MSSE672 – Component Based Software

Week 1 Submission Guide

Student: John Michael Kreski

Instructor: Mohammad Abu Matar

Assignment: Activity 3 – Instrument Project Code with Log4j

Date: 07/06/2025

## Summary

This document demonstrates successful compilation and execution of the updated Triangle Middleware App. For Week 2, logging was implemented across multiple backend layers using Logback, the default logging framework in Spring Boot.

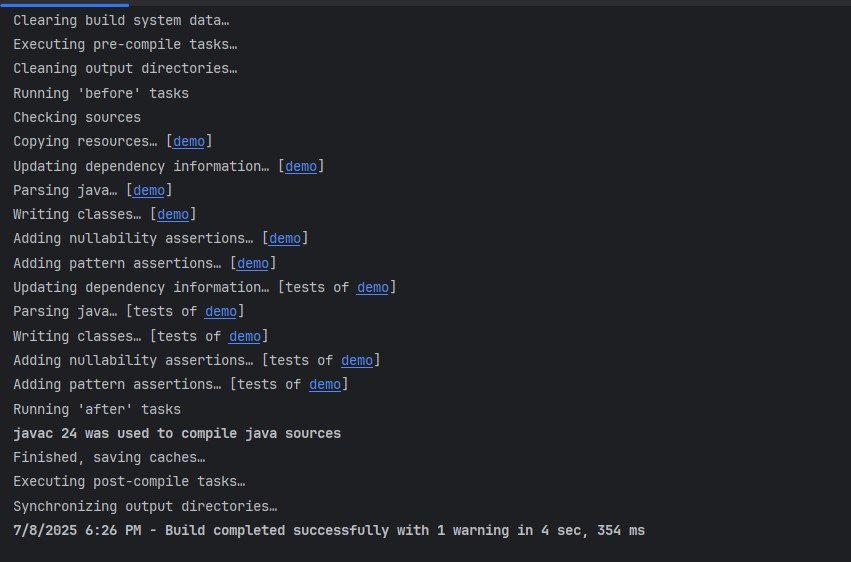
The application builds and runs successfully using IntelliJ IDEA on Windows, with Maven as the build system. Log statements were added in the Service, Controller, and Model layers, and messages appear in both the console and a log file: logs/triangle-app.log.

## Changes this week

* Logging added using SLF4J + Logback.
* logback-spring.xml created to define custom logging format and file output.
* Added logger.info(), logger.debug(), and logger.error() in:
  + QuadService.java (Service layer – in-memory logic)
  + QuadController.java (Controller layer – request handling)
  + Quadrilateral.java (Model layer – shape validation and type detection)
* Confirmed log output in:
  + Console
  + Log file (logs/triangle-app.log)

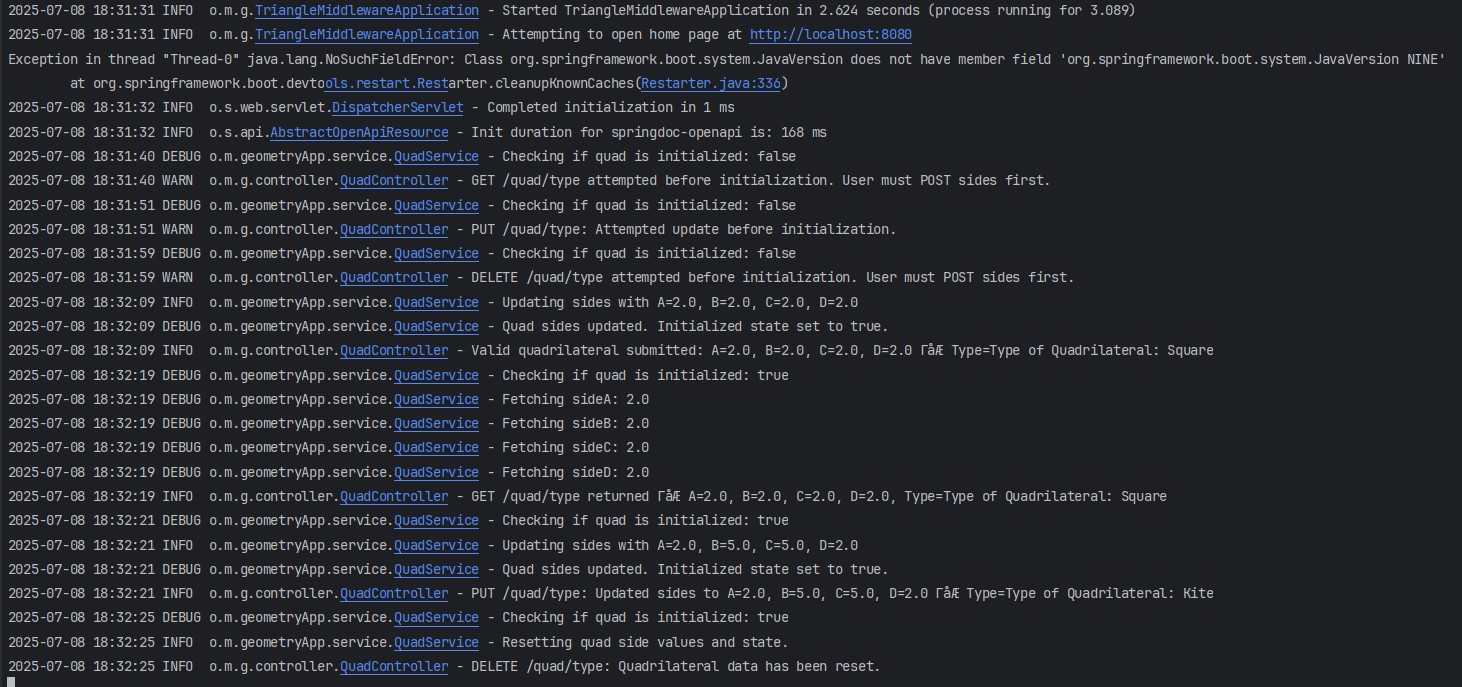
## Screenshot 1 – Successful Compilation

