

Marketplace Builder Hackathon 2025 Day 3:

DAY 3 - API

INTEGRATION AND

DATA MIGRATION

**PROCESS AFTER INSTALLATION SANITY IN
PROJECT**

CREATE SANITY PROJECT WITH NAME DAY3/

S

amna khan

day 3\

30 days left in trial

a

D3

amna khan

day 3\

PLAN

Growth Trial

STATUS

Active

PROJECT ID

4n58q6yu

Getting started

Overview

Members

Studios

Datasets

Access

Activity

Usage

Plan

API

Settings

Webhooks

CORS origins

Tokens

GROQ-powered webhooks

HTTP callbacks to a given URL triggered by changes in your content lake

Learn more about webhooks

+ Create webhook

0 of 2 webhooks (2 included in plan)

Get more webhooks

What's new

Sanity Create Content Mapping, Visual Editing, and Content Releases

GENERATE TOKEN:

sanity.io/manage/personal/project/4n58q6yu/api

Google Chrome isn't your default browser [Set as default](#)

amna khan day 3\ 30 days left in trial

Getting started Overview Members Studios Datasets Access Activity Usage Plan API Settings

Webhooks

CORS origins

Tokens

NAME	PERMISSIONS	CREATED
read token	Developer	just now

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

```
skumX5F9hoxF58YXx9hNi1J7Z8de4NNPjtg5J7X3zKqgR9tLNWr7VFFUM31rQvygBqI33B6tAxNYKPCcdwmDgM  
yX6M8QPTCuQmt0tTGzE4BmebXGdEnytNtHJG1aHS2TxFWjQWgKjYsdY8soQbkxUo6eP2TJ9TzOKMQ617sPXfHpgi  
61tdjVF1
```

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

Token created

PASTE THAT TAKEN AND ID IN ENV.LOCAL FILE

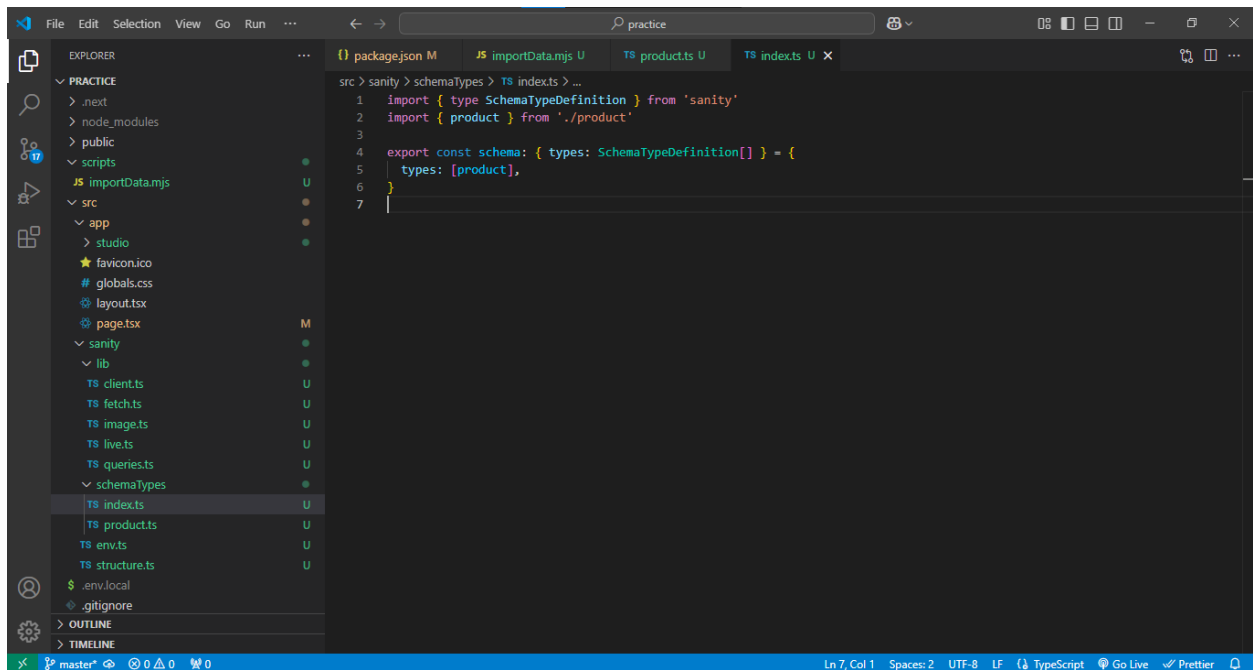
The screenshot shows a code editor with the following files in the Explorer:

