

DAY 3 - API INTEGRATION AND DATA MIGRATION

The goal for Day 3 is to integrate APIs and migrate data into Sanity CMS to create a fully functional backend for your marketplace. This process mirrors real-world practices, equipping you with the skills to integrate headless APIs and migrate data from platforms like Shopify, E-Commerce, or custom backends.

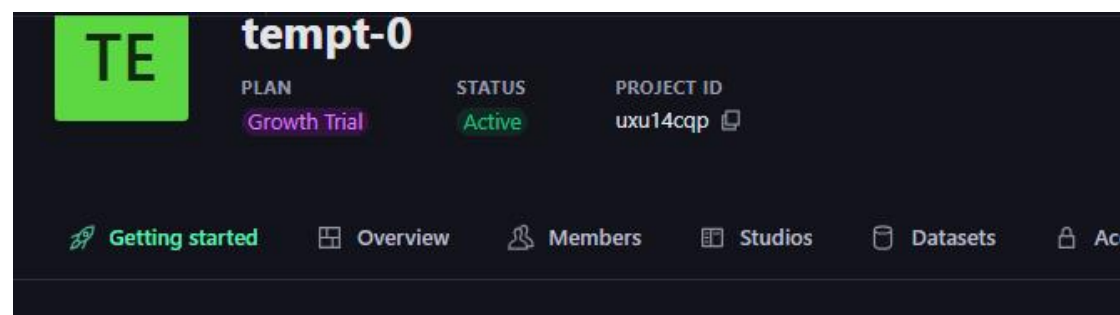
```
1 high severity vulnerability

To address all issues, run:
  npm audit fix --force

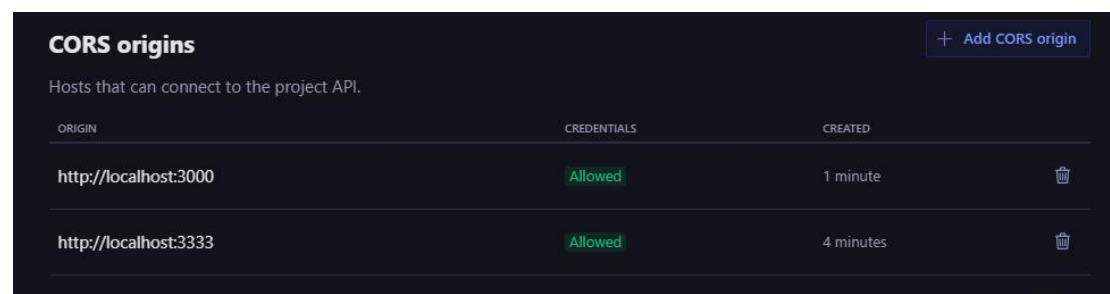
Run `npm audit` for details.

Success! Your Sanity configuration files has been added to this project
```

Once you have set up your Sanity.io project, you can find your **Project ID** in the Sanity dashboard.



To enable CORS for your Sanity project, you can add an allowed origin using the Sanity CLI:



To save an **API token** in the Sanity dashboard:

Examples: "Employee import", "Website preview" or "PDF generator".

Permissions
Choose the access privileges for the token.

☐ **Contributor**
Read and write access to draft content within all datasets, with no access to project settings. (Tokens: read+write drafts)

☐ **Deploy Studio (Token only)**
Access to deploy Sanity Studio and GraphQL APIs to our hosted service.

☐ **Developer**
Read and write access to all datasets, with access to project settings for developers. (Tokens: read+write)

☒ **Editor**
Read and write access to all datasets, with limited access to project settings. (Tokens: read+write)

☐ **Viewer**
Read access to all datasets, with limited access to project settings. (Tokens: read-only)

To generate and use an API token in Sanity, you can create a token from your Sanity dashboard:

NAME	PERMISSIONS	CREATED
tempt0	Editor	just now

Copy the token below – this is your only chance to do so!

```
skFAju2M0QND0MOYXJzbXUGvGY1cPBqgotN7uZEAylZdcz2rXkzqwpXGNXR58bf8vemHgqbQ1SwUr0KMmgApPeAqqAhXbvuDosaJyfyfjiRJKWDTeUIx2j4nn3IXiMSTSBdTVaKEBQg7DaDx3WSIhz0sNmhcntIMoziZrnJP59TjmmZJyxNdFM
```

