

# Tutorial: Einbinden von Nuclos in IntelliJ IDEA

## Lernziel

In diesem Tutorial lernen Sie, wie Sie ein Nuclos-Projekt in die Entwicklungsumgebung **IntelliJ IDEA** einbinden.

## Voraussetzungen

- Funktionierende Installation des Nuclos-Projekts (im folgenden **Projekt-Installation**).
- JetBrains IntelliJ IDEA Ultimate: <https://www.jetbrains.com/idea/download/#section=windows>
- Maven
- Git
- Java

## Sonstiges

- Troubleshooting



DRAFT

marked by Claudia  
Hintermair on Nov 25,  
2019 09:55

Dokumentenstatus

---

Unter Windows muss Git für lange Dateinamen konfiguriert werden:

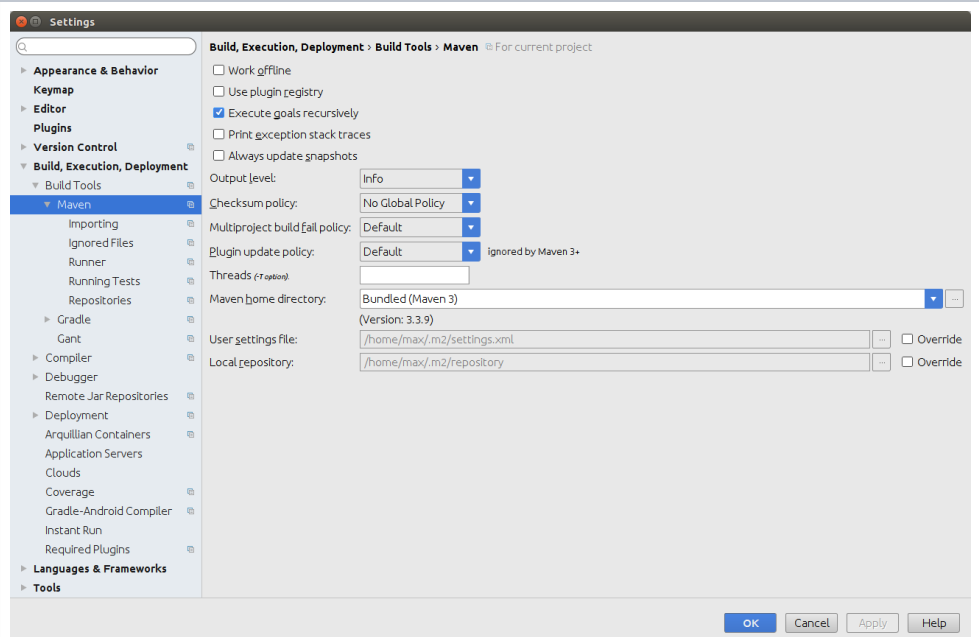
```
git config --system core.longpaths true
```

Bevor Sie mit der Konfiguration von IDEA beginnen, müssen Sie sowohl das git-Repository des Projekts (im folgenden **Projekt-git**), als auch das git-Repository des Nuclos-Kerns (im folgenden **Kern-git**) klonen.

Wechseln Sie nun im git zu dem Tag des Kerns, welcher der Nuclos-Version der Projekt-Installation entspricht (git checkout).

Nun sind Sie bereit, die Konfiguration von IDEA anzugehen.