- .env.local
- .gitignore
- eslint.config.mjs
- next-env.d.ts
- next.config.ts
- package-lock.json
- package.json
- postcss.config.mjs
- README.md
- sanity.d.ts
- sanity.config.ts
- tailwind.config.ts
- tsconfig.json

The .env.local file contains the following content:

```
1 NEXT_PUBLIC_SANITY_PROJECT_ID="sa5nz3ss"  
2 NEXT_PUBLIC_SANITY_DATASET="production"  
3 SANITY_API_TOKEN="sk30T6b0IbZoygJfGVtaFq6lq8CHYURkUpr8BFhimN11SE2W1WMyk1i1mLqmS1XUHxuXtF2dr71cyHeBUzi9k5DRkoKAMFGuZOAK"  
4
```

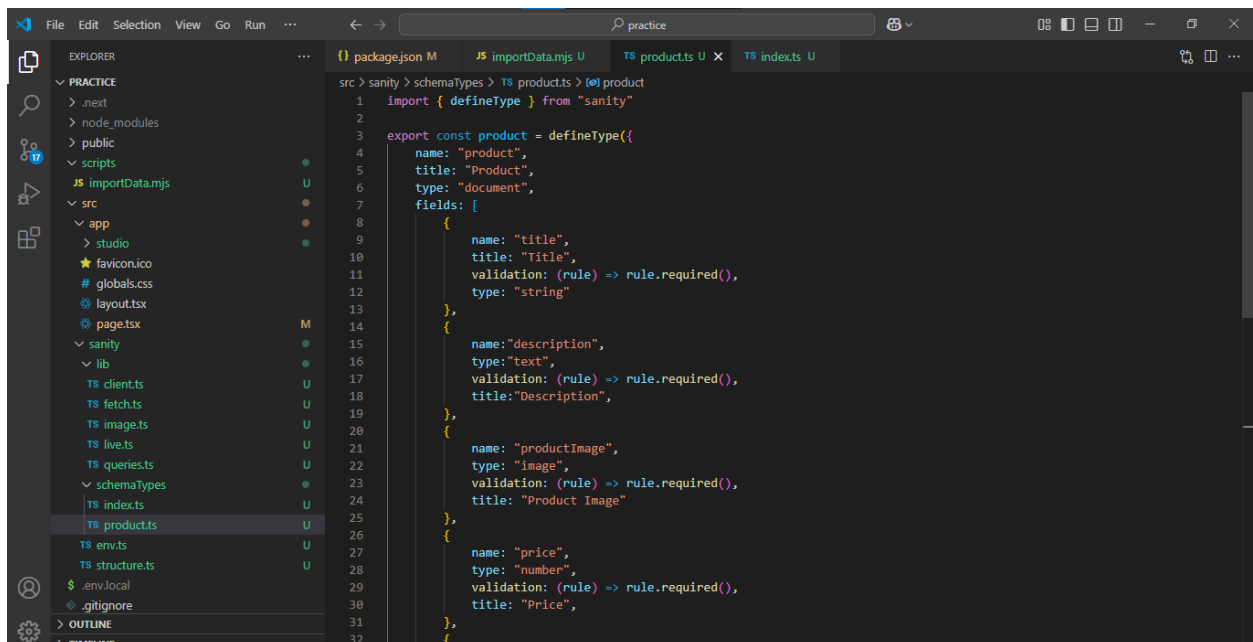
IMPORT PRODUCT IN INDEX.TS:



The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer shows a project structure with folders like .next, node_modules, public, scripts, src, and sanity. The src folder contains subfolders like app, studio, and sanity. The sanity folder contains lib, schemaTypes, and other files. The schemaTypes folder contains TS client.ts, TS fetch.ts, TS image.ts, TS live.ts, TS queries.ts, and TS schemaTypes. The TS schemaTypes folder contains TS indexes.ts, TS product.ts, TS env.ts, and TS structure.ts. The TS product.ts file is selected in the Explorer. The main editor shows the content of TS product.ts, which is a TypeScript file defining a schema for a product. The code is as follows:

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

PASTE SCHEMA IN PRODUCT.TS



The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer shows a project structure with folders like .next, node_modules, public, scripts, src, and sanity. The src folder contains subfolders like app, studio, and sanity. The sanity folder contains lib, schemaTypes, and other files. The schemaTypes folder contains TS client.ts, TS fetch.ts, TS image.ts, TS live.ts, TS queries.ts, and TS schemaTypes. The TS schemaTypes folder contains TS indexes.ts, TS product.ts, TS env.ts, and TS structure.ts. The TS product.ts file is selected in the Explorer. The main editor shows the content of TS product.ts, which is a TypeScript file defining a schema for a product. The code is as follows:

```
src > sanity > schemaTypes > TS products > TS product
1  import { defineType } from "sanity"
2
3  export const product = defineType({
4    name: "product",
5    title: "Product",
6    type: "document",
7    fields: [
8      {
9        name: "title",
10       title: "Title",
11       validation: (rule) => rule.required(),
12       type: "string"
13     },
14     {
15       name: "description",
16       type: "text",
17       validation: (rule) => rule.required(),
18       title: "Description",
19     },
20     {
21       name: "productImage",
22       type: "image",
23       validation: (rule) => rule.required(),
24       title: "Product Image"
25     },
26     {
27       name: "price",
28       type: "number",
29       validation: (rule) => rule.required(),
30       title: "Price",
31     },
32   ],
33 })
```

CHANGED PROJECT ID AND TOKEN IN IMPORT-DATA.JS :

The screenshot shows a VS Code editor with the file explorer on the left and the editor window on the right. The file explorer shows a project structure with a 'scripts' directory containing 'importData.mjs'. The editor window shows the code for 'importData.mjs', which includes the 'uploadImageToSanity' function. The function uses the 'createClient' function from '@sanity/client' to create a client instance with projectId, dataset, useCdn, apiVersion, and token. It then uses the 'upload' method to upload an image to the Sanity cloud. The function returns the asset ID if successful, or null if an error occurs.

```
1 import { createClient } from '@sanity/client';
2
3 const client = createClient({
4   projectId: 'sa5n23ss',
5   dataset: 'production',
6   useCdn: true,
7   apiVersion: '2025-01-13',
8   token: 'sks30T6b0IbZoyg3fGVTaFqG1q8cHYURKUpR8BFh1mN115E2W1MMYk11imLqmS1XUHXuXtF2dr71cyHebUz19k5DRk0KAMfGwZOAKE3tjI1Txbf7
9 });
10
11 async function uploadImageToSanity(imageUrl) {
12   try {
13     console.log(`Uploading image: ${imageUrl}`);
14
15     const response = await fetch(imageUrl);
16     if (!response.ok) {
17       throw new Error(`Failed to fetch image: ${imageUrl}`);
18     }
19
20     const buffer = await response.arrayBuffer();
21     const bufferImage = Buffer.from(buffer);
22
23     const asset = await client.assets.upload('image', bufferImage, {
24       filename: imageUrl.split('/').pop(),
25     });
26
27     console.log(`Image uploaded successfully: ${asset._id}`);
28     return asset._id;
29   } catch (error) {
30     console.error(`Failed to upload image:`, imageUrl, error);
31     return null;
32   }
```

MADE CHANGES IN PACKAGES.JSON

The screenshot shows a VS Code editor with the file explorer on the left and the editor window on the right. The file explorer shows a project structure with a 'package.json' file. The editor window shows the code for 'package.json', which includes the 'name', 'version', 'private', 'scripts', 'dependencies', and 'devDependencies' fields. The 'scripts' field has been updated to include 'dev', 'build', 'start', 'lint', and 'import-data'. The 'dependencies' field has been updated to include '@sanity/image-url', '@sanity/vision', 'next', 'next-sanity', 'react', 'react-dom', 'sanity', and 'styled-components'. The 'devDependencies' field has been updated to include '@eslint/eslintre', '@types/node', '@types/react', '@types/react-dom', 'eslint', 'eslint-config-next', 'postcss', 'tailwindcss', and 'typescript'.

```
1 {
2   "name": "practice",
3   "version": "0.1.0",
4   "private": true,
5   "scripts": {
6     "dev": "next dev --turbo",
7     "build": "next build",
8     "start": "next start",
9     "lint": "next lint",
10    "import-data": "node scripts/importData.mjs"
11  },
12  "dependencies": {
13    "@sanity/image-url": "^1.1.0",
14    "@sanity/vision": "^3.70.0",
15    "next": "15.1.5",
16    "next-sanity": "^9.8.39",
17    "react": "^19.0.0",
18    "react-dom": "^19.0.0",
19    "sanity": "^3.70.0",
20    "styled-components": "^6.1.14"
21  },
22  "devDependencies": {
23    "@eslint/eslintre": "^3",
24    "@types/node": "^20",
25    "@types/react": "^19",
26    "@types/react-dom": "^19",
27    "eslint": "^9",
28    "eslint-config-next": "15.1.5",
29    "postcss": "^8",
30    "tailwindcss": "^3.4.1",
31    "typescript": "^5"
```

