

# Microservices Architecture Projects

Generated on 2025-10-22

## Project 1: Inventory Service

### Data Model - Product Table

Field	Type	Constraints
product_id	UUID / INT	Primary Key
name	STRING	Not Null, Unique
description	STRING	Optional
price	DECIMAL	Not Null
quantity	INTEGER	Not Null
created_at	TIMESTAMP	Default: now()
updated_at	TIMESTAMP	Auto-updated

### API Endpoints (CRUD)

Method	Endpoint	Description
GET	/products	Get all products
GET	/products/{id}	Get product by ID
POST	/products	Create new product
PUT	/products/{id}	Update existing product
DELETE	/products/{id}	Delete a product

## Project 2: Order Service

### Data Model - Order Table

Field	Type	Constraints
order_id	UUID	Primary Key
customer_id	UUID	Not Null, FK
status	STRING	Enum: PENDING, CONFIRMED, SHIPPED, DELIVERED, CANCELLED
total_price	DECIMAL	Computed field
created_at	TIMESTAMP	Default: now()
updated_at	TIMESTAMP	Auto-updated

### ***OrderItem Table***

Field	Type	Constraints
order_item_id	UUID	Primary Key
order_id	UUID	Foreign Key → Order(order_id)
product_id	UUID	Foreign Key → Product(product_id)
quantity	INTEGER	Not Null
price	DECIMAL	Snapshot of product price

Total price in Order should be recalculated on insert/update/delete of OrderItem.

### ***Order Lifecycle***

PENDING → CONFIRMED → SHIPPED → DELIVERED → CANCELLED

### ***API Endpoints (CRUD + Item Management)***

Method	Endpoint	Description
GET	/orders	Get all orders
GET	/orders/{order_id}	Get specific order
POST	/orders	Create a new order
PUT	/orders/{order_id}	Update order (e.g., status, totals)
DELETE	/orders/{order_id}	Delete an order
POST	/orders/{order_id}/items	Add item to order
PUT	/orders/{order_id}/items/{item_id}	Update order item quantity
DELETE	/orders/{order_id}/items/{item_id}	Remove item from order

## **Project 3: User Authentication Service**

### ***Data Model - User Table***

Field	Type	Constraints
user_id	UUID	Primary Key
email	STRING	Not Null, Unique
username	STRING	Not Null, Unique
password	STRING	Hashed (bcrypt, argon2, etc.)
role	STRING	e.g. 'customer', 'admin'
is_active	BOOLEAN	Default: true

created_at	TIMESTAMP	Default: now()
updated_at	TIMESTAMP	Auto-updated

### ***Authentication Flow (JWT-based)***

- Login → Validate credentials and issue JWT.
- JWT Access Token → Sent in Authorization: Bearer <token> header.
- Token Verification → Middleware verifies JWT on protected routes.
- Token Expiry → Set a reasonable TTL (e.g., 15m access, 7d refresh).

### ***API Endpoints***

Method	Endpoint	Description
POST	/auth/register	Register a new user
POST	/auth/login	Login and receive JWT token
GET	/auth/me	Get current logged-in user info
POST	/auth/logout	(Optional) Invalidate token/client
POST	/auth/refresh	(Optional) Refresh access token