

Day 3 Activities

API INTEGRATION AND MIGRATION

STEP 1

API INTEFGRATION:

- Create nextjs project [`npx create-next-app@14.2.2`]
- Create Sanity project from sanity io
- Overview copy sanity command
- Paste overview sanity command on nextjs terminal then run command

The screenshot shows the Sanity.io dashboard for a project named 'my_project'. The browser address bar displays 'sanity.io/manage/personal/project/jfwkxpt'. The user 'Asif Khan' is logged in. The project details show a 'Growth Trial' plan, 'Active' status, and project ID 'jfwkxpt'. A navigation bar includes links for 'Getting started', 'Overview' (selected), 'Members', 'Studios', 'Datasets', and a lock icon. Under the 'Next steps' section, there is a box titled 'Initialize your project with the CLI' which instructs the user to run a command in their terminal. The command is: `npm create sanity@latest -- --project jfwkxpt --dataset produ`. A 'Copy' button is next to the command. Below the command box is a link: 'Having issues with the CLI?'. The 'Usage' section at the bottom shows a progress bar with '0 / 1m' and '145 / 250k'.

sanity.io/manage/personal/project/jfwkxpt

Asif Khan my_project

MY my_project

PLAN Growth Trial STATUS Active PROJECT ID jfwkxpt

Getting started Overview Members Studios Datasets

Next steps

Initialize your project with the CLI

Run this command in your Terminal to continue setting up your project.

```
npm create sanity@latest -- --project jfwkxpt --dataset produ
```

Copy

[Having issues with the CLI?](#)

Usage View more →

0 / 1m 145 / 250k

STEP 2

- Copy schema from schema link according template
- Sanity → Schematype → create file product.ts → paste schema → Import file [product.ts]

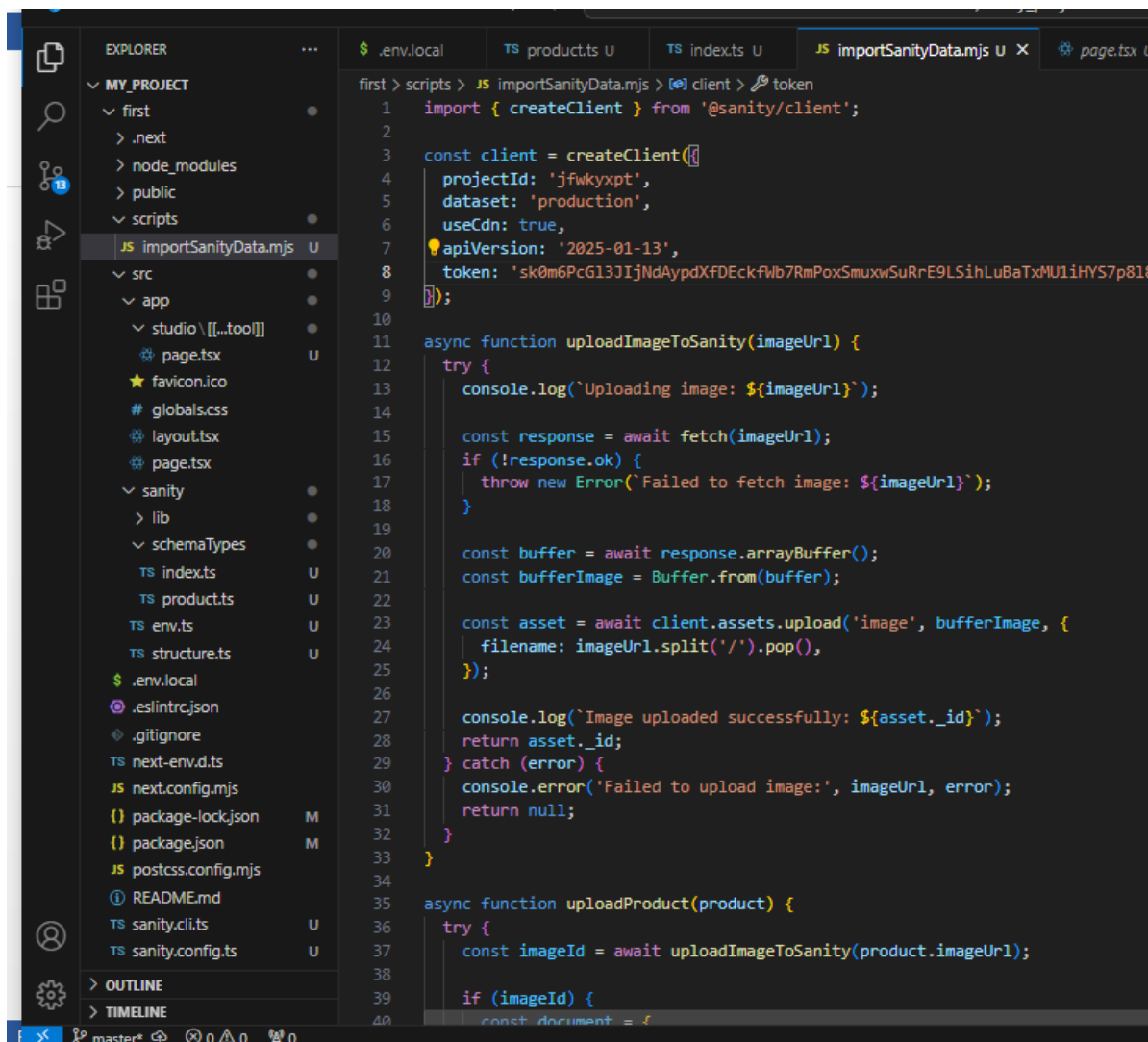
The screenshot shows a VS Code editor with the following components:

- EXPLORER (Left Panel):** Displays the project structure. The `src` directory is expanded, showing `sanity` > `schemaTypes`. The file `product.ts` is selected and highlighted in blue.
- Editor (Main Panel):** Shows the content of `product.ts`. The file path in the breadcrumb is `first > src > sanity > schemaTypes > product.ts`. The code is as follows:

```
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      type: "string"
12    },
13    {
14      name: "description",
15      type: "text",
16      title: "Description",
17    },
18    {
19      name: "productImage",
20      type: "image",
21      title: "Product Image"
22    },
23    {
24      name: "price",
25      type: "number",
26      title: "Price",
```
- Bottom Panel:** Shows the status bar with "master*" and other icons.

STEP 3

- Create folder [scripts] then create file[importSanityData.mjs]
- Copy code from migration link and paste file[importSanityData.mjs]



The screenshot shows a VS Code editor interface. On the left, the Explorer sidebar displays a project structure with folders like 'first', 'node_modules', 'public', and 'scripts'. The 'scripts' folder is expanded, showing the file 'importSanityData.mjs'. The main editor area displays the code for 'importSanityData.mjs'. The code imports the 'createClient' function from '@sanity/client', configures the client with project ID, dataset, useCdn, and API version, and then defines two async functions: 'uploadImageToSanity' and 'uploadProduct'.

```
first > scripts > JS importSanityData.mjs > client > token
1  import { createClient } from '@sanity/client';
2
3  const client = createClient({
4    projectId: 'jfwkxypt',
5    dataset: 'production',
6    useCdn: true,
7    apiVersion: '2025-01-13',
8    token: 'sk0m6PcG13JIjNdAypdXfDEckFwb7RmPoxSmuxwSuRrE9LSihLuBaTxMU1iHYS7p818
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    }
33  }
34
35  async function uploadProduct(product) {
36    try {
37      const imageId = await uploadImageToSanity(product.imageUrl);
38
39      if (imageId) {
40        const document = {
```

STEP 4

- Add one one in [package.json]
"import-data": "node scripts/importSanityData.mjs"
- Then command run [npm run import-data]
- Response this

```
esign meets everyday convenience.',
  dicountPercentage: 0,
  isNew: true,
  price: 300,
  productImage: {
    _type: 'image',
    asset: {
      _ref: 'image-b7565060672485ede
    }
  },
  tags: [ 'sleek ', 'modern ', 'eleg
  title: 'Sleek Living'
}
Uploading image: https://cdn.sanity.
7aad1cc88e24116b-3777x5990.jpg
Image uploaded successfully: image-1
Product Serene Seat uploaded success
  _createdAt: '2025-01-18T11:36:15Z'
  _id: 'aWt125CyehY2YfnCtyGEc8',
```

master* 0 0 0

STEP 5

Successfully Fetch

