

Regeln als Maven Projekt in einer IDE einbinden

- [Definition](#)
- [Voraussetzungen:](#)
- [Stand:](#)

Definition

Ab Nuclos v4.30 können Geschäftsregeln einfach als Maven Projekt in einer IDE eingebunden werden. Nuclos generiert hier für automatisch eine pom.xml unter <NUCLOS-HOME>/data/codegenerator

Voraussetzungen:

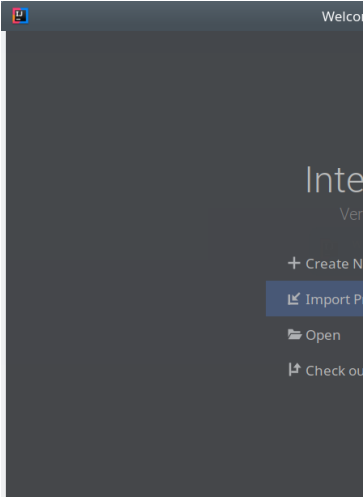
1. Der Server wird lokal ausgeführt und das **codegenerator** Verzeichnis befindet sich entsprechend auf dem Entwicklungsrechner, oder es wurden die [Parameter](#) CODEGENERATOR_POM_LIBDIR, CODEGENERATOR_POM_AXIS_LIBDIR und CODEGENERATOR_POM_EXTENSION_LIBDIR konfiguriert.
2. Der Server wird im **Entwicklungsmodus** betrieben (Siehe [Installation fortgeschritten](#))
3. Eine IDE Ihrer Wahl mit Maven Unterstützung. Hier im Beispiel wird die [IntelliJ IDEA Community](#) Edition verwendet.

Berechtigung

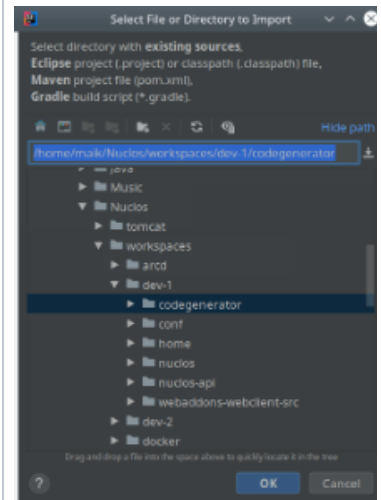
Es ist auch ratsam Nuclos mit normalen Benutzerrechten zu installieren und zu betreiben, nicht als Dienst. Ansonsten könnte es Probleme mit den Berechtigungen und den Zugriff auf die Dateien geben. Auch kann es unter Umständen helfen das codegenerator Verzeichnis zu löschen und beim nächsten Start vom Server neu erstellen zu lassen.

Stand:

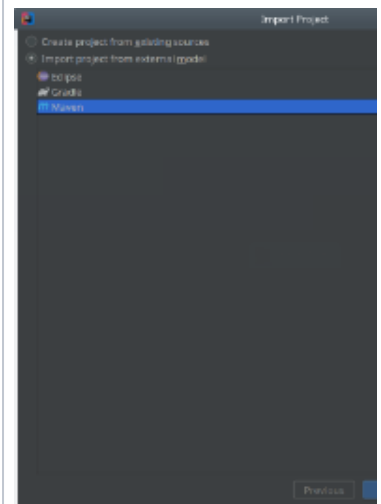
Die aktuelle Anleitung wurde getestet mit Nuclos 4.2024.8 und IntelliJ IDEA 2024.1.1 (Community Edition)

Neues Projekt anlegen.	

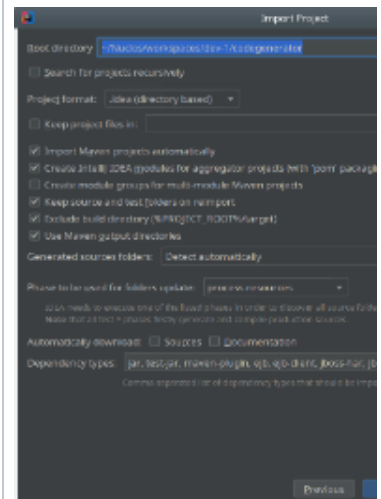
Wählen Sie das **codegenerator** Verzeichnis aus. Normalerweise befindet sich in <NUCLOS-HOME>/data Ihrer Installation.



IntelliJ erkennt dieses Verzeichnis automatisch als Maven Projekt an.

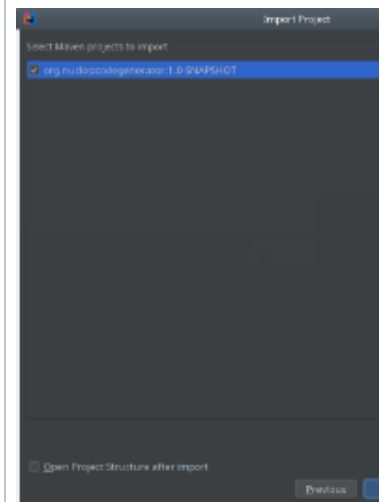


*"Import Maven projects automatically" wurde hier zusätzlich ausgewählt.



Gefundenes **codegenerator** Projekt bestätigen.

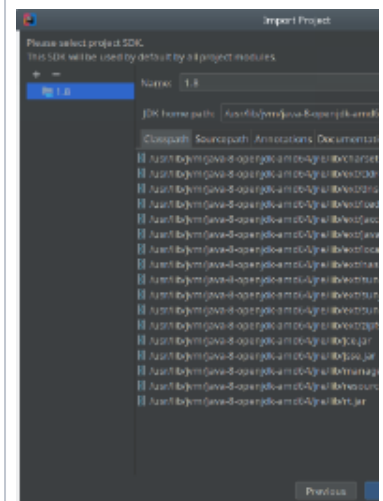
⚠ **Wichtig**, der Server sollte im Hintergrund ausgeführt werden. Nur dann ist sichergestellt, dass auch alle verwendeten Bibliotheken gefunden werden.



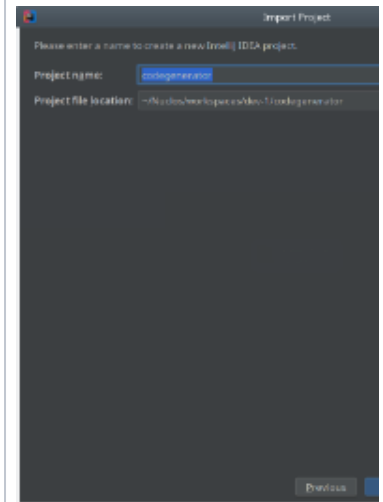
Beim ersten Start wird vermutlich noch ein SDK fehlen. Derzeit benötigt Nuclos ein Java JDK der Version 8. Über den Plus-Button können Sie eines hