

Nachdenkzettel Logging

.....
Vorname, Name, Matrikelnummer

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

- was geloggt wird **I**
 - wieviel geloggt wird **I**
 - wo geloggt wird **I**
 - wie geloggt wird **I**
- 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>
```

2. Geben Sie je ein Beispiel wann Sie den loglevel

- error **beim try catch block von einem Error**
 - info **Das Programm hat etwas erfolgreich ausgeführt**
 - debug **Zwischenstatus von Variablen (Ersatz für System.out.println())**
- verwenden

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

verlinken?

tracken?

(nach einer Weile die Daten löschen)

(Disc)

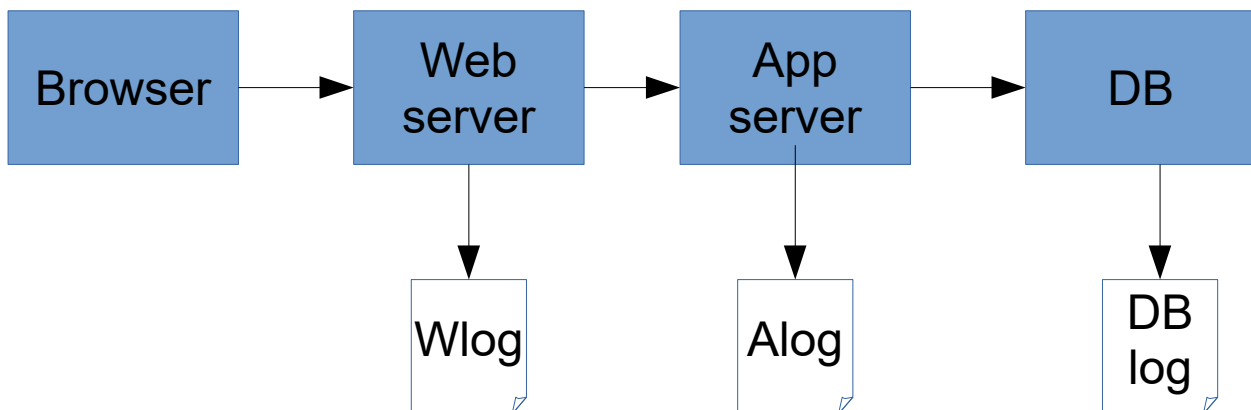
Rolling- oder Netzwerk FileAppender, damit Daten nicht zu groß werden.

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

nur sehr geringfügig -> meist mehr Nutzen als Nachteil.

Sollte es in die Konsole ausgeben

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?

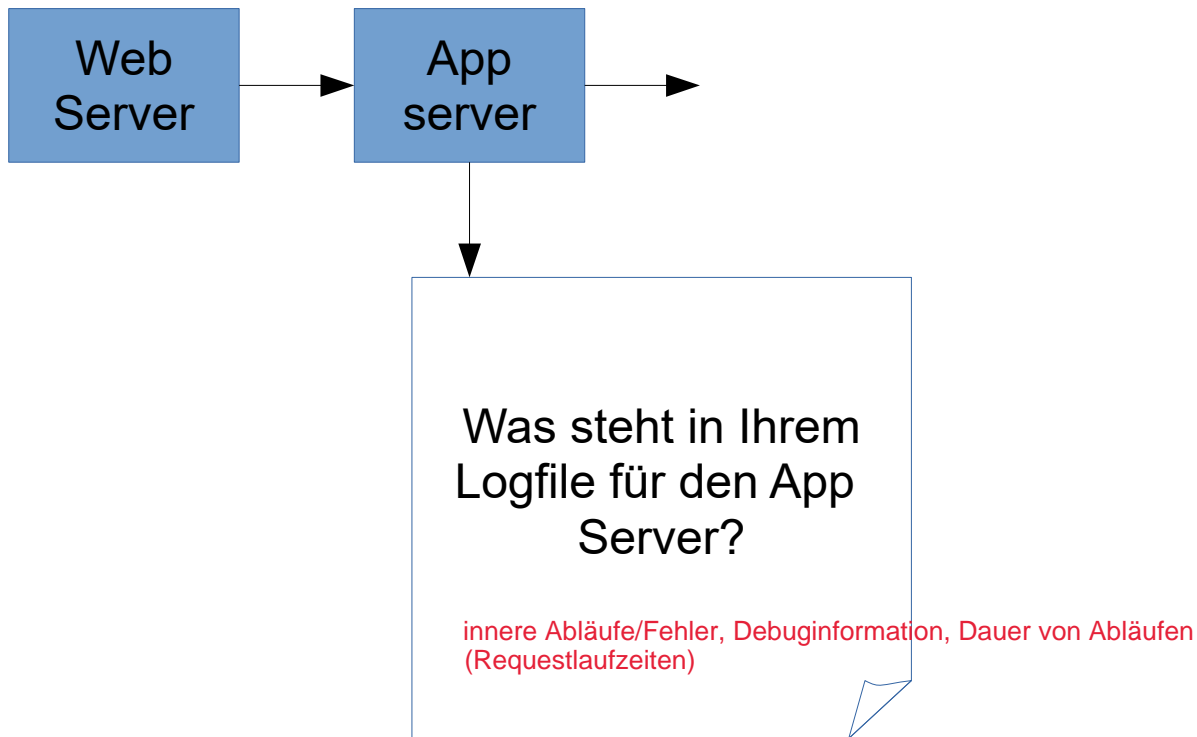


Speicherkapazität

Eventuell verwirrend, wegen Dopplungen

Eine ID am Anfang, die durchfließen soll -> sonst schwer Logs nicht mehr zusammenhängend

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?

Wird überhaupt/richtig rausgelesen. getConnection gibt was zurück?

Erfolg der 2. Methode

Wenn diese Ressource genommen wird muss Protokolliert werden wer sie bekommen und aktualisiert hat.

Muss ein Thread zu lange warten? Hält eine Anwendung sie zulange fest?