glcAHrGoUWFygksS"

Product | Default

localhost:3000/studio/structure/product

S

Default

+ Create

Q

Structure

Vision

Schedules

Blog

Posts

Categories


Authors


Product


Product


+ ...


Q Search list


 Cantilever Chair


 Luxury Flower Shell Sofa Chair


 Nordic Net Red Chair


 Sobuy Blue Folding Chair Wooden Pad...


 Varmora Plastic Chair Solid


 Matilda Velvet Chair – Pink

 Diondre Chair - Tuft Button - Acrylic Le...

 Leisure Sofa Chair Set

 Liberty Wood 63' Floating Entertainme...

 Futuristic Sleek Modern Chair

 Replica Hans Wegner Wishbone Chair

What's new

Sanity Create Content Mapping, Visual Editing, and Content Releases

localhost:3000/studio/structure/product