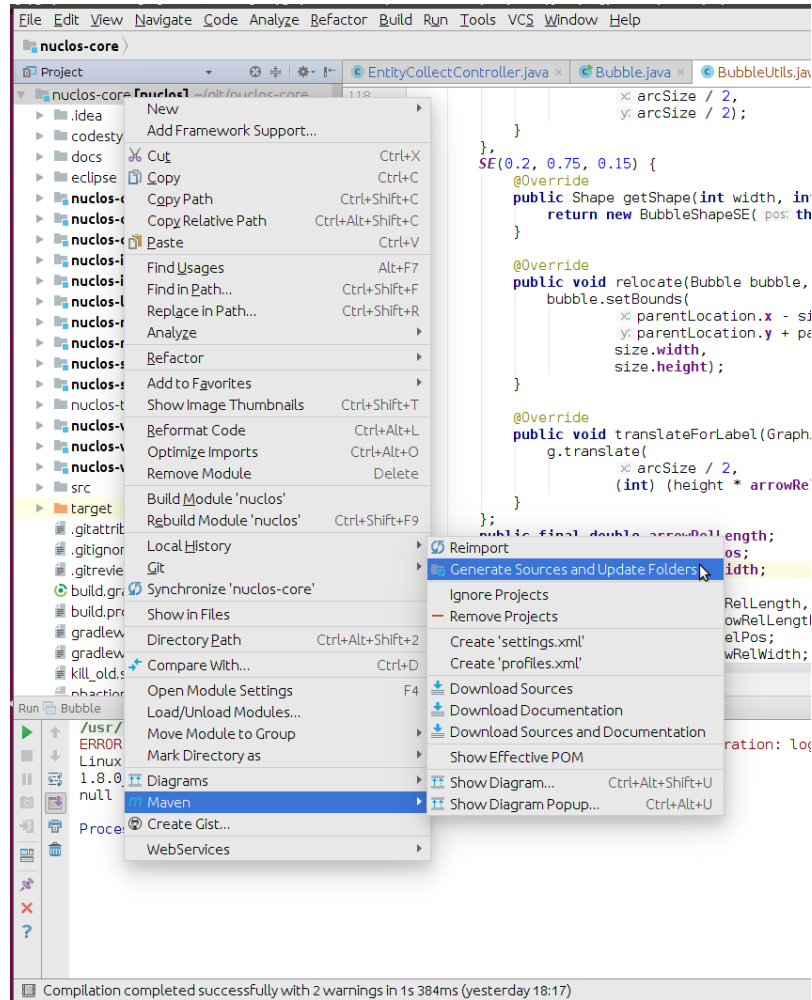
Richten Sie maven ein: File -> Settings -> Build, Execution, Deployment -> Build Tools -> Maven



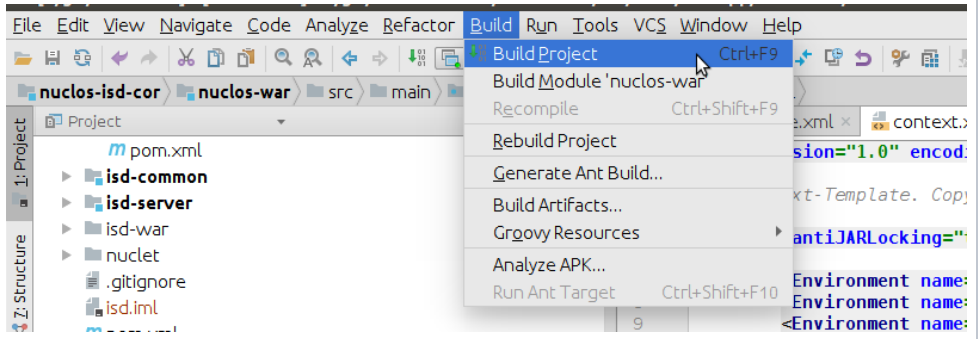
Öffnen Sie das Projekt (**Projekt-git**) in IDEA: File -> Open...

Importieren Sie das Maven Projekt (ein Popup sollte erscheinen).

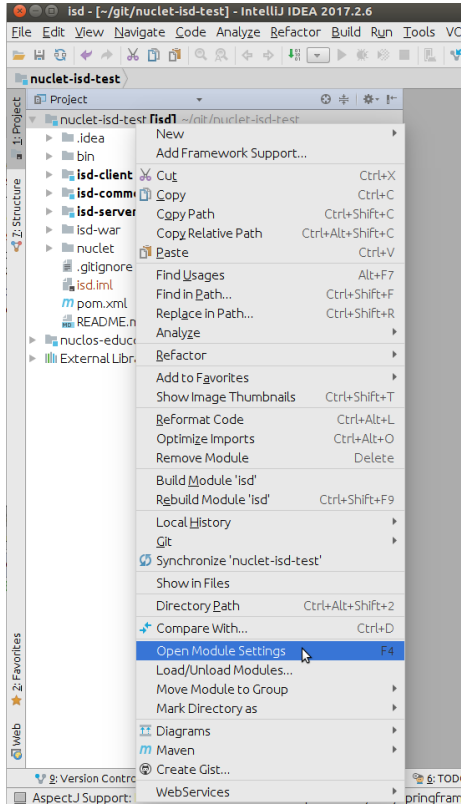
Lassen Sie Maven die Quellen generieren.



