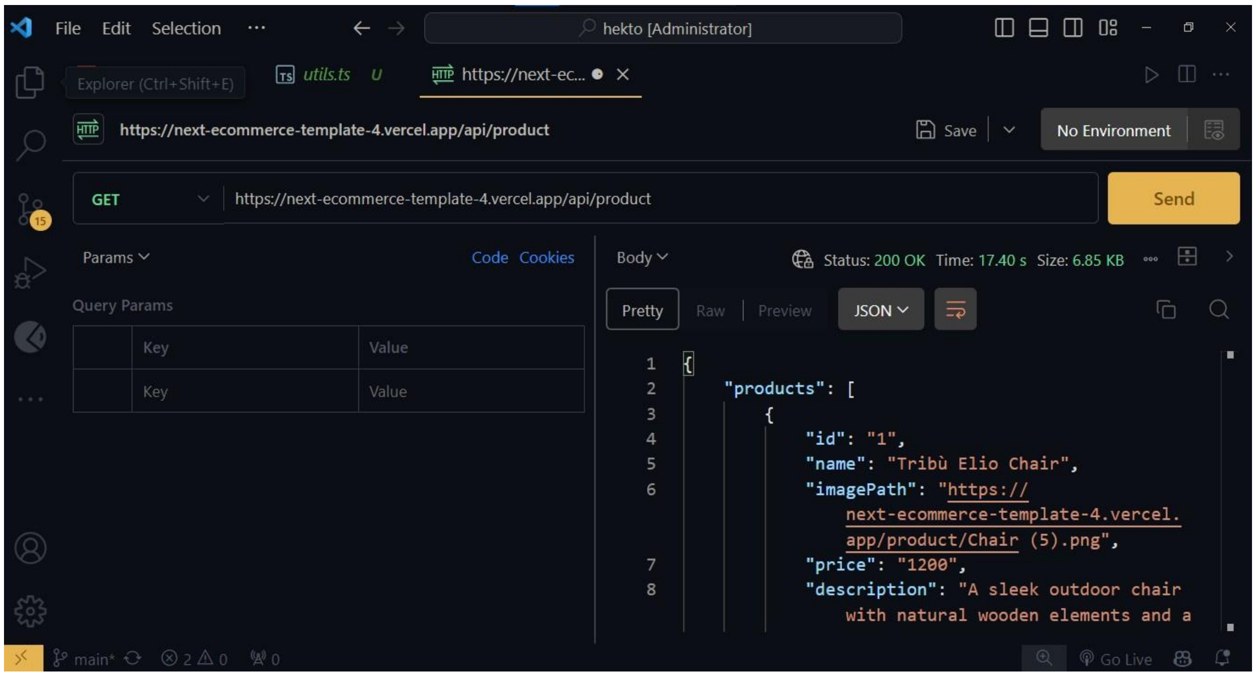
# DAY 3 – API Integration And Data Migration

## API Integration Process:

- Data was initially fetched from the endpoint: [https://next-ecommercetemplate-4.vercel.app/api/product](https://next-ecommercetemplate-4.vercel.app/api/product).
- The fetched data was then imported into Sanity.
- Sanity API was used to retrieve data on both the server and client sides.
- The API output was tested to ensure it met the required specifications.



https://next-ecommerce-template-4.vercel.app/api/product

GET https://next-ecommerce-template-4.vercel.app/api/product

Query Params

| | Key | Value |
|---|---|---|
| | Key | Value |

Status: 200 OK Time: 17.40 s Size: 6.85 KB

```
{
    "products": [
        {
            "id": "1",
            "name": "Tribù Elio Chair",
            "imagePath": "https://
                next-ecommerce-template-4.vercel.
                app/product/Chair (5).png",
            "price": "1200",
            "description": "A sleek outdoor chair
                with natural wooden elements and a
```

## Schema Adjustments:

- The provided schema was mostly aligned with my generated schema, with slight differences in naming.
- Removed Fields: **short_description, specification, sku.**
- New Field Added: **isFeatureProduct**.

