

SQL Logging

🌐 Sprache: **Deutsch** · [English](#)

🔧 SQL Logging

SQL-Statements protokollieren – per REST/Server-Info zur Laufzeit oder über log4j2, inkl. SQLTimer und StackTrace.

HOW-TO

ADMINISTRATOR

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

Auf dieser Seite

- [Zur Laufzeit ein-/ausschalten \(Webclient\)](#)
- [Konfiguration in log4j2.xml](#)
- [StackTrace-Logging](#)
- [Weiterführend](#)
- [Verwandte Seiten](#)

SQL-Logging protokolliert die vom Server ausgeführten SQL-Statements – ideal zur Performance-Analyse.

Zur Laufzeit ein-/ausschalten (Webclient)

Ab 4.18.2/4.19.0 kann der Super-User das Logging im laufenden Betrieb umschalten: *Zahnrad Server Info*, Log-Level des gewünschten Loggers auf **DEBUG** (zum Abschalten wieder **INFO**).

Per REST

```
curl --cookie-jar cookies.txt http://localhost:8080/nuclos-war/rest -X POST -H "Content-Type: application/json" -d '{"username":"'nuclos'", "password":""}';
curl --cookie cookies.txt http://localhost:8080/nuclos-war/rest/maintenance/logging/SQLTimer/level -X PUT -H "Content-Type: application/json" -d '{"level":"'debug'"}';
```

Logger-Namen: SQLLogger (vor Ausführung), SQLTimer (nach Ausführung inkl. Laufzeit), SQLUpdate (nur UPDATE/INSERT/DELETE).

Konfiguration in log4j2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
  <Appenders>
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout pattern="%d %p [%c] - %m%n"/>
    </Console>
    <RollingFile name="Logfile" fileName="<Nuclos-Log-Verzeichnis>/server.log"
      filePattern="<Nuclos-Log-Verzeichnis>/server-%i.log" append="true">
      <PatternLayout pattern="%d %p [%c] - %m%n"/>
      <Policies>
        <SizeBasedTriggeringPolicy size="5 MB"/>
      </Policies>
      <DefaultRolloverStrategy max="20"/>
    </RollingFile>
  </Appenders>
  <Loggers>
    <Logger name="org.apache.log4j.xml" level="info"/>
    <Logger name="SQLLogger" level="debug"/>
    <Root level="info">
      <AppenderRef ref="Console"/>
      <AppenderRef ref="Logfile"/>
    </Root>
  </Loggers>
</Configuration>
```

Für Laufzeitmessung zusätzlich den SQLTimer auf debug setzen. Typische Ausgabe:

```
2018-01-04 18:10:54,844 DEBUG [SQLTimer] - SELECT COUNT (t.INTID) FROM D2SC_FH_LASTPOSITION t WHERE 1=1
=[ ]=(2 ms)
```

StackTrace-Logging

Optional lässt sich der Server- und Client-Stacktrace bis zur Ausführung mitloggen – gefiltert per Textsuchbegriff (Server Info) oder in der Klasse DataSourceExecutor via setDebugSQL("SELECT COUNT").



Hinweis

Der **Client**-Stacktrace ist nur verfügbar, wenn der Server im Entwicklungsmodus (-Dfunctionblock.dev=true) läuft.

Weiterführend

- [SQL-Logging in eine eigene Datei](#)
- [SQL-Logging für alte Versionen \(bis 4.5\)](#)

Verwandte Seiten



SQL Logging in separates Log-File

Eigene Datei.

[Öffnen](#)



Logging

Übersicht.

[Öffnen](#)