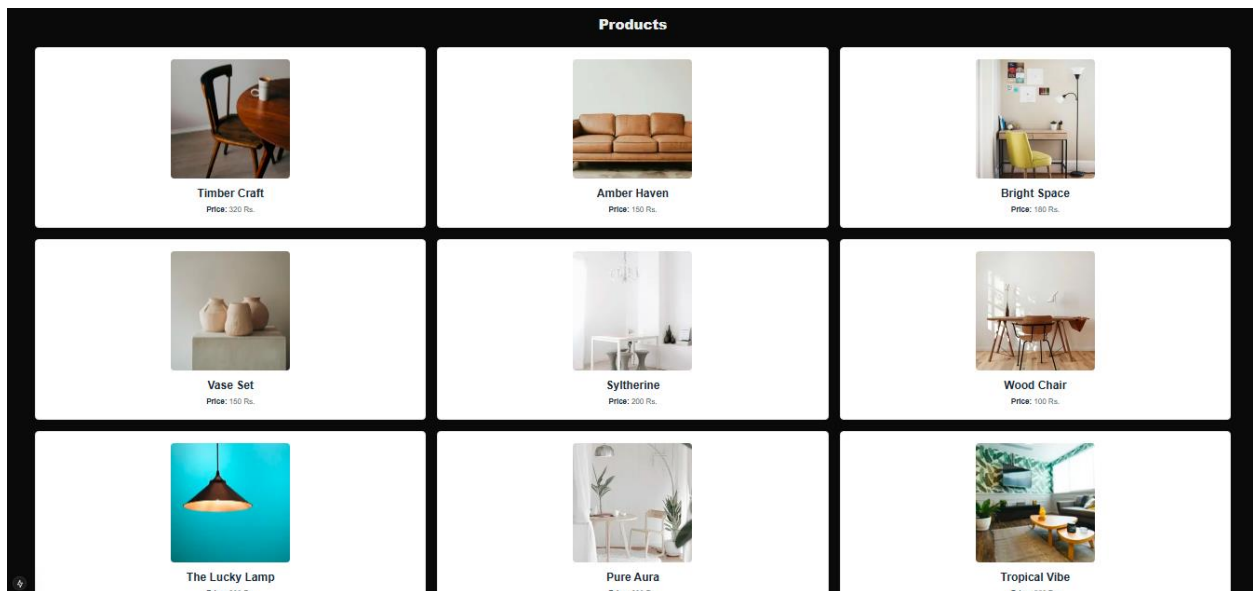
START FETCHING DATA:

```
File Edit Selection View Go Run ... UI-UX-Hackathon
EXPLORER
  UI-UX-HACKATHON
    > .next
    > node_modules
    > public
      file.svg
      globe.svg
      next.svg
      vercel.svg
      window.svg
    > sanity
      > lib
      > schemaTypes
      TS index.ts U
      TS product.ts U
      TS env.ts U
      TS structure.ts U
      > script
        JS importData.js U
        > src
        $ .env.local
        .gitignore
        JS eslint.config.mjs
        TS next-env.d.ts
        TS next.config.ts
        () package-lock.json M
        () package.json M
        JS postcss.config.mjs
    > OUTLINE
    > TIMELINE
script > JS importData.js > client > projectId
1 import { createClient } from '@sanity/client';
2

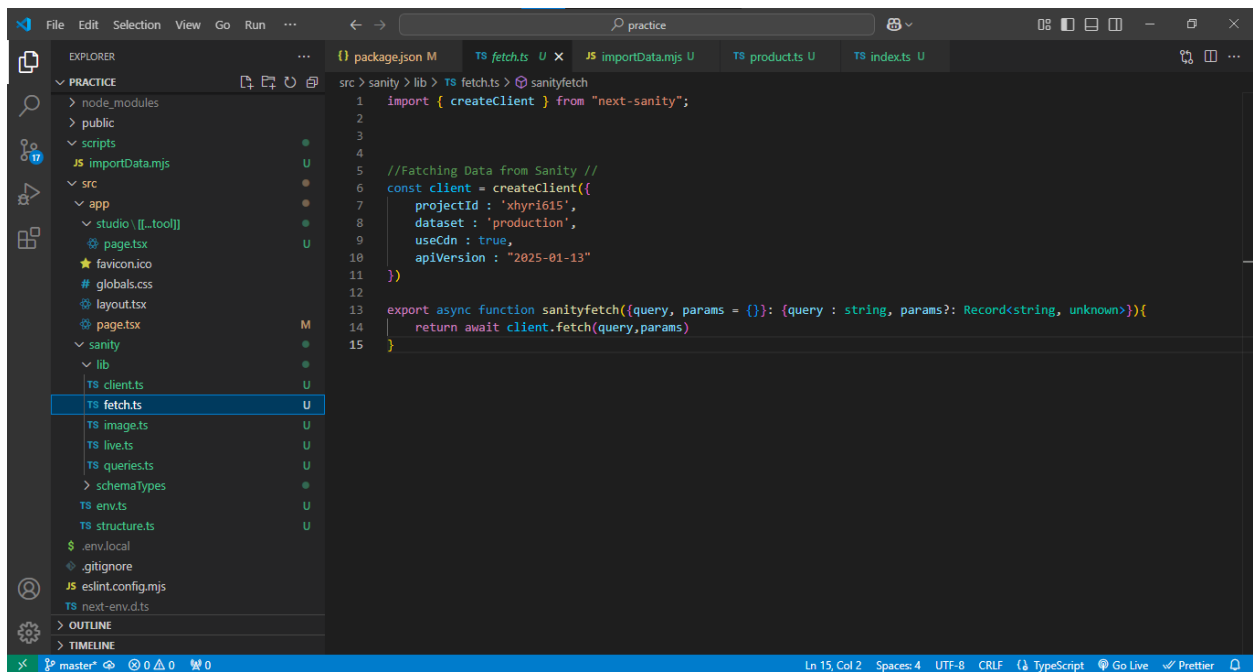
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node + v ...

xperience. The chair's clean lines and refined shape give it a stylish yet relaxed feel, while its bold yet understated presence adds a touch of refinement to any space.\n' +
'\n' +
'Whether you choose a neutral tone or a pop of color, The Dandy Chair effortlessly complements a wide variety of decor styles, from modern minimalism to traditional luxury. Its versatility and comfort make it ideal for those who value both aesthetics and practicality in their furniture choices.\n' +
'\n' +
'Key Features:\n' +
'\n' +
'Comfortable, plush cushioning with ergonomic support\n' +
'Timeless design that adds sophistication to any space\n' +
'Crafted with high-quality materials for durability and longevity\n' +
'Versatile style that complements both modern and traditional interiors\n' +
'Ideal for living rooms, offices, or any area that calls for a touch of elegance\n' +
'Experience the perfect blend of comfort and style with The Dandy Chair—where luxury meets functionality and every detail is crafted to impress.',
discountPercentage: 35,
isNew: true,
price: 150,
productImage: {
  _type: 'image',
  asset: {
    _ref: 'image-e2cc69a6c9f9d1ff1fcf8e839844240e16f5f758-305x375-png'
  }
},
tags: [ 'chair ', 'elegant ', 'vintage ', 'classic ', 'furniture' ],
title: 'The Dandy chair'
}
Uploading image: https://cdn.sanity.io/images/7xt4qcah/production/8795ff7bdb3d810dbe0c3eb52cd22116c9115bce-305x375-png
Image uploaded successfully: image-8795ff7bdb3d810dbe0c3eb52cd22116c9115bce-305x375-png
```

