

Nachdenkzettel Logging

Toni, Johanna, Lisa, Lisa
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>
```

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?

```
<Logger name="random.package" level="debug">
  <AppenderRef ref="randomAppender"/>
</Logger>
```

ja

```
<Logger name="another.package" level="info">
  <AppenderRef ref="anotherFileAppender"/>
</Logger>
```

2. Geben Sie je ein Beispiel wann Sie den loglevel

- error file not found
 - info successfully completed a task
 - debug info that is diagnostically helpful (not just for developers but also system administrators, etc.)
- verwenden
- ↳ 'parsing browsers from query'
- 'parsed browsers: ...'

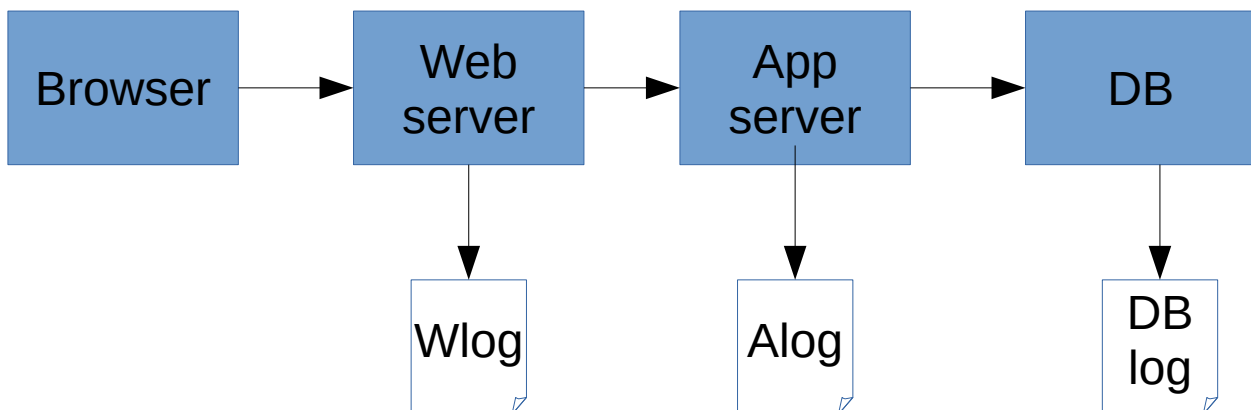
3. Sie verwenden einen FileAppender für das Logging. Jetzt soll Ihre Application im Datacenter laufen. Was machen Sie mit dem FileAppender?

Switch to RollingFileAppender to prevent log files from getting too big

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

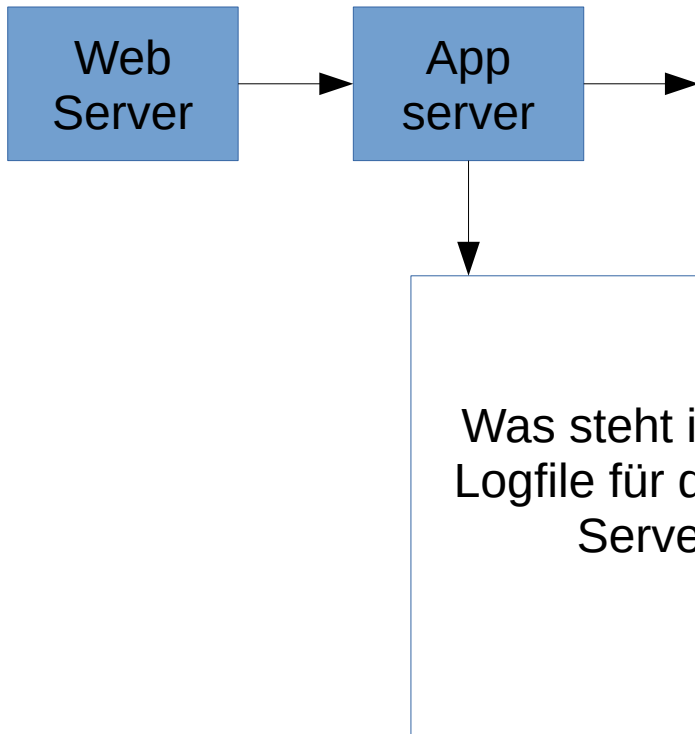
Yes it slows down the application, however the benefits of logging are more significant.

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?



The log data is split into multiple files which makes debugging more tedious as you'd have to find the corresponding log entries in the different log files.

6. Was sollten Sie pro Komponente/Tier loggen?



- *http-requests, status codes*
- *incoming connections (webserver)*
- *app server details: exceptions, loading data, loading times*

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?

- *number of available database-connections*
- *established connections*

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

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

- *do not log credentials like password etc.*
- *log failed login-attempts*
- *log unexpected errors to then be able to fix them*
- *who accessed which resource + when?*