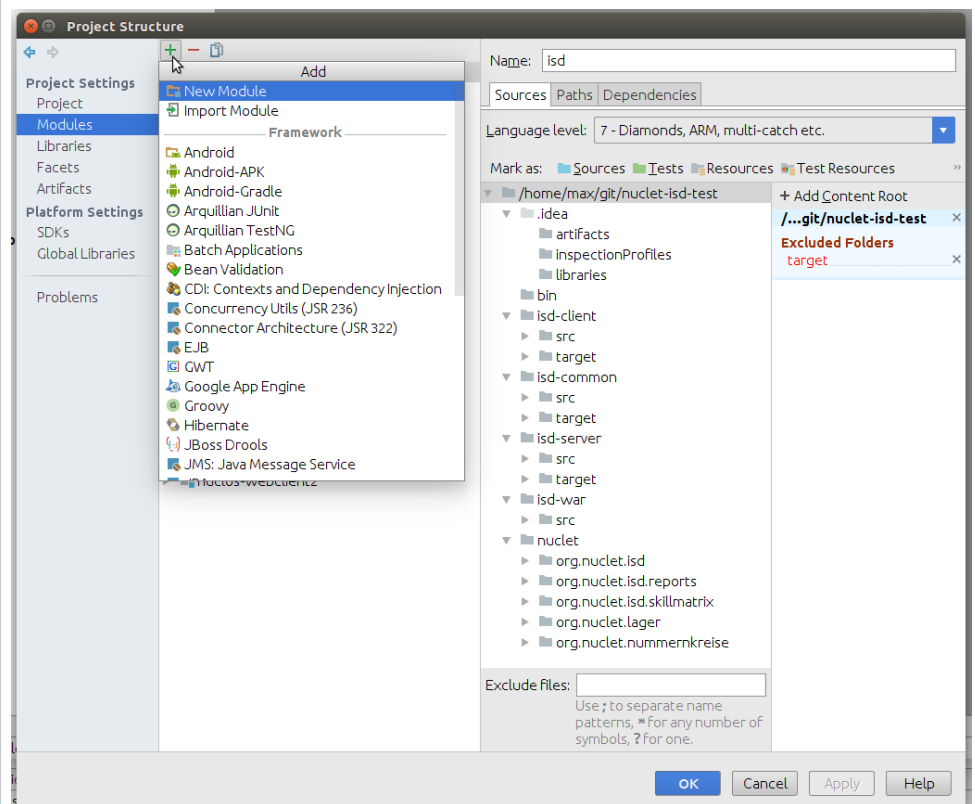
Bauen Sie das Modul.



Nun kann der **Kern-git** eingebunden werden. Öffnen Sie zu diesem Zweck die Modul-Einstellungen:



Hier kann der **Kern-git** importiert werden (*Import Module*):

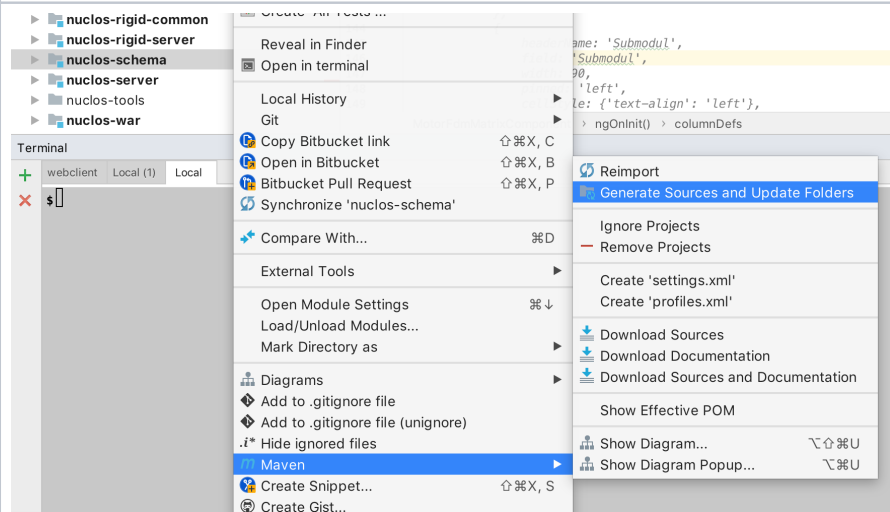


Einige Quellen werden in Nuclos generiert und sind nicht im Git eing检ekt.

