

Nachdenkzettel Logging

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>
```

1.2.1 Wie würde man erreichen, dass für alle Klassen innerhalb eines Packages ein spezieller Loglevel gelten würde?

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

1.2.2 Könnte man auch alle Klassen eines Packages in ein anderes File loggen?

Yes:

```
<Logger name="some.different.package" level="info">
  <AppenderRef ref="SomeDifferentFileAppender"/>
</Logger>
```

2. Geben Sie je ein Beispiel wann Sie den loglevel
- error

A selected file does not exist.

- info

A server instance has been successfully loaded.

- debug

If you want to log a specific detail that could come in handy when debugging.

verwenden

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

You should change to a RollingFileAppender so that the log files won't get bigger indefinitely.

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, ideally it shouldn't. There is a way so make Loggers run asynchronously, but that comes with different tradeoffs:

<https://logging.apache.org/log4j/2.x/manual/async.html>

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?

- logs are split into multiple files thus it takes more time to debug an error. I.e. if you have many requests per second you can not guarantee that you'll find the exact corresponding log entry in the DB-Log for a log entry in the Application Server Log

6. Was sollten Sie pro Komponente/Tier loggen?

- log incoming connections at web server
- log details in app server (both of app server and the database. Otherwise you'll probably never find out what caused the log entry in the database log)

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?

- How many DBConnections are free (or are in use)
- When a connection gets used and freed

(- DB's logs as mentioned in 5)

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

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

- Don't log credentials
- Log invalid login attempts as thoroughly as possible → In case, someone tries to get hold of an account he does not own.
- Don't log login attempts where a field is empty (these shouldn't even reach the backend...)