

Nachdenkzettel Logging

.....
Vorname, Name, Matrikelnummer

1. Kennzeichnen Sie in der Config die Stellen wo über das

- was geloggt wird
 - wieviel geloggt wird
 - wo geloggt wird
 - wie geloggt wird
- entschieden wird

```
<Configuration>
  <Appenders>
    <File name="A1" fileName="A1.log" append="false">
      <PatternLayout pattern="%t %-5p %c{2} - %m%n"/>
    </File>
    <Console name="STDOUT" target="SYSTEM_OUT">
      <PatternLayout pattern="%d %-5p [%t] %C{2} (%F:%L) - %m%n"/>
    </Console>
  </Appenders>
  <Loggers>

    <!-- You may want to define class or package level per-logger rules -->
    <Logger name="se2examples.core.businessLogic.VehicleManager" level="debug">
      <AppenderRef ref="A1"/>
    </Logger>
    <Root level="debug">
      <AppenderRef ref="STDOUT"/>
    </Root>
  </Loggers>
</Configuration>
```

↙ konfigurierter Level

default-
level →

1.2 Wie würde man erreichen, dass für alle Klassen innerhalb eines Packages ein spezieller Loglevel gelten würde? Könnte man auch alle Klassen eines Packages in ein anderes File loggen?

→ mit `<Root level=" " ">` → ja geht
`</Root>`

2. Geben Sie je ein Beispiel wann Sie den loglevel

- error → zu wenig Verbindungen zur DB konfiguriert
 - info → Infos zum Anwendungsstatus → Appl. gestartet → Kunde
 - debug → Fehleranalyse
- verwenden

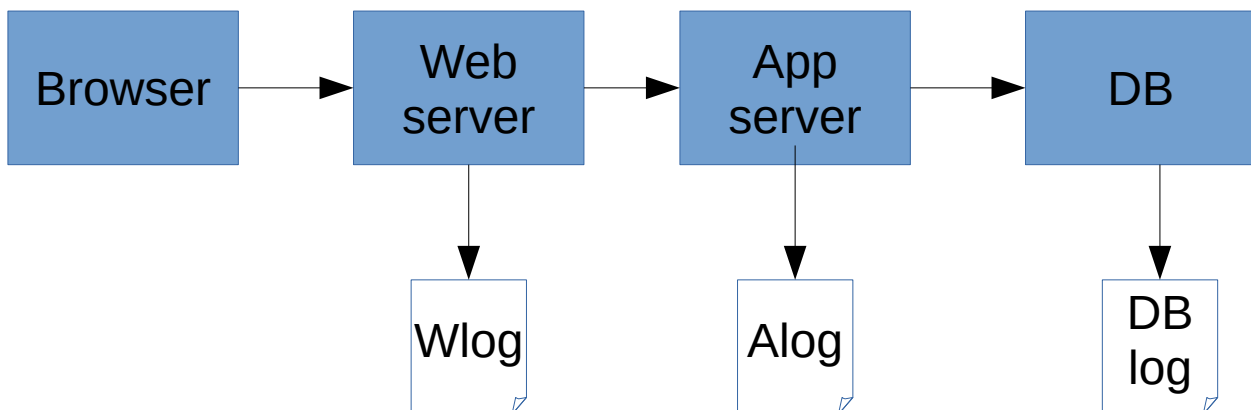
3. Sie verwenden einen FileAppender für das Logging. Jetzt soll Ihre Application im Datacenter laufen. Was machen Sie mit dem FileAppender?

→ File Appender wird „ausgeschaltet“
⇒ Grund: Files werden mit Appender nicht überschrieben und somit sehr groß in einem Datacenter

4. Macht logging Ihre Application langsamer? Was passiert wenn Sie `log.debug(„foobar“);` aufrufen? Wie sollte sich das Logging Subsystem verhalten?

