

# Migration Script:

```
import { createClient } from '@sanity/client'
import axios from 'axios'
import dotenv from 'dotenv'
import { fileURLToPath } from 'url'
import path from 'path'

// Load environment variables from .env.local
const __filename = fileURLToPath(import.meta.url)
const __dirname = path.dirname(__filename)
dotenv.config({ path: path.resolve(__dirname, '../.env.local') })
// Create Sanity client
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2021-08-31'
})

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${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()
    })
    console.log(`Image uploaded successfully: ${asset._id}`)
    return asset._id
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error)
    return null
  }
}

async function importData() {
  try {
    console.log('Fetching products from API...')
    const response = await axios.get('https://sanity-nextjs-rouge.vercel.app/api/foods')
    const products = response.data
    console.log(`Fetched ${products.length} products`)
    for (const product of products) {
      console.log(`Processing product: ${product.title}`)
      let imageRef = null
      if (product.image) {
        imageRef = await uploadImageToSanity(product.image)
      }
      const sanityProduct = {
        _type: 'product',
        name: product.title,
        description: product.description,
        price: product.price,
        discountPercentage: 0,
        priceWithoutDiscount: product.price,
        rating: product.rating?.rate || 0,
        ratingCount: product.rating?.count || 0,
        tags: product.category ? [product.category] : [],
        sizes: [],
        image: imageRef ? {
          _type: 'image',
          asset: {
            _type: 'reference',
            _ref: imageRef,
          },
        } : undefined,
      }
      console.log('Uploading product to Sanity:', sanityProduct.name)
      const result = await client.create(sanityProduct)
      console.log(`Product uploaded successfully: ${result._id}`)
    }
    console.log('Data import completed successfully!')
  } catch (error) {
    console.error('Error importing data:', error)
  }
}

importData()
```