

## INCIDENT REPORT & ROOT CAUSE ANALYSIS (RCA)

Service: jenish-payment-service

Environment: Local (WSL – Ubuntu)

Date: 02-Dec-2025

Author: Jenish J

### 1. Incident Summary

During testing of the jenish-payment-service, the /payment endpoint displayed intermittent failures, slow responses, and incorrect log entries. The issues were reproducible under repeated API calls. After debugging, the problems were traced to intentionally faulty logic inside the code.

### 2. Captured Logs (Before Fix – Buggy Behavior)

2025-12-02T12:39:01.767Z INFO: Processed payment ID 9999 (actual: 103)

2025-12-02T12:39:21.909Z ERROR: Payment 101 failed randomly

2025-12-02T12:39:23.357Z ERROR: Payment 102 failed randomly

2025-12-02T12:39:30.373Z ERROR: Payment 103 failed randomly

2025-12-02T12:40:07.383Z WARN: Slow processing for payment 103

2025-12-02T12:40:09.386Z INFO: Processed payment ID 9999 (actual: 103)

### 3. Steps to Reproduce

1. Start service: node app.js
2. Repeatedly call endpoint: curl http://localhost:8080/payment
3. Observed:
  - 500 errors (random failure)
  - 2-second delay (slow request)
  - Wrong payment ID logged (9999)
4. Verified evidence in service.log.

#### 4. Root Cause Analysis

##### Root Cause 1 – Random Failure in Code

20% random failure caused intermittent 500 responses.

##### Root Cause 2 – Slow Response Logic

A hardcoded 2-second delay caused performance issues.

##### Root Cause 3 – Incorrect Payment ID Logged

Logging always printed ID 9999 instead of the real payment ID.

#### 5. Fix Implemented

- Removed random failures
- Removed artificial delay
- Corrected log statement to show actual payment ID

#### 6. Captured Logs (After Fix – Correct Behavior)

2025-12-02T12:45:33.984Z jenish-payment-service (FIXED) running on port 8080

2025-12-02T12:46:16.801Z INFO: Successfully processed payment ID 102

2025-12-02T12:46:16.977Z INFO: Successfully processed payment ID 103

2025-12-02T12:46:17.345Z INFO: Successfully processed payment ID 101

#### 7. Impact

Before fix:

- Unstable API
- Misleading logs
- Slow response times

After fix:

- Stable API responses
- Correct logs

- No delays

## 8. Preventive Measures

- Disable fault injection in production
- Add structured logging
- Add response time monitoring
- Add health checks and alerts
- Add unit tests for log validation

## 9. Conclusion

The incident occurred due to intentionally injected faults (random failure, slow response, and incorrect logging). After applying fixes, the service now behaves predictably, and logs are accurate.