

# Java 17 (experimentell)

Diese Java LTS Version wird vom Tomcat 9.x wie auch Spring 5.x unterstützt, jedoch nicht von neueren Versionen oder dem Payara 5 Glassfish Server (nur bis Java 11).

Diese Java LTS-Version wird von Nuclos erstmalig ab Version 4.2023.31 unterstützt ([NUCLOS-10398](#)).

Aktuell noch BETA

## Tests

Die automatischen Webclient- und Server-Integrationstest laufen mit Java 17, jedoch befindet sich der Richclient und insbesondere die Konfiguration eines Nuclets noch in der manuellen Testphase. Wenn Ihnen etwas auffällt würden wir uns über eine Rückmeldung oder auch ein Ticket sehr freuen: <https://support.nuclos.de>

## HiDPI

Das von Nuclos eingesetzte Richclient Look&Feel unterstützt zwar Java bis max v17, allerdings wird möglicherweise Windows eine unsaubere bzw. verschwommene Benutzeroberfläche erzeugen, wenn von Anwendungen eine Skalierung gefordert wird. Die allgemeine Windows Bildschirm Einstellung "**Größe von Text, Apps und anderen Elementen ändern**: ??%" (empfohlen)" bei Werten > 100% und < 200% ist dafür verantwortlich. Sollte es nicht zu klein sein (Test lohnt sich!) können Sie nur den Nuclos Richclient mit der VM Option **sun.java2d.uiScale.enabled** auf 100% setzen, unabhängig von der Windows Bildschirm Einstellung. Im Launcher fügen Sie das Argument "-Dsun.java2d.uiScale.enabled=false" (mit den Anführungszeichen!) unter Einstellung > Java Konfiguration > **JAVA\_VMARGS** hinzu und starten den Client neu. Alternativ für sehr hochauflösende 4k Notebook Displays, und einer z.B. Windows Einstellung von 175%, könnten Sie nur Nuclos auf 200% mit dem folgenden Argument setzen: "-Dsun.java2d.uiScale=2.0"

Wir evaluieren derzeit Look&Feel Updates oder auch einen Wechsel um diese HiDPI Funktionalität unterstützen zu können.

## Fehlende Reflection Argumente

Sollten Sie Meldungen der Art

```
java.lang.reflect.InaccessibleObjectException: Unable to make protected void java.net.URLClassLoader.addURL(java.net.URL) accessible: module java.base does not "opens java.net" to unnamed module...
```

erhalten, können Sie uns das melden und wir prüfen ob wir das nötige Argument fest im Nuclos Kern hinterlegen. Übergangsweise, und evtl. auch dauerhaft für Drittbibliotheken die Nuclets/Extensions fordern und nichts mit dem Nuclos Kern zu tun haben, können Sie diese Ausnahme im Server wie auch im Richclient selbst hinterlegen:

- Für den Nuclos Server über den Installer mittels **Additional JVM arguments** im Bereich der Server Konfiguration, Wert (Beispiel): `--add-opens=java.base/java.net=ALL-UNNAMED`
- Für den Nuclos Richclient über den Launcher mittels **JAVA\_VMARGS** (Einstellung > Java Konfiguration), Wert (Beispiel): `--add-opens=java.base/java.net=ALL-UNNAMED`

## Integrationstest

Um eine möglichst hohe Kompatibilität zu gewährleisten wurden die Integrationstest umgestellt (Stand Oktober 2023 / Nuclos 4.2023.31ff)

Nuclos	Apache Tomcat	Payara Glassfish
bis 4.2023.30	nur Server (JAVA_HOME)	Server (JAVA_HOME) + Webclient
ab 4.2023.31	Server (java.17.home) + Webclient	nur Server (JAVA_HOME)

`java.17.home` (Property in `nuclos/pom.xml`)

Vorbelegt mit `/usr/lib/jvm/java-17-openjdk-amd64` (Standard unseres Jenkins / Ubuntu), kann aber für lokale Testläufe mittels `mvn -Djava.17.home=/path/to/java/17/home ...` überschrieben werden.

## Entwicklungsumgebung

Zum Starten von Server und Client aus einer Entwicklungsumgebung sind weitere VM Options mit anzugeben:

VM Options
------------

Tomcat 9.x	--add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.net=ALL-UNNAMED --add-opens=java.desktop/java.awt=ALL-UNNAMED --add-opens=java.desktop/javafx.swing.tree=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
Richclient	--add-exports=java.desktop/sun.swing=ALL-UNNAMED --add-exports=java.desktop/sun.swing.table=ALL-UNNAMED --add-exports=java.desktop/sun.swing.plaf.synth=ALL-UNNAMED --add-opens=java.desktop/javafx.swing.plaf.synth=ALL-UNNAMED --add-opens=java.desktop/javafx.swing.plaf.basic=ALL-UNNAMED --add-opens=java.desktop/javafx.swing=ALL-UNNAMED --add-opens=java.desktop/javafx.swing.tree=ALL-UNNAMED --add-opens=java.desktop/java.awt.event=ALL-UNNAMED --add-exports=java.desktop/sun.awt.shell=ALL-UNNAMED --add-exports=java.base/sun.security.action=ALL-UNNAMED --add-exports=java.desktop/com.sun.swing.internal.plaf.basic.resources=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.desktop/java.awt=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED

Ein Nuclos Installer Lauf setzt dies für den Tomcat automatisch, und ein laufender Server gibt es beim Download der JNLP für die Clients mit.

Siehe auch	Im Code	BitBucket Link
Installer	Klasse <code>org.nuclos.installer.Config</code> Variable <code>sEnvironmentJvm</code>	<a href="#">Config.java</a>
Server	Resource <code>jnlp.xml</code> Attribute <code>java-vm-args</code>	<a href="#">jnlp.xml</a>