Spezifikation: Logger und LoggerSetup – Schnittstelle

Version 2.1.0 (HS 2017) / 10. Oktober 2017

Gruber Mischa, Huber Kevin, Padrutt Nino, Salzmann Joel, Wicki Sandro, Arnold Lukas

---

Inhalt

Version	Datum	Author	Bemerkung	Status
1.0.0	30.09.2017	<alle>	Initiale Fassung	Zur Überprüfung
2.1.0	03.10.2017	<alle>	Überarbeitung nach Besprechung	Zur Überprüfung
2.1.0	10.10.2017	Kevin Huber, Nino Padrutt	Bespiel hinzugefügt	Zur Überprüfung

1. Steckbrief

1.1. Logger

Name	**Logger**
Beschreibung	Definiert eine Schnittstelle um Logmeldungen zu loggen.
Typ	Java Interface

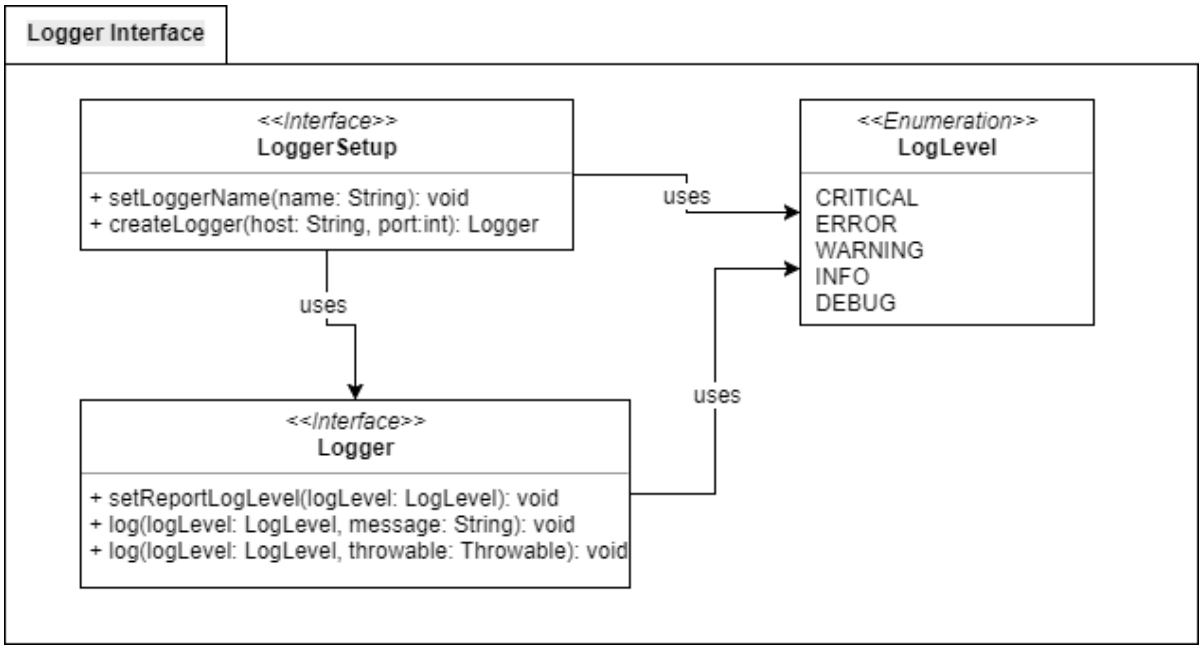
1.2. LoggerSetup

Name	**LoggerSetup**
Beschreibung	Definiert eine Schnittstelle um einen Logger zu konfigurieren und ein Loggerobjekt zu erzeugen.
Typ	Java Interface

1.3. LogLevel

Name	**LogLevel**
Beschreibung	Definiert einen Enum zum klassifizieren der geloggten Meldungen.
Typ	Java Enum

2. Operationen und Datenstrukturen



3. Einsatz, Abläufe, Voraussetzungen und Zusicherungen

3.1. Allgemein

- Im Programm muss der Fully-Qualified-Name des Logger-Package konfiguriert werden können
- Beim Start muss das LoggerSetup dynamisch über den «URLClassLoader» geladen werden.

3.2. Logger

- Um zu verhindern, dass die Anwendung blockiert, muss das Senden der Logmeldungen asynchron implementiert werden.

3.3. LoggerSetup

- Die Methode «getLogger» muss prüfen ob valide Einstellungen für die ServerKommunikation gesetzt worden sind. Andernfalls muss ein Fehler ausgegeben werden.
  Falls nicht anderst definiert, soll der LogLevel auf Warning stehen.
  Die Einstellungen müssen laufend angepasst werden können.
- Mit der Methode «setLogLevel» wird festgelegt ab welchem minimalen Level Meldungen geloggt werden sollen.
- Mit der Methode «setServer» wird festgelegt an welchen Server die Logmeldungen geschickt werden sollen.

3.4. LogLevel

- Die Enumeration beginnt mit Debug=0 und steigt auf bis zu Critical=4

4. Aufbau und Konfiguration

Die Konfiguration des Loggers (z.B. Konfigurationsdatei) ist nicht Bestandteil der Interfaces. Ansonsten keine Konfiguration nötig.

5. Fehlerbehandlung

Die Fehlerbehandlung wird über unchecked Exceptions realisiert. Details siehe JavaDoc.

6. Qualitätsmerkmale

- Es müssen 2000 Logs pro Sekunde gespeichert werden können

7. Entwurfsentscheidungen

- Die Implementierung mit nur einer Methode bietet eine einfache Nutzung der Schnittstelle. Mithilfe des Enums kann bei einem Log vermerkt werden um welche Art von Log-Level es sich handelt.
- Der Datentyp String wurde verwendet um die Implementation zu vereinfachen.
- Die Überladung der Log-Methode mit einer Exception ist dafür da ganze Expections zu Loggen um auf allfällige Bugs aufmerksam machen zu können.

8. Beispielverwendung

```
// Setup
Logger logger;
try {
      Class loadedClass = Class.forName(“fully-qualified-name”);
      LoggerSetup setup = (LoggerSetup) loadedClass;
      setup.setLoggerName = “name”;
      logger = setup.createLogger(host, port);
}
catch (ClassNotFoundException e) {
      e.printStackTrace();
}

// Usage
logger.setReportLogLevel(LogLevel.INFO);
logger.log(LogLevel.INFO, "TestLog”);
logger.log(LogLevel.CRITICAL, exception);
```

9. Änderungsmanagement

Änderungswünsche an dieser Schnittstellendefinition können mit entsprechender Begründung während der Unterrichtszeit angebracht werden. Im Plenum wird dann entschieden ob diese Änderung vorgenommen werden soll oder nicht.