# Exercise 1: Logging Error Messages and Warning Levels

### Objective

Understand how to use the SLF4J logging framework in a Java application to log warning and error level messages based on different runtime conditions.

### Scenario

You are developing a Java service where:  
- A warning should be logged when the input is empty or missing.  
- An error should be logged when an exception occurs during processing.

## Step-by-Step Implementation

### Step 1: Add SLF4J and Logback to pom.xml

<dependencies>  
 <!-- SLF4J API -->  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-api</artifactId>  
 <version>2.0.9</version>  
 </dependency>  
   
 <!-- Logback for SLF4J Implementation -->  
 <dependency>  
 <groupId>ch.qos.logback</groupId>  
 <artifactId>logback-classic</artifactId>  
 <version>1.4.11</version>  
 </dependency>  
</dependencies>

### Step 2: Java Class Using SLF4J Logging

package com.example;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class MyLoggerService {  
 private static final Logger logger = LoggerFactory.getLogger(MyLoggerService.class);  
  
 public void process(String input) {  
 if (input == null || input.isEmpty()) {  
 logger.warn("Input is missing or empty.");  
 return;  
 }  
  
 try {  
 if ("fail".equals(input)) {  
 throw new RuntimeException("Simulated failure");  
 }  
 logger.info("Processing input: {}", input);  
 } catch (Exception e) {  
 logger.error("Error occurred during processing", e);  
 }  
 }  
}

### Step 3: Run the Program

public class Main {  
 public static void main(String[] args) {  
 MyLoggerService service = new MyLoggerService();  
  
 service.process(""); // logs warning  
 service.process("fail"); // logs error  
 service.process("Hello Log"); // logs info  
 }  
}

## Expected Output in Console

WARN Input is missing or empty.  
ERROR Error occurred during processing  
java.lang.RuntimeException: Simulated failure  
 at com.example.MyLoggerService.process(MyLoggerService.java:...)  
INFO Processing input: Hello Log