



The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project structure for 'FURNITURE' with folders like '.next', 'Documentation', 'node_modules', 'notes', 'public', 'scripts', 'src', and 'sanity'. The 'scripts' folder is expanded, showing 'importSanityData.mjs'. The main editor area displays the code for this file, which is an asynchronous function 'importData()' that fetches product data from an API, logs it, and then iterates over each product to upload its image to Sanity and prepare a product object for storage. The code includes comments and uses 'await' for asynchronous operations. The status bar at the bottom indicates the current line and column (Ln 16, Col 17), the number of spaces (2), the encoding (UTF-8), the line ending (CRLF), and the file type (JavaScript).

```
37 // Function to import data into Sanity
38
39 async function importData() {
40   try {
41     console.log('Fetching products from API...');
42     const response = await axios.get('https://template-0-beta.vercel.app/api/product');
43     const products = response.data;
44     console.log('Fetched ${products.length} products');
45
46     for (const product of products) {
47       console.log('Processing product: ${product.name}');
48
49       // Upload image to Sanity and get the reference ID
50       let imageRef = null;
51       if (product.imagePath) {
52         imageRef = await uploadImageToSanity(product.imagePath);
53       }
54
55       // Prepare the product data for Sanity
56       const sanityProduct = {
57         _type: 'product',
58         name: product.name,
59         description: product.description,
60         price: parseFloat(product.price),
61         discountPercentage: parseFloat(product.discountPercentage) || 0,
62         priceWithoutDiscount: parseFloat(product.price) / (1 - (parseFloat(product.discountPercentage) || 0) / 100),
63         isFeaturedProduct: product.isFeaturedProduct || false,
64         stockLevel: product.stockLevel || 0,
65         category: product.category || '',
66         image: imageRef
67         ? {
68             _type: 'image',
```