*This screenshot shows successful compilation and no build errors in IntelliJ.*

**

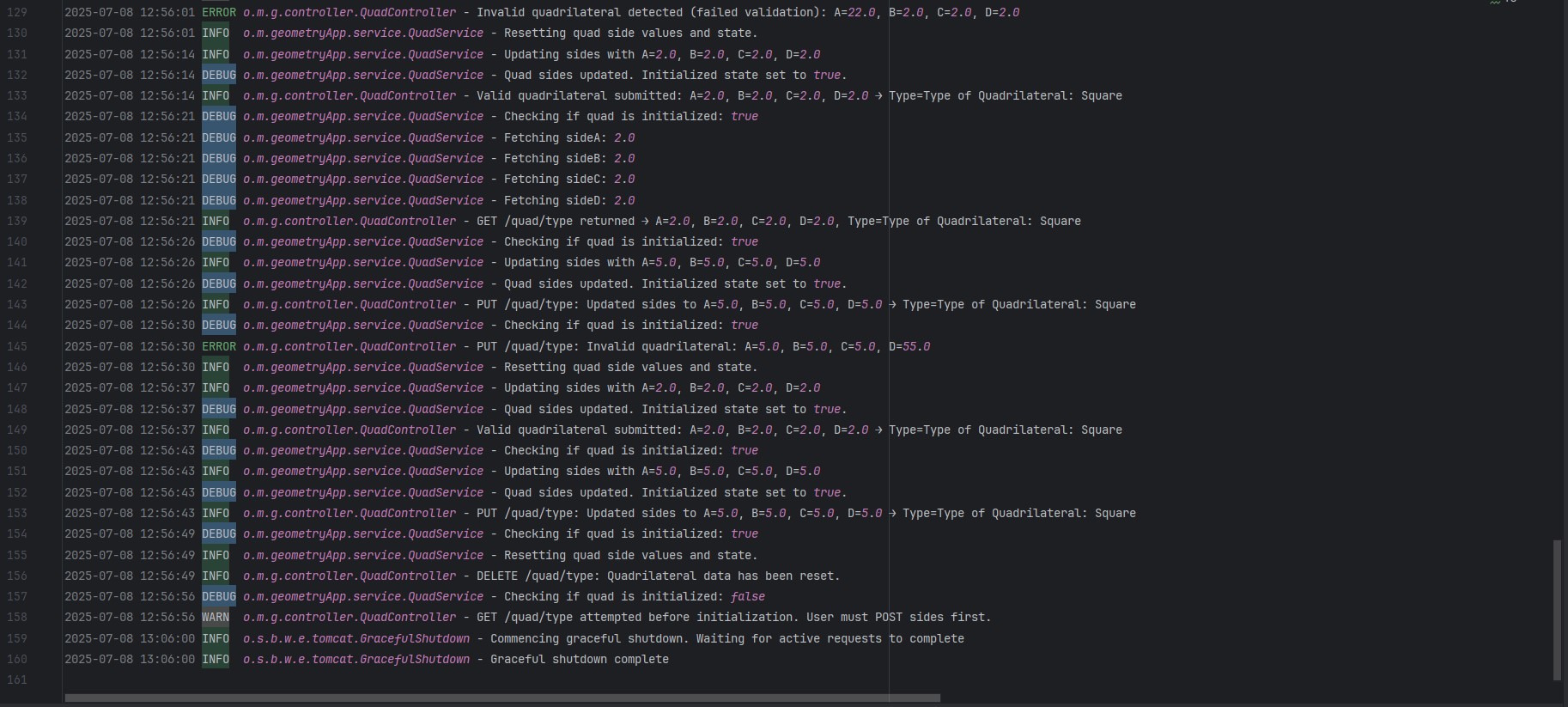
**Screenshot 2 – Logging Output in Console**