FINALLY GOT THIS ON FRONTEND:



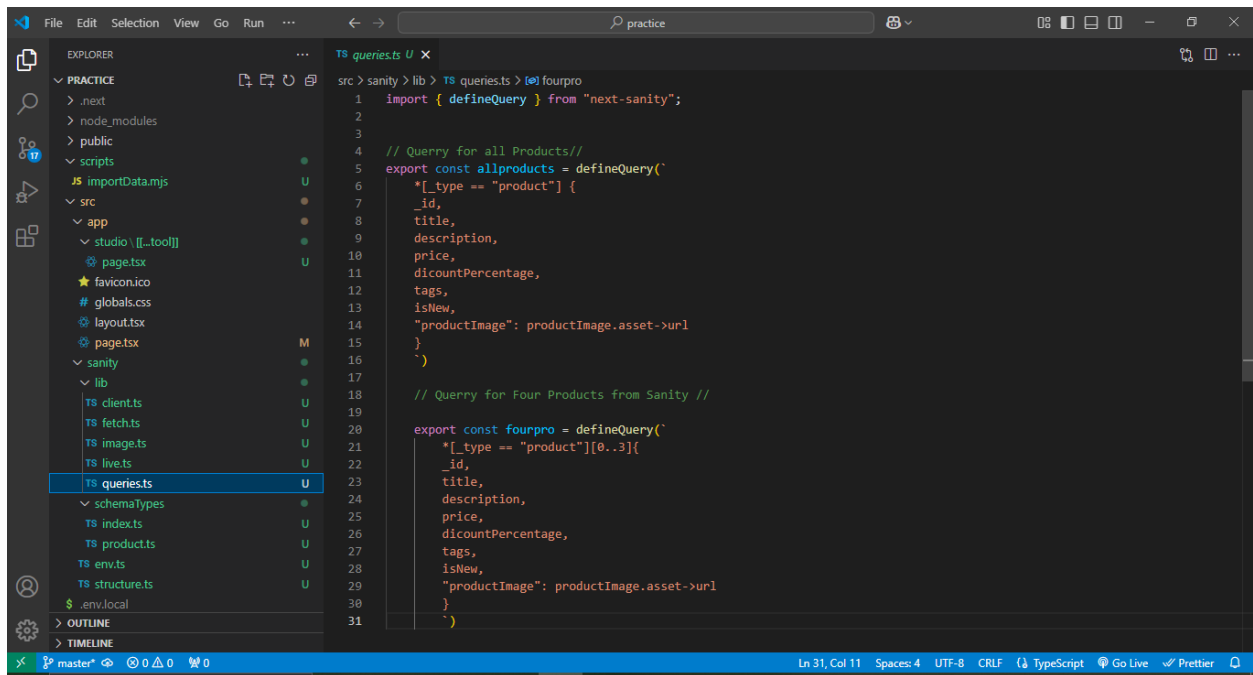
FETCH.TS FILE :



The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project structure with a 'src' directory containing 'sanity' and 'lib' subdirectories. The 'sanity' directory contains 'lib' and 'fetch.ts'. The 'lib' directory contains 'client.ts', 'image.ts', 'live.ts', 'queries.ts', 'schemaTypes', 'env.ts', 'structure.ts', and 'env.local'. The 'fetch.ts' file is selected and its content is displayed in the editor. The code defines a 'sanityfetch' function that uses 'createClient' from 'next-sanity' to fetch data from a Sanity dataset.

```
1  import { createClient } from "next-sanity";
2
3
4
5  //Fetching Data from Sanity //
6  const client = createClient({
7    projectId : 'xhyri615',
8    dataset : 'production',
9    useCdn : true,
10    apiVersion : "2025-01-13"
11  })
12
13  export async function sanityfetch(query, params = {}): {query : string, params?: Record<string, unknown>}{
14    return await client.fetch(query,params)
15  }
```