To store your Sanity API token securely in an `.env` file, you can add it like this

```
$ .env.local x
$ .env.local
1  NEXT_PUBLIC_SANITY_PROJECT_ID="uxu14cqp"
2  NEXT_PUBLIC_SANITY_DATASET="production"
3  SANITY_API_TOKEN="skFAju2M0QND0MOYXJzbXUGvGY1cPBqgotN7uZEAylZdcz2rXkzqwpXGNXR58bf8vemHgqbQ1SwUr0KMmgApPeAqqAhXbvuDosaJyfyfjiRJKWDTeUIx2j4nn3IXiMSTSBdTVaKEBQg7DaDx3WSIhz0sNmhcntIMoziZrnJP59TjmmZJyxNdFM"
4
5
```

In your Sanity project, the folder structure typically looks something like this.

```
JS product.js 11
  ▼ sanity 12
    > lib 13
      ▼ schemaTypes 14
        TS index.ts U 15
        TS product.ts U 16
      TS env.ts U 17
      TS structure.ts U 18
    $ .env.local 19
```

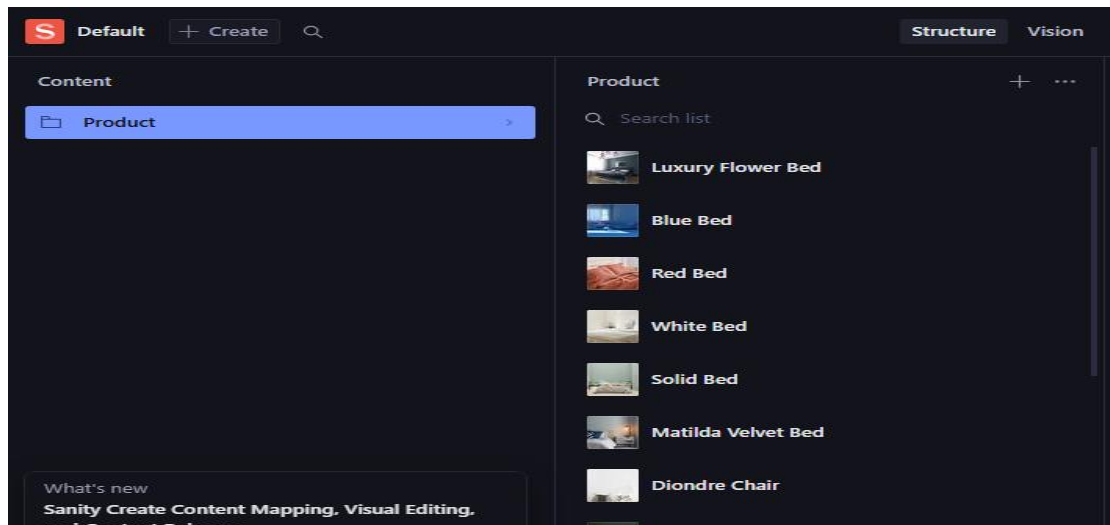
For the product.ts schema, it would be defined like this.

```
src > sanity > schemaTypes > TS product.ts > default > fields > options
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Name',
    },
    {
      name: 'image',
      type: 'image',
      title: 'Image',
      options: {
        hotspot: true,
      },
    },
    {
      name: 'price',
      type: 'number',
      title: 'Price',
    },
    {
      name: 'description',
      type: 'text',
      title: 'Description',
    },
  ],
}
```

In the index.ts file, you would import and export the schema:
This structure is typical for organizing your schema files and allows easy management of different content types.

```
src > sanity > schemaTypes > TS index.ts > schema > types
1 import { type SchemaTypeDefinition } from 'sanity'
2 import product from './product'
3
4 export const schema: { types: SchemaTypeDefinition[] } = {
5   types: [product],
6 }
7
```

After running Sanity locally with `sanity start`, your Sanity studio should be accessible at `http://localhost:3333`. You can use this URL to interact with your Sanity CMS locally.



To fetch data from Sanity using Vision, you can use the following code:

This will fetch and log all the products from Sanity using a basic GROQ query.

