Sprint 2 Review

Sprint Goal: Apply feedback and deliver monitoring/logging capabilities

Sprint Duration: Sprint 2
Date: February 19, 2026

Retrospective Improvements Applied

Improvement #1: Add Structured Logging

Status: Implemented

- Created `src/utils/logger.js` with structured JSON logging
- Added logging to all controller operations (CREATE, LIST, GET, UPDATE, DELETE)
- Included warning logs for validation failures and 404 errors
- All logs include timestamps and relevant metadata

Example Log Output:


```
{"timestamp":"2026-02-19T19:05:00.000Z","level":"INFO","message":"API Operation: CREATE
```


Improvement #2: Commit More Frequently

Status: Implemented

Sprint 2 commits show smaller, focused changes:

- `Add structured logger utility`
- `Add logging to task controller operations`
- `Add request logging middleware`
- `Enhance health endpoint with monitoring details`

Improvement #3: Test Edge Cases First

Status: Applied - All existing tests continue to pass

Completed Deliverables

1. Structured Logging System

Files Created:

- `src/utils/logger.js` - Logger utility with JSON format
- `src/middleware/requestLogger.js` - HTTP request logging

Features:

- Log levels: INFO, WARN, ERROR, DEBUG
- JSON structured format for easy parsing
- Timestamps on all log entries
- Operation-specific logging (CREATE_TASK, GET_TASK, etc.)

Controller Logging Demo:

```
# Creating a task produces:
curl -X POST http://localhost:3000/tasks -H "Content-Type: application/json" -d '{"titl

# Log output:
{"timestamp":"2026-02-19T19:05:00.000Z","level":"INFO","message":"API Operation: CREATE
```

2. Request Logging Middleware

All HTTP requests are now logged with:

- HTTP method (GET, POST, PATCH, DELETE)
- Request path
- Response status code
- Response time in milliseconds

- User agent

Example:


```
{"timestamp":"2026-02-19T19:05:00.000Z","level":"INFO","message":"HTTP Request","method
```


3. Enhanced Health Endpoint

Endpoint: `GET /health`

New Response:


```
{
  "status": "healthy",
  "timestamp": "2026-02-19T19:05:00.000Z",
  "uptime": "120 seconds",
  "version": "1.0.0",
  "environment": "development",
  "memory": {
    "heapUsed": "15 MB",
    "heapTotal": "30 MB"
  }
}
```


Monitoring Capabilities Added:

- Server uptime tracking
- Memory usage metrics (heap used/total)
- Version information
- Environment indicator

Sprint 2 Commit History

```
ecf3061 Enhance health endpoint with monitoring details
32470bd Add request logging middleware
22a145d Add logging to task controller operations
156f325 Add structured logger utility
57e2187 Add Sprint 1 Review and Retrospective documents
```

Test Results

All 21 tests continue to pass after Sprint 2 changes:

```
PASS  tests/tasks.test.js
  Task API
    POST /tasks: 6 tests passing
    GET /tasks: 3 tests passing
    GET /tasks/:id: 3 tests passing
    PATCH /tasks/:id: 5 tests passing
    DELETE /tasks/:id: 3 tests passing
    GET /health: 1 test passing

Test Suites: 1 passed, 1 total
Tests:       21 passed, 21 total
```

Sprint Metrics

Metric	Value
Retrospective Improvements Applied	3/3
New Files Created	2
Files Modified	2
Tests Passing	21/21

Metric	Value
Sprint 2 Commits	5 (more granular than Sprint 1)

Definition of Done Checklist

- [x] Logging implemented for all API operations
- [x] Request logging middleware integrated
- [x] Health endpoint enhanced with monitoring data
- [x] All tests passing
- [x] CI pipeline passing
- [x] Incremental commits (Improvement #2 applied)
- [x] Code committed to main branch

Repository

GitHub: https://github.com/illonaaddae/task-manager-api-assessment