

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 '{"title": "New task"}
```

Log output:

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

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

Commits on Feb 19, 2026		
Update acceptance criteria formatting in SPRINT_1_REVIEW.md	Verified 722f5ed	
illonaaddae authored 4 days ago · ✓ 3 / 3		
Update SPRINT_1_RETROSPECTIVE.md	Verified 491cd0c	
illonaaddae authored 4 days ago · ✓ 3 / 3		
Add Sprint 2 Review and Final Retrospective	111913b	
illonaaddae committed 4 days ago · ✓ 3 / 3		
Enhance health endpoint with monitoring details	ecf3061	
illonaaddae committed 4 days ago		
Add request logging middleware	32470bd	
illonaaddae committed 4 days ago		
Add logging to task controller operations	22a145d	
illonaaddae committed 4 days ago		
Add structured logger utility	156f325	
illonaaddae committed 4 days ago · ✓ 3 / 3		
Add Sprint 1 Review and Retrospective documents	57e2187	
illonaaddae committed 4 days ago · ✓ 3 / 3		
Add README with setup and API documentation	3e610d7	
illonaaddae committed 4 days ago · ✓ 3 / 3		
Add GitHub Actions CI pipeline and comprehensive tests	55e7696	
illonaaddae committed 4 days ago		
Add Express app structure with task CRUD endpoints	5d43b21	
illonaaddae committed 4 days ago		
Initial project setup: npm init, Express, Jest, Supertest	913d652	
illonaaddae committed 4 days ago		

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
Sprint 2 Commits	5 (more granular than Sprint 1)

Definition of Done Checklist

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

Repository

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