## Field Name Changes:

Old Name ⟶ New Name
Title ⟶ Name

Product_image → Image
Quantity → Stock_level

## Migration Steps & Tools Used:

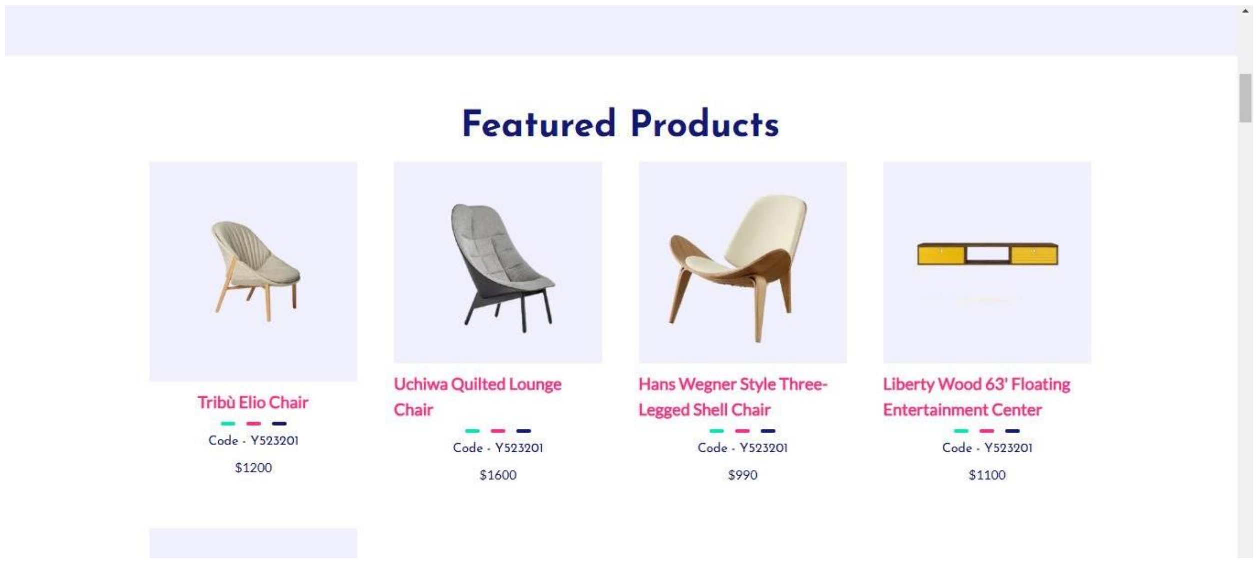
### Migration Process:

- Used two functions:
    1. **Uploading Image**
    2. **Posting Product in Sanity** using the create function, which fetched data from the given API endpoint.
- Called these functions in a **POST request**, secured with a **Dev_Key**.
    - Data could only be inserted if the **Dev_Key** in the request body matched the actual key.
- Sent a simple request to the API, which successfully imported all data into the **Sanity DB**.

### Testing & API Call:

- **Tool Used:** Postman
- **Endpoint:** https://next-ecommerce-template-4.vercel.app/api/product

