**CST-361 Design Patterns in Java**

**Student** **Name**: Michael Weaver

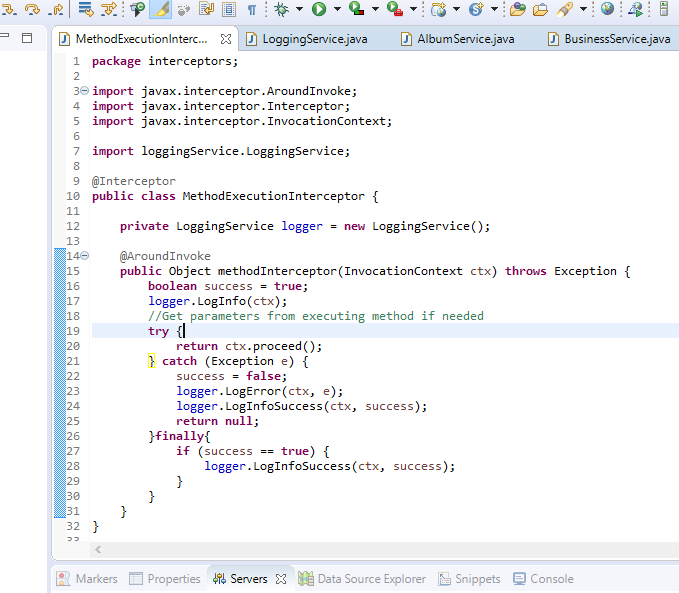
**Project** **Title**: INTEGRATION LAYER PATTERNS 7B

**Date**: 02/03/2019

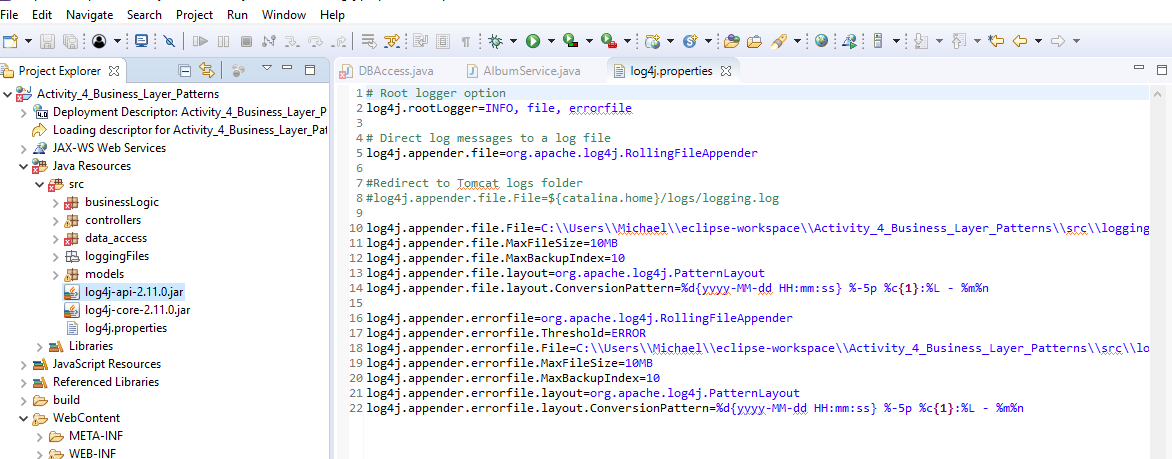
**GitHub URL**: <https://github.com/battousairurik/CST361_Design_Patterns_In_Java>

SCREENSHOTS

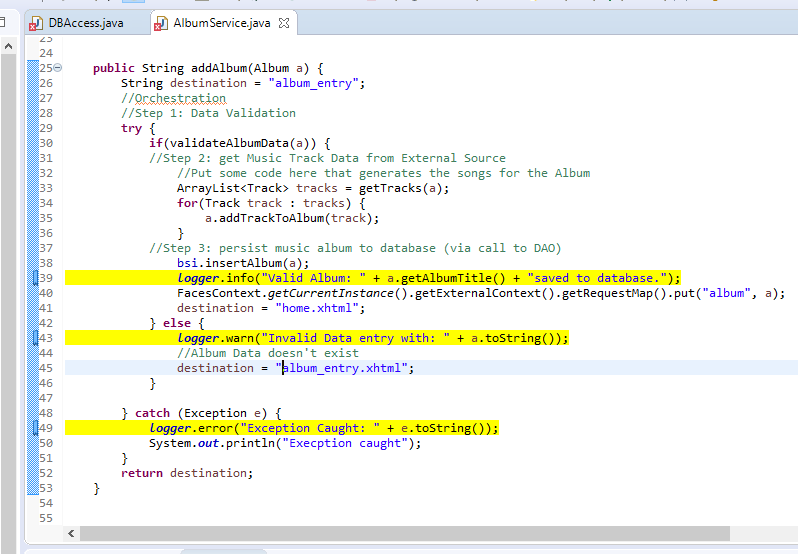
INTERCEPTOR



LOG4J PROPERTIES FILE

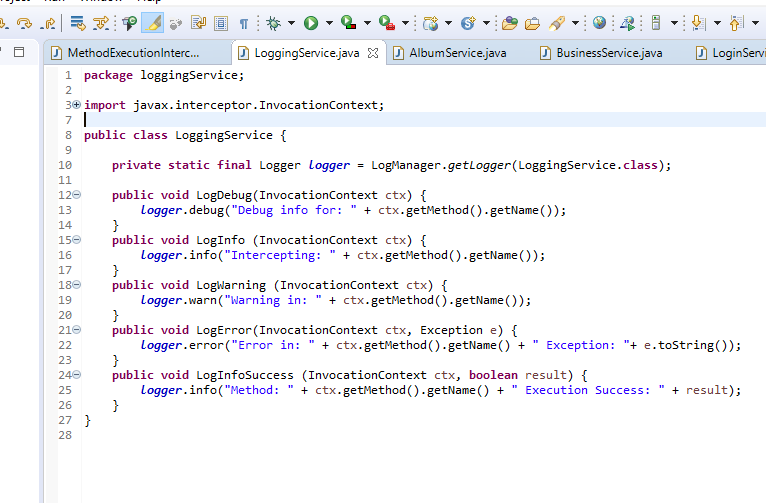


LOGGER IN ALBUM SERVICE



\*Note: I spent roughly 2 hours learning about the logging service which resulted in my first, yet inaccurate attempt. After which I learned how to properly implement the interceptor and logger as a service.

LOGGER SERVICE



INTERCEPTOR DESIGN PATTERN

Each EJB is marked with the @Interceptors annotation, marking it to be handled by my custom MethodExecutionInterceptor. This catches an EJB as it is being processed and then calls a few methods on it. It creates an implementation of the logging service, passes data to the logging service which then logs that data. Depending on the execution of the interceptor, a variety of logging services can be called. This also abstracts the interceptor from the logger logic.

NEW DESIGN PATTERNS USED

Interceptor and Logger are new design patterns used In this activity.