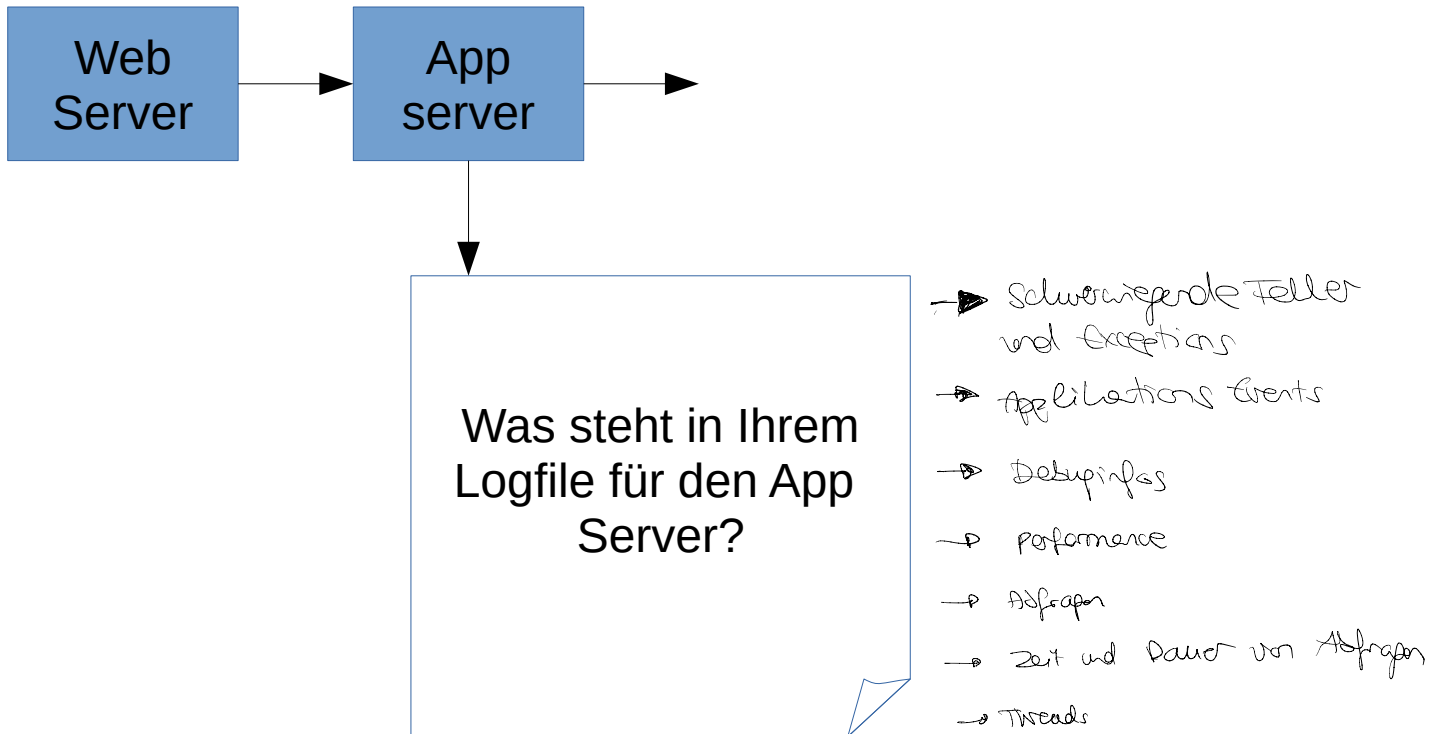
→ Ansicht: ja Grund: produzierten große Datenmengen
→ je weniger Log-Infos desto schnellere Ergebnisse hat man

5. Ein Request an Ihre Application durchläuft einen Proxy Server, dann einen Web Server, dann einen Application Server und dann die Datenbank. Auf jedem Server loggen Sie die Requests. Welches Problem tritt auf?



⇒ großen Datenmengen entstehen

6. Was sollten Sie pro Komponente/Tier loggen?



7. Aus Geschwindigkeitsgründen halten Sie teure DB-Connections auf Vorrat in einem Pool. Jeder Request vom Client braucht dann eine Connection. Der Pool hat die Methoden:
`DB Connection con = ConnectionPool.getConnection();`
`ConnectionPool.freeConnection(DBConnection dbCon);`

Was loggen Sie in Ihrem App Server? Oder anders gefragt: Was wollen Sie beim Umgang mit dem Pool als Software-Architektin wissen?

8. Sie fügen log-statements in die Login-Klasse ein. Was müssen Sie unbedingt beachten???

Tipp: Denken Sie über Userverhalten nach. Und über Mitarbeiter....

→ Passwort sollte nicht geloggt werden