### Data displayed on frontend


Featured Products

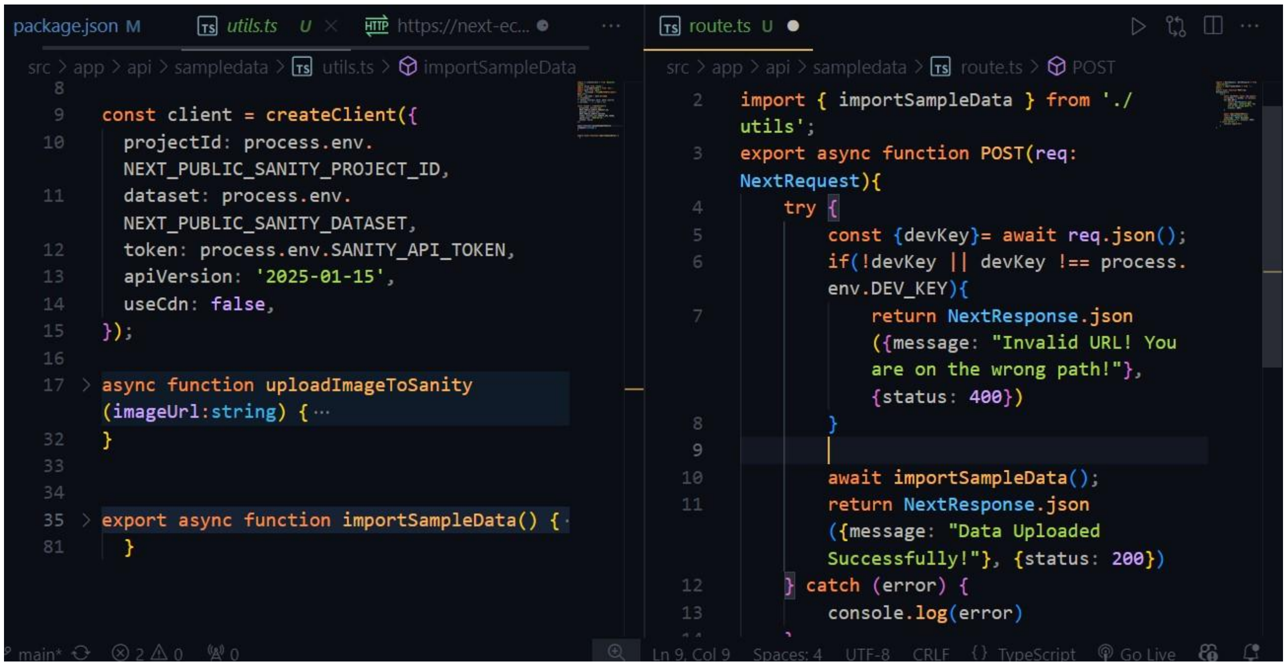
Tribù Elio Chair
Code - Y523201
$1200

Uchiwa Quilted Lounge Chair
Code - Y523201
$1600

Hans Wegner Style Three-Legged Shell Chair
Code - Y523201
$990

Liberty Wood 63' Floating Entertainment Center
Code - Y523201
$1100

## Image Displaying Functions and API Code Snippet

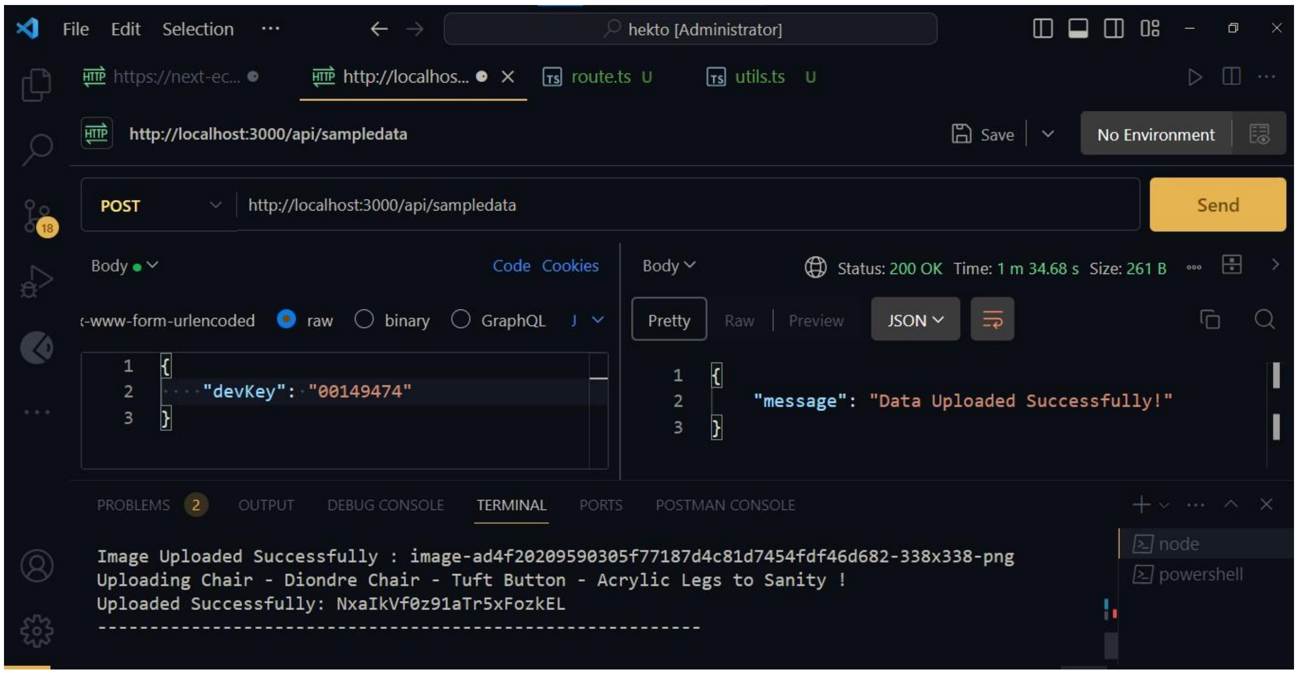


```
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2025-01-15',
  useCdn: false,
});

async function uploadImageToSanity(imageUrl:string) {…
}


export async function importSampleData() {…
  }
```

```
import { importSampleData } from './utils';
export async function POST(req: NextRequest){
    try {
        const {devKey}= await req.json();
        if(!devKey || devKey !== process.env.DEV_KEY){
            return NextResponse.json({message: "Invalid URL! You are on the wrong path!"}, {status: 400})
        }

        await importSampleData();
        return NextResponse.json({message: "Data Uploaded Successfully!"}, {status: 200})
    } catch (error) {
        console.log(error)
```

package.json M
utils.ts U
route.ts U
src > app > api > sampledata > utils.ts > importSampleData
src > app > api > sampledata > route.ts > POST
main*
Ln 9, Col 9  Spaces: 4  UTF-8  CRLF  {} TypeScript  Go Live