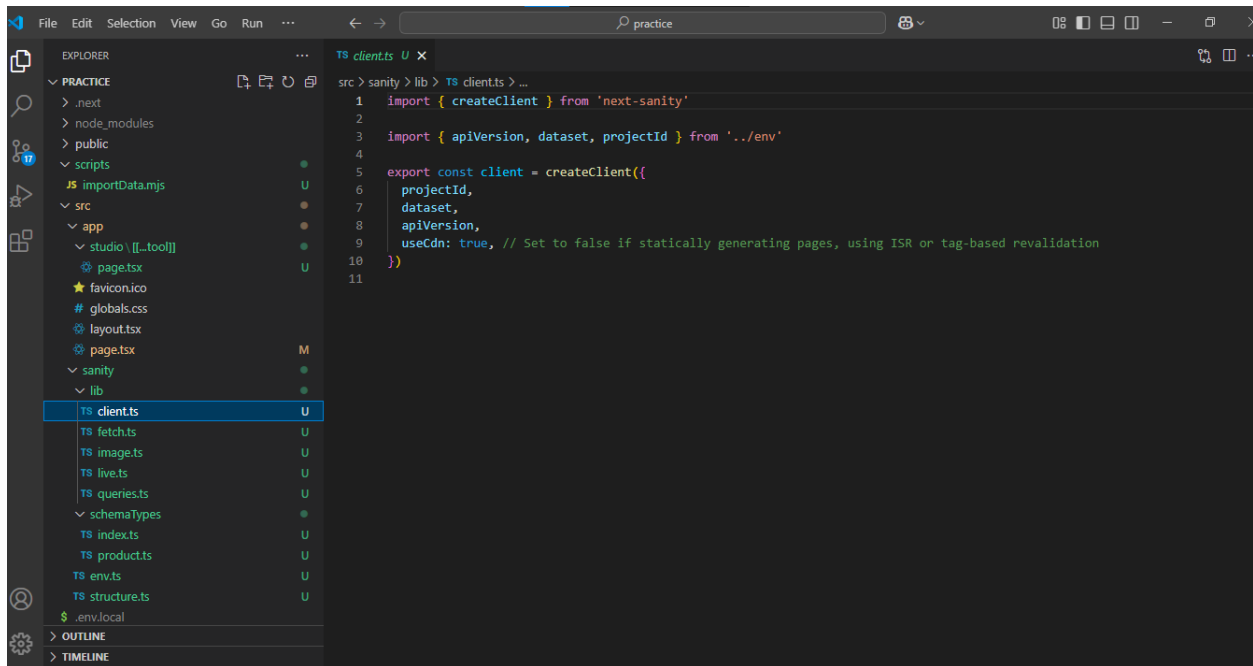
QUERIES.TS FILE :



The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project structure with a 'src' directory containing 'sanity' and 'lib' subdirectories. The 'sanity' directory contains 'lib' and 'queries.ts'. The 'lib' directory contains 'client.ts', 'image.ts', 'live.ts', 'queries.ts', 'schemaTypes', 'env.ts', 'structure.ts', and 'env.local'. The 'queries.ts' file is selected and its content is displayed in the editor. The code defines two queries: 'allproducts' and 'fourpro', both using 'defineQuery' from 'next-sanity' to fetch data from a Sanity dataset.

```
1  import { defineQuery } from "next-sanity";
2
3
4  // Query for all Products//
5  export const allproducts = defineQuery("
6    *[_type == 'product'] {
7      _id,
8      title,
9      description,
10     price,
11     dicountPercentage,
12     tags,
13     isNew,
14     "productImage": productImage.asset->url
15   }
16   ")
17
18  // Query for Four Products from Sanity //
19
20  export const fourpro = defineQuery("
21    *[_type == 'product'][0..3]{
22      _id,
23      title,
24      description,
25      price,
26      dicountPercentage,
27      tags,
28      isNew,
29      "productImage": productImage.asset->url
30    }
31  ")
```

