SQL logging

Language: Deutsch · **English**

SQL logging

> **Log SQL statements – at runtime via REST/Server Info or via log4j2, including SQLTimer and stack trace.**

HOW-TO | ADMINISTRATOR | UPDATED: JUL 2026 | APPLIES TO NUCLOS 4.2026.X

On this page

SQL logging records the SQL statements executed by the server – ideal for performance analysis.

Toggle at runtime (web client)

From 4.18.2/4.19.0 the super user can toggle logging while running: *gear Server Info*, set the log level of the desired logger to **DEBUG** (back to **INFO** to disable).

Via 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 names: `SQLLogger` (before execution), `SQLTimer` (after execution incl. runtime), `SQLUpdate` (only UPDATE/INSERT/DELETE).

Configuration 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>
```

For runtime measurement additionally set `SQLTimer` to `debug`. Typical output:

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

Stack trace logging

Optionally the server and client stack trace up to execution can be logged – filtered by a text search term (Server Info) or in the `DataSourceExecutor` class via `setDebugSQL("SELECT COUNT")`.

> **Note**
>
> The **client** stack trace is only available if the server runs in development mode (`-Dfunctionblock.dev=true`).

See also

- SQL logging to a separate file
- SQL logging for old versions (up to 4.5)

Related pages

SQL logging to a separate log file

Separate file.

Open

Logging

Overview.

Open