

Data Flow Documentation - Product Inventory Management System

1. Introduction

The Product Inventory Management System is designed to facilitate efficient management of product inventories through a web-based application. It features:

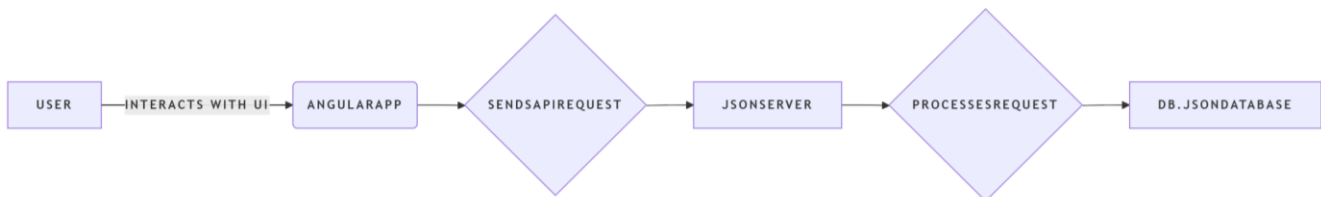
- Angular Frontend for user interface and API interactions.
- Mock Backend (JSON Server) to simulate real API behavior.
- Database (db.json) for storing user and product data.

This document outlines the data flow, system components, services, and API interactions.

2. System Components

- User: Engages with the application through the UI.
- Angular Application: Manages UI rendering, API requests, and data processing.
- JSON Server (Mock API): Handles requests and interacts with the database.
- Database (db.json): Stores product and user details.

3. Data Flow Diagram Overview



4. Detailed Data Flow

4.1 User Authentication Flow

- User submits login/registration details.
- Angular App sends authentication requests to /users via AuthService.
- JSON Server validates credentials and responds.
- User session data is stored, and successful login redirects to the dashboard.

4.2 Product Management Flow

- User performs product actions (view, add, edit, delete).
- Angular App interacts with /products via ProductService.
- JSON Server updates db. json accordingly.
- UI reflects real-time changes based on updated data.

5. Components & Services

5.1 UI Components

- AppComponent: Initializes the application.
- ProductListComponent: Displays available products.
- ProductFormComponent: Handles product addition and modification.
- NavbarComponent: Manages navigation.
- AuthComponent: Handles user authentication.

5.2 Services

- DataService: Manages product CRUD operations.
- AuthService: Handles authentication logic.
- CoreService: Shares data across components.

6. API Endpoints

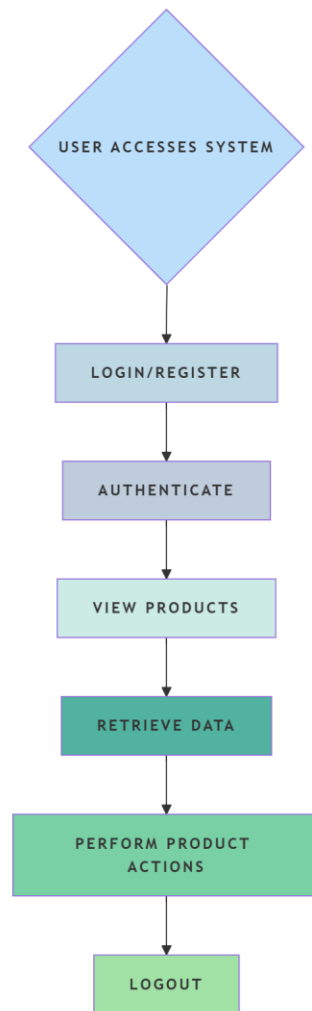
6.1 Authentication APIs

- GET /users - Fetch user list.
- POST /users - Register a new user.

6.2 Product APIs

- GET /products - Retrieve products.
- POST /products - Add a new product.
- PUT /products/:id - Update an existing product.
- DELETE /products/:id - Remove a product.

7. System Flowchart



8. Conclusion

This document provides a structured flow of the Product Inventory Management System, outlining data movement, component interactions, and API communications. The architecture ensures a seamless experience for managing inventory efficiently. Future enhancements can include real-time data updates and additional authentication security measures.