Über:

"nuclos-schema" > rechte Maustaste > Maven > Generate Sources and Update Folders

werden die Quellen generiert.



Sobald das Modul geladen ist, navigieren Sie ins Verzeichnis `~/Pfad zu Kern-git/nuclos-war/src/main/webapp/META-INF/`.

Hier finden sie eine Datei `context-template.xml` - Erstellen Sie eine Kopie der Datei im selben Verzeichnis und nennen Sie diese `context.xml`.

Passen Sie nun die vier **value**-Parameter (von `context.xml`) an, so dass diese auf die entsprechenden Konfigurationsdateien zeigen, die sich im Verzeichnis ihres installierten Nuclos-Projekts (**Projekt -Installation**) befinden.

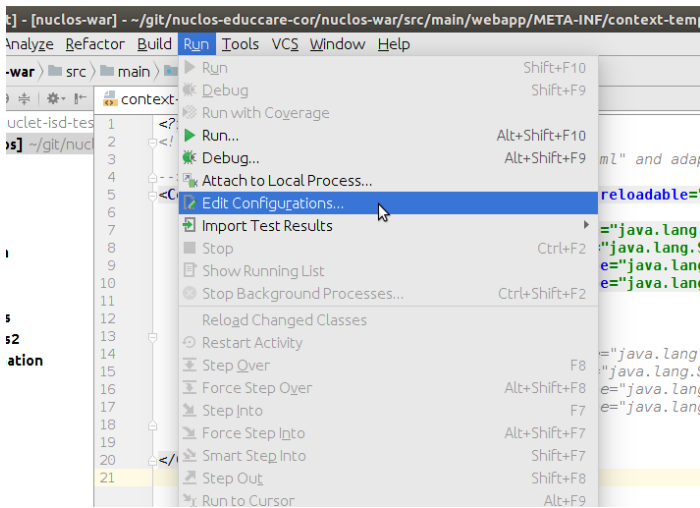
Nun können Sie auch für den Kern die Quellen mit Maven importieren und das Modul bauen.



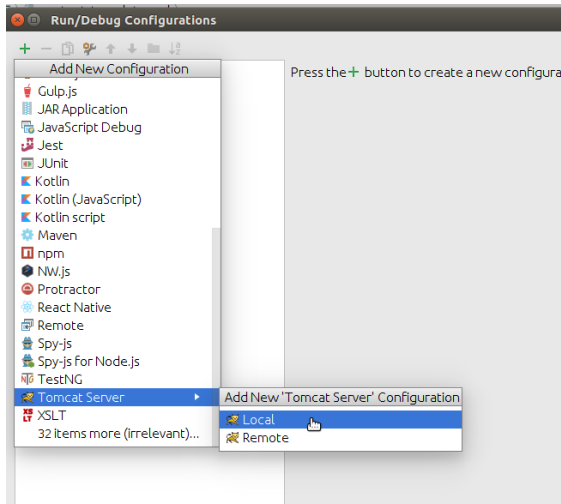
Anmerkung: Falls es im Extension-Projekt eine `extension-war` gibt, hat es sich bewährt, die `context.xml` dort zu erstellen. also im Verzeichnis `~/Pfad zu Project-git/<extension>-war/src/main/webapp/META-INF/`.

```
context-template.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3 Context-Template. Copy this file to "context.xml" and adapt the following paths to your system.
4 -->
5 <context antiJARLocking="true" path="/nuclos-war" reloadable="false" sessionCookiePathUsesTrailingSlash="true">
6
7
8 <Environment name="nuclos-conf-log4j" type="java.lang.String" value="file:///home/user/nuclos/nuclos-conf/conf/log4j2.xml"/>
9 <Environment name="nuclos-conf-jdbc" type="java.lang.String" value="file:///home/user/nuclos/nuclos-conf/conf/jdbc.properties"/>
10 <Environment name="nuclos-conf-quartz" type="java.lang.String" value="file:///home/user/nuclos/nuclos-conf/conf/quartz.properties"/>
11 <Environment name="nuclos-conf-server" type="java.lang.String" value="file:///home/user/nuclos/nuclos-conf/conf/server.properties"/>
12 </context>
```

