**🌐 Level 2: Microservices Project – WebsiteGoApplication**

GitHub Repo: [niharikaluna11/WebsiteGoApplication](https://github.com/niharikaluna11/WebsiteGoApplication)   
Project Type: Go Application – Microservices Architecture

---------------------------------------------------------------------------------------------------

📌 Overview

This Level 2 project demonstrates a microservices-based system using Go, focusing on Orders and Payments.

---------------------------------------------------------------------------------------------------

🧩 Microservices Included

🔹 Order Service

* Responsible for creating and managing customer orders.
* Publishes events to Google Pub/Sub when an order is created.

🔹 Payment Service

* Subscribes to order events.
* Processes payments automatically.
* Sends payment updates back to the Order Service via Pub/Sub.

---------------------------------------------------------------------------------------------------

🔐 Authentication Flow

* Users register and log in via the /users API.
* On login, a JWT token is issued.
* All protected APIs (like /orders) require this token.

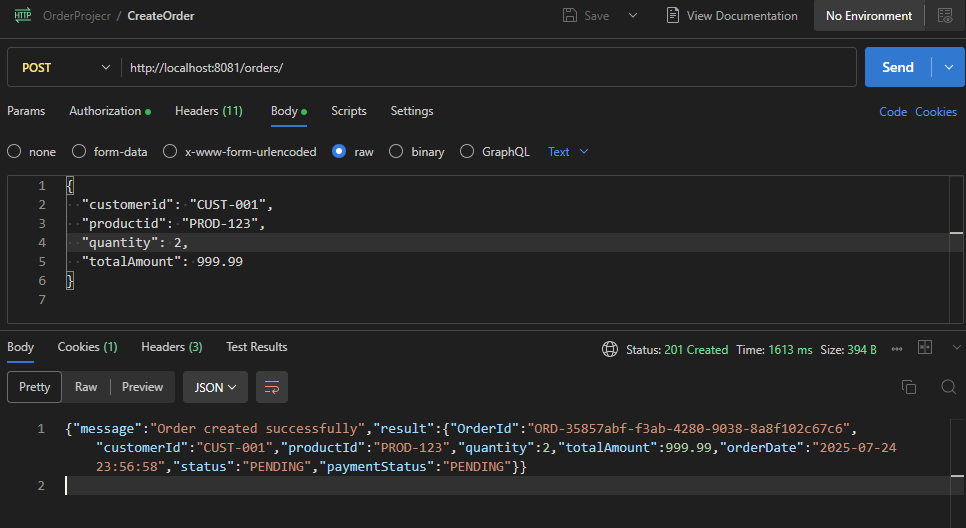
----------------------------------------------------------------------------------------------------------

⚙️ Functional Flow

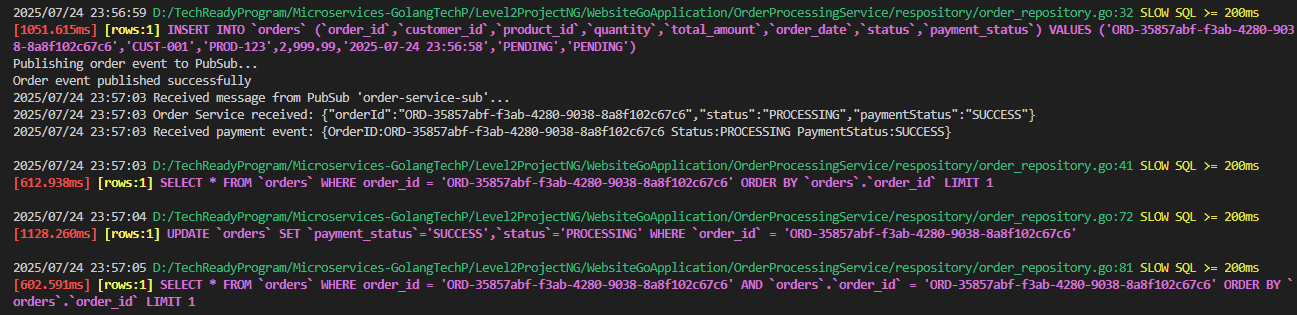
✅ Step 1: Create Order

* Endpoint: POST /orders
* Requires JWT token
* Order is saved with:
* order\_status: PENDING
* payment\_status: PENDING

📷 Visual:   
*A screenshot showing the form/input for creating an order*

   
📋 Logs:   
Log confirms:

* Order inserted into the DB
* Message published to payment-events topic

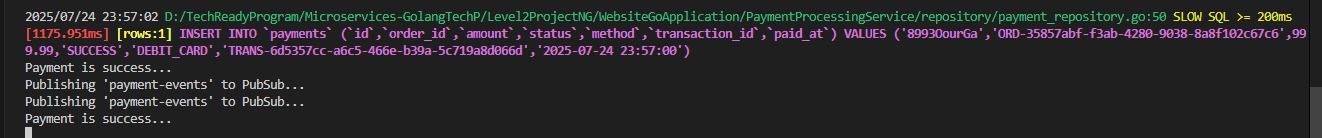


Shape

✅ Step 2: Payment Service (Auto Triggered)

* Payment service listens to the payment-events topic
* Upon receiving a message:
* Processes the payment
* Sends an update to order-updates topic

📷 Visual:   
*Screenshot of Payment Service logs confirming event reception and processing*



Shape

✅ Step 3: View Order by ID

* Endpoint: GET /orders/{id}
* JWT token required
* Returns updated status:
* order\_status: PROCESSING
* payment\_status: SUCCESS

📷 Visual:   
*Screenshot of JSON response or UI showing order details with updated statuses*

A screenshot of a computer

AI-generated content may be incorrect., Picture

----------------------------------------------------------------------------------------------------------

📊 Database Snapshots

🧾 Order Table

📷 Visual:   
*Shows row with sample data — pending initially, then updated to PROCESSING*

A screenshot of a computer

AI-generated content may be incorrect., Picture

💳 Payment Table

📷 Visual:   
*Shows record linked to order, payment marked as SUCCESS*

A screenshot of a computer

AI-generated content may be incorrect., Picture

----------------------------------------------------------------------------------------------------------

🔁 System Interaction Summary

User (JWT Auth)

   ↓

[POST /orders]

   ↓

Order Service → Insert order → Publish to Pub/Sub: payment-events

   ↓

Payment Service ← Subscribed to payment-events

   ↓

Process payment → Publish to Pub/Sub: order-updates

   ↓

Order Service ← Subscribed to order-updates → Update order status

----------------------------------------------------------------------------------------------------------

**🚀 Deployment**

The microservices are deployed on **Google Cloud Platform (GCP)** using **Cloud Run**:

* 🔗 **Order Service**: <https://order-microservice-747901258630.us-central1.run.app>
* 🔗 **Payment Service**: <https://payment-microservice-747901258630.us-central1.run.app>