*Log messages show correct INFO/DEBUG/ERROR output when validating quadrilaterals.*

**

**Screenshot 3 – Logging Output in Log File**

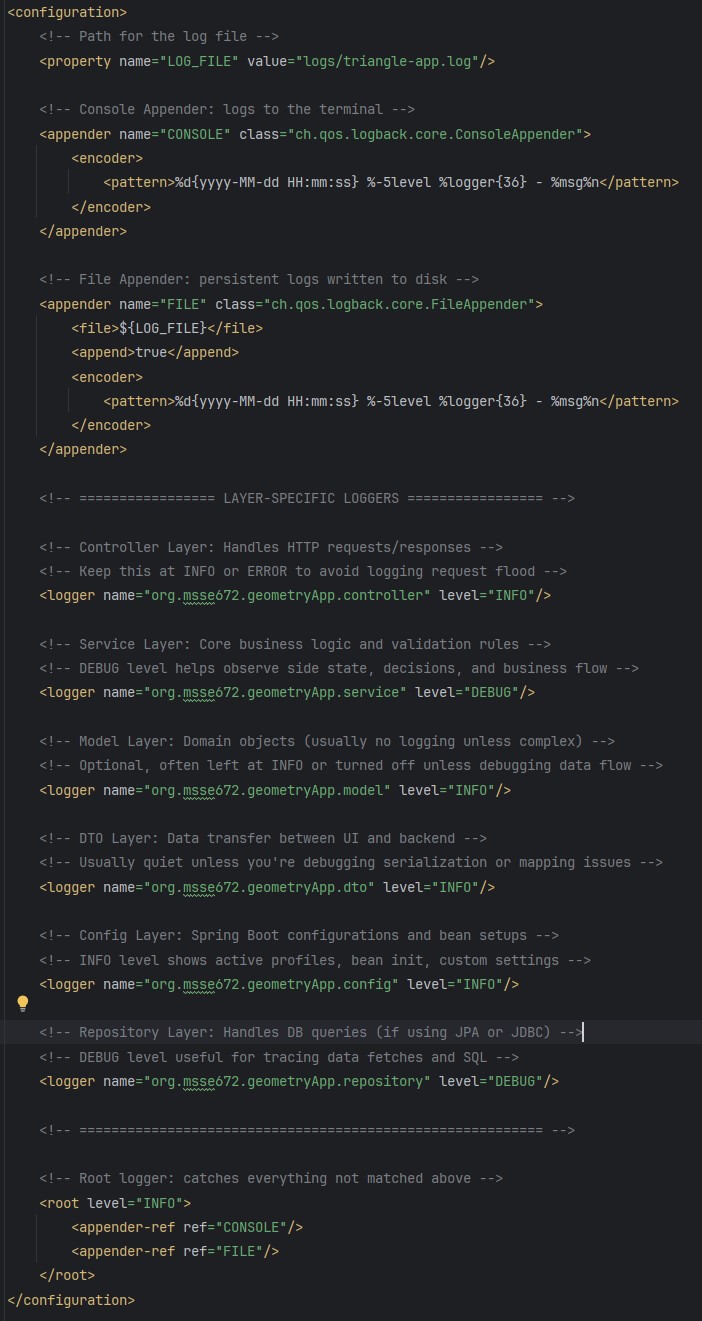
*Custom Logback config directs all logs to logs/triangle-app.log.*

**

**Screenshot 4 – logback-spring.xml**

*Logback config in src/main/resources.*

Key configuration in blue:

****

## Refactored Files

For Week 1, I instrumented the application with logging using Spring Boot’s default logging framework (SLF4J + Logback). The following files were updated:

* QuadService.java – Added logger.info() and logger.debug() to track quad updates and reset actions.
* QuadController.java – Added logger.error() and logger.info() to capture invalid requests and incoming GET/POST/PUT traffic.
* Quadrilateral.java – Added optional logging hooks to trace validation logic (currently minimal, may expand in future weeks).
* logback-spring.xml – Added to src/main/resources to define logging format, output levels, and log file destination (logs/triangle-app.log).

These changes ensure application behavior is observable across all backend layers, supporting future debugging, validation, and testing.

## Notes

- This project will be maintained in a public GitHub repository for version control and learning practice.  
- A `.zip` file containing only `.java` files and the `/Docs` folder is submitted as per course instructions.  
- Screenshots are provided in this single document for clarity, as per the professor’s recent guidance.