Um den Server einzurichten, gehen Sie auf `Run -> Edit Configurations...`

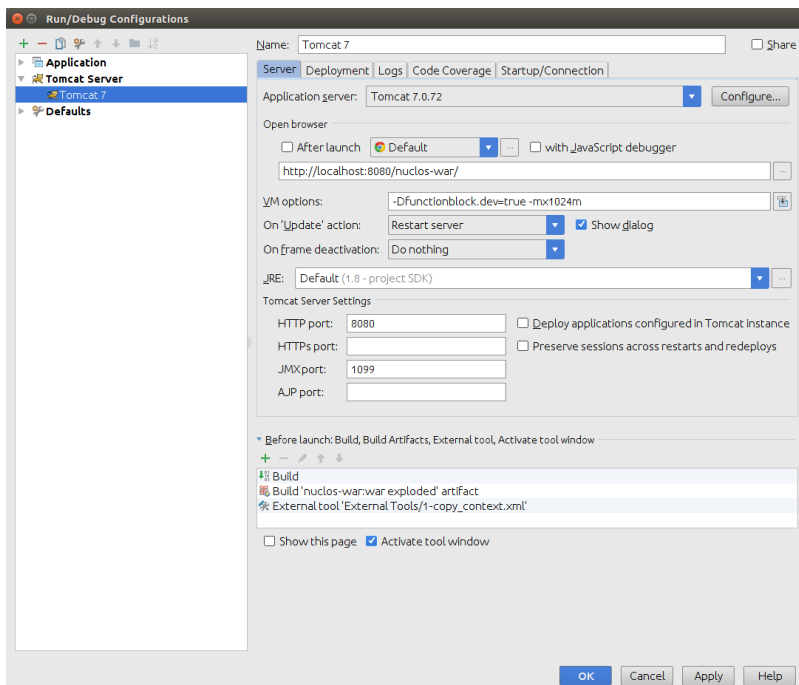


Erstellen Sie einen neuen Tomcat-Server:



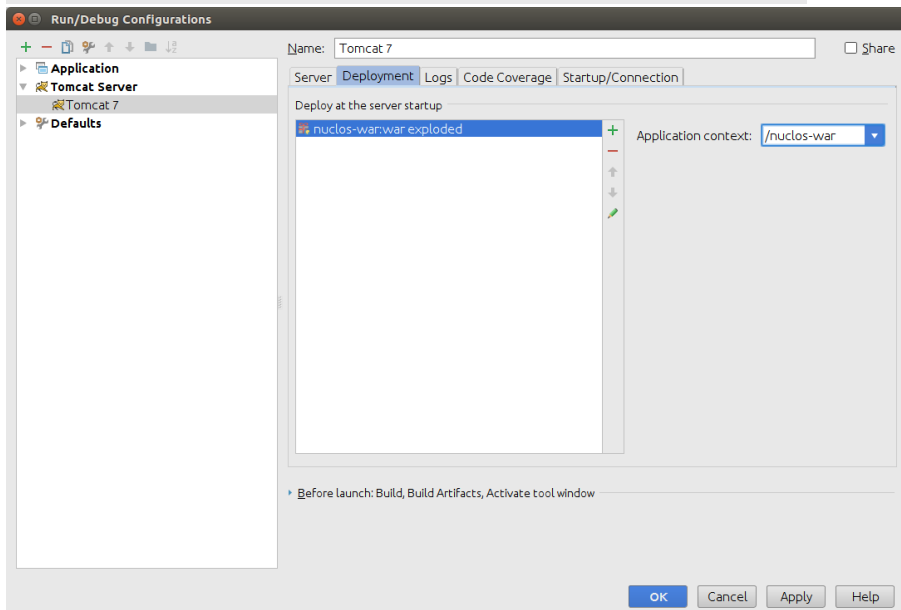
Konfigurieren Sie diesen:

Application Server	das Tomcat-Verzeichnis von <b>Projekt-Installation</b> (z. B.: [Pfad zum Projekt]/nuclos/tomcat/apache-tomcat-7.0.72)
VM Options	-Dfunctionblock.dev=true -mx1024m
Deployment start up	Add -> Artifact... -> nuclos-war-exploded (oder falls ein extension-war existiert, natürlich extension-war-exploded)
Application context	/nuclos-war