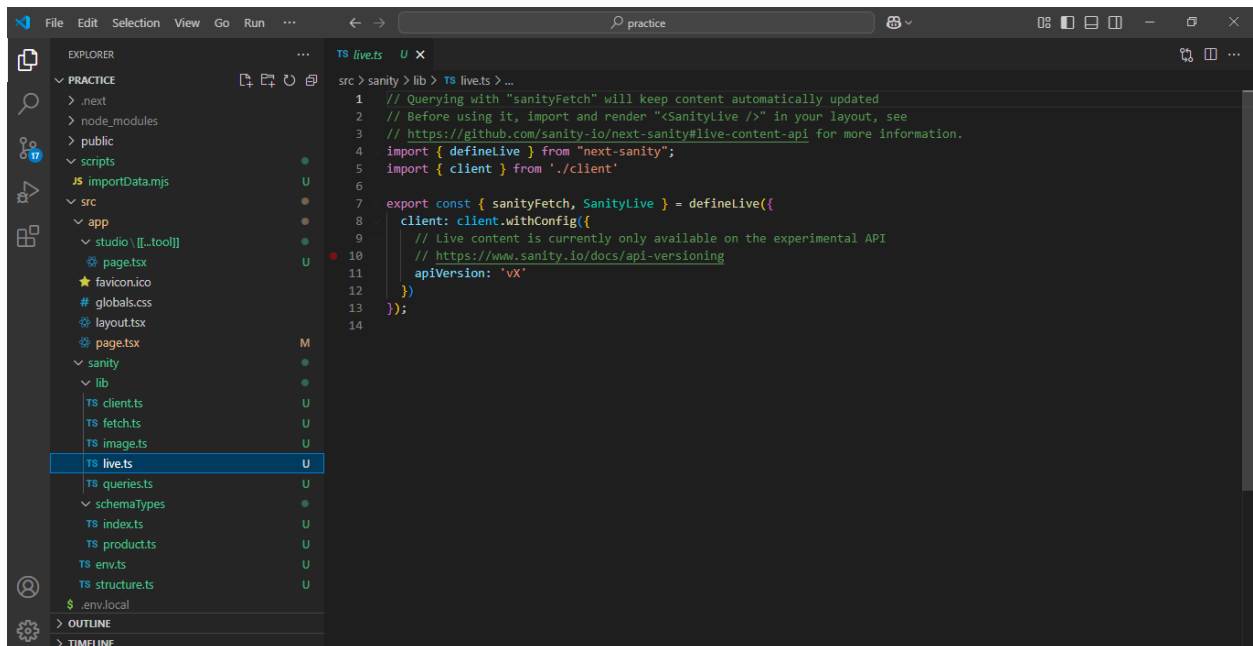
CLIENT.TS FILE :



The screenshot shows the VS Code interface with the Explorer on the left and the editor on the right. The Explorer shows a project structure with folders like .next, node_modules, public, scripts, and src. The src folder contains subfolders app, studio, and sanity. The sanity folder contains a lib folder with client.ts selected. The editor shows the content of client.ts, which imports createClient from 'next-sanity' and defines a client constant.

```
src > sanity > lib > TS client.ts > ...
1  import { createClient } from 'next-sanity'
2
3  import { apiVersion, dataset, projectId } from '../env'
4
5  export const client = createClient({
6    projectId,
7    dataset,
8    apiVersion,
9    useCdn: true, // Set to false if statically generating pages, using ISR or tag-based revalidation
10  })
11
```

LIVE.TS FILE:



The screenshot shows the VS Code interface with the Explorer on the left and the editor on the right. The Explorer shows the same project structure as the previous screenshot, but with the live.ts file selected in the sanity/lib folder. The editor shows the content of live.ts, which imports defineLive and client from 'next-sanity' and defines a live content function.

```
src > sanity > lib > TS live.ts > ...
1  // Querying with "sanityFetch" will keep content automatically updated
2  // Before using it, import and render "<SanityLive />" in your layout, see
3  // https://github.com/sanity-io/next-sanity#live-content-api for more information.
4  import { defineLive } from "next-sanity";
5  import { client } from './client'
6
7  export const { sanityFetch, SanityLive } = defineLive({
8    client: client.withConfig({
9      // Live content is currently only available on the experimental API
10     // https://www.sanity.io/docs/api-versioning
11     apiVersion: "vx"
12   })
13 });
14
```