## Image Displaying that all the data is imported



POST http://localhost:3000/api/sampledata

Body: raw

```
{
    "devKey": "00149474"
}
```

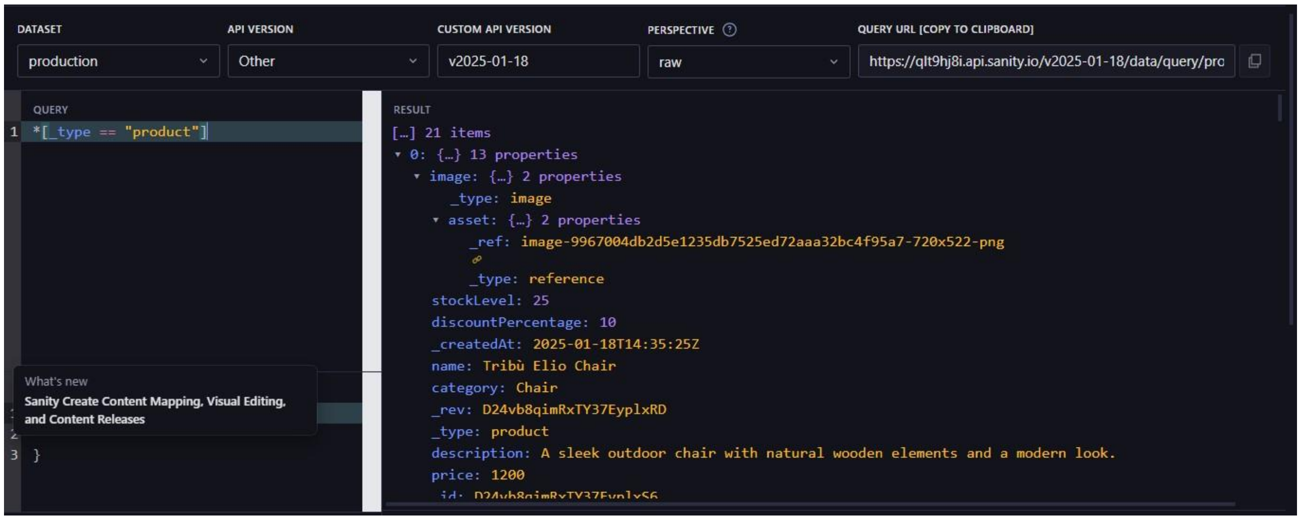
Status: 200 OK  Time: 1 m 34.68 s  Size: 261 B

```
{
    "message": "Data Uploaded Successfully!"
}
```

```
Image Uploaded Successfully : image-ad4f20209590305f77187d4c81d7454fdf46d682-338x338-png
Uploading Chair - Diondre Chair - Tuft Button - Acrylic Legs to Sanity !
Uploaded Successfully: NxaIkVf0z91aTr5xFozkEL
-----------------------------------------------------------
```

File  Edit  Selection
hekto [Administrator]
Save
No Environment
Send
Code  Cookies
x-www-form-urlencoded  raw  binary  GraphQL
Pretty  Raw  Preview  JSON
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POSTMAN CONSOLE
node
powershell

## Sanity Dashboard Showing Data


DATASET: production
API VERSION: Other
CUSTOM API VERSION: v2025-01-18
PERSPECTIVE: raw
QUERY URL [COPY TO CLIPBOARD]: https://qlt9hj8i.api.sanity.io/v2025-01-18/data/query/pro

QUERY

```
*[_type == "product"]
```

RESULT

```
[…] 21 items
▾ 0: {…} 13 properties
  ▾ image: {…} 2 properties
      _type: image
    ▾ asset: {…} 2 properties
        _ref: image-9967004db2d5e1235db7525ed72aaa32bc4f95a7-720x522-png
        _type: reference
    stockLevel: 25
    discountPercentage: 10
    _createdAt: 2025-01-18T14:35:25Z
    name: Tribù Elio Chair
    category: Chair
    _rev: D24vb8qimRxTY37EyplxRD
    _type: product
    description: A sleek outdoor chair with natural wooden elements and a modern look.
    price: 1200
    _id: D24vb8qimRxTY37EyplxS6
```

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

## Conclusion

Now we conclude that the API integration and data migration were successfully implemented, ensuring that data was accurately fetched, processed, and stored in Sanity. The schema adjustments aligned the data structure with project requirements, and the secured migration process ensured data integrity. Testing via Postman confirmed the accuracy of the imported data, and the successful display of data on the frontend and Sanity dashboard validated the entire process.