Unter Umständen müssen noch die Ausführungsrechte der Tomcat-Start-Skripte gesetzt werden, z.B.:

chmod u+x apache-tomcat-..../\*.sh

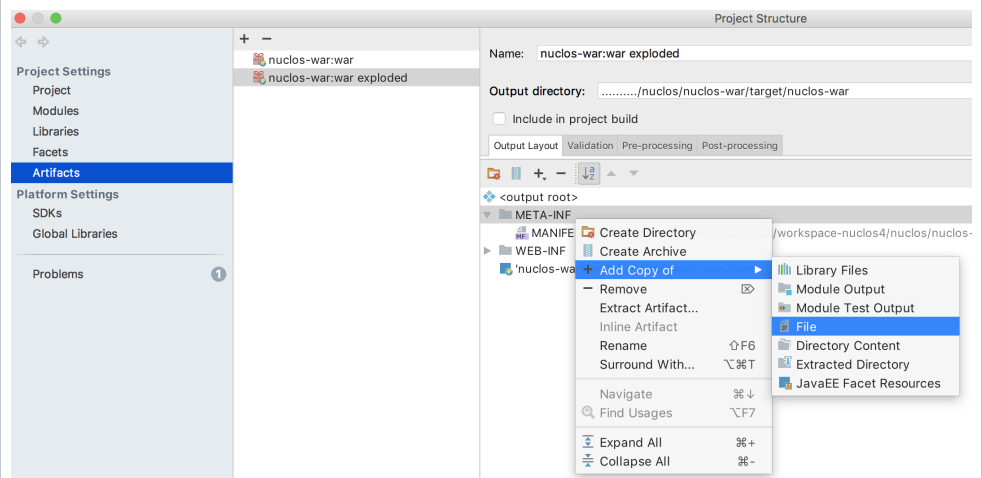


In manchen Fällen wird context.xml beim Deployment nicht in das target-Verzeichnis übernommen.

In diesem Fall hilft es die Datei context.xml unter

File > Project Structure... > Project Settings > Artifacts > Output Layout > META-INF

hinzuzufügen.

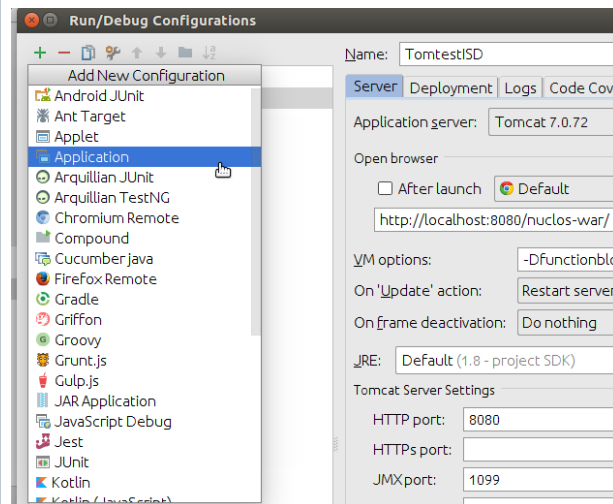


Wenn das Projekt über server-extensions verfügt und/oder auf einer Oracle-Datenbank basiert, ist ein weiterer Schritt notwendig:

Die *ojdbc6.jar* (Oracle.db) und alle *.jar*-Extensions ([Pfad zu **Projekt-Installation**]/extensions/server bzw. [Pfad zu **Projekt-Installation**]/extensions/common) müssen ins Tomcat-lib-Verzeichnis kopiert werden ([Pfad zu **Projekt-Installation**]/tomcat/apache-tomcat-7.0.72/lib).

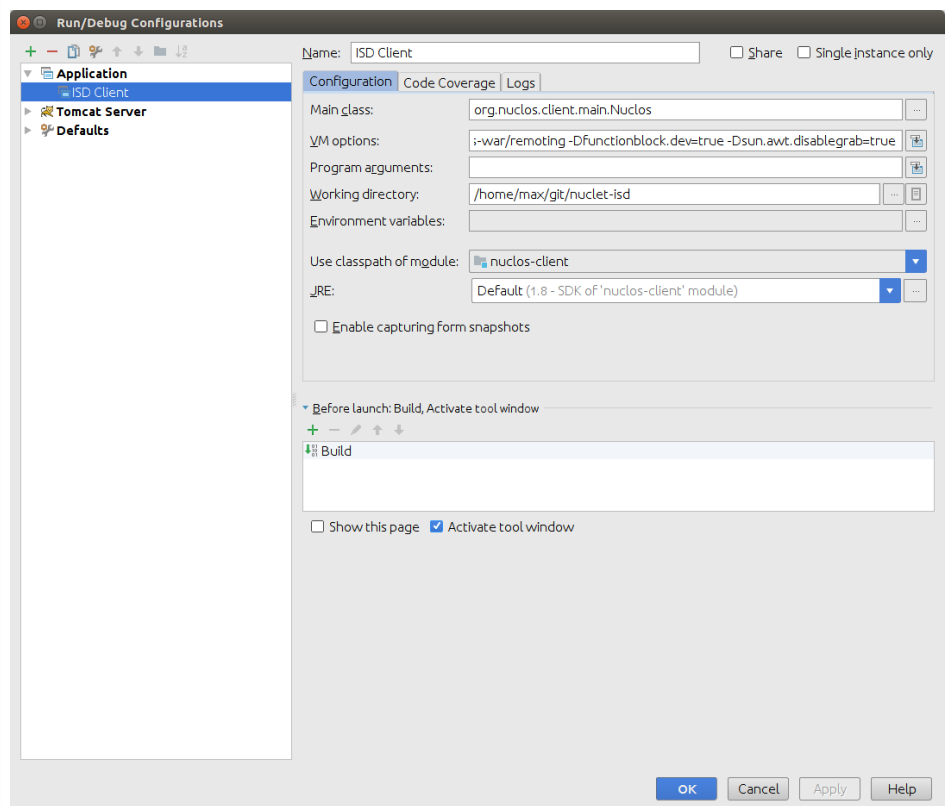
Der Server kann jetzt gestartet werden.

Abschließend muss noch der Client erstellt werden:



### Konfiguration des Clients:

<i>Main class</i>	org.nuclos.client.main.Nuclos
<i>VM options (Port muss ggf. angepasst werden)</i>	-ea -Xmx768m -Dserver=http://localhost:8080/nuclos-war - Dfunctionblock.dev=true -Dsun.awt.disablegrab=true
<i>Use classpath of module</i>	nuclos-client



## Troubleshooting

IntelliJ erkennt /[Pfad zu <b>Projekt-Installation</b> ]/tomcat/apache-tomcat-9.x nicht als Tomcat-Home an.	Die entsprechende Tomcat-Version muss manuell installiert werden. Hier eine Beispielanleitung: <a href="https://medium.com/@dolthead/upgrade-intellij-idea-to-tomcat-9-0-8-d2a560606231">https://medium.com/@dolthead/upgrade-intellij-idea-to-tomcat-9-0-8-d2a560606231</a>
Das Projekt wird gebaut, aber der Server startet nicht mit der Meldung:  "SEVERE: Unable to process Jar entry [module-info.class] from Jar [jar:file:/sahibinden/repo/services/sahibinden-web/target/sahibinden-web/WEB-INF/lib/classgraph-4.6.7.jar!/] for annotations org.apache.tomcat.util.bcel.classfile.ClassFormatException: Invalid byte tag in constant pool: 19"	Es handelt sich um einen Bug mit der Kombination aus Tomcat - und Java - Version ( <a href="https://github.com/classgraph/classgraph/issues/291">https://github.com/classgraph/classgraph/issues/291</a> ). Die neueste JDK 8 benötigt z.B. Tomcat 9.
localhost:8080 is already in use.	Es läuft bereits ein Programm auf dem Port 8080. Dieses passiert z.B. in Ubuntu, wenn man Tomcat aus den Paketquellen installiert und nicht manuell abschaltet.
Weitere Gründe für das Scheitern des Server-Starts	Das in der <b>Projekt-Installation</b> angegebene Schema muss in der Datenbank existieren (es kann aber leer sein).

## Mögliche Probleme und Behebung

Serverstart: nuclos-conf.xml nicht gefunden

Projekt - open Module Settings



The screenshot shows the IntelliJ IDEA IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The main workspace is divided into several panes:

- Project Structure:** Shows the project hierarchy. The 'Artifacts' tab is active, displaying a tree view of artifacts. A red circle highlights the 'META-INF' folder under the 'bmw-fdm-war' artifact. The 'MANIFEST.MF' file is also visible.
- Platform Settings:** Shows settings for the project, including the output directory and platform-specific configurations.
- Console:** Displays the output of the application. The output shows the start of a Spring Boot application, including the main method call, classpath logging, and the start of the application server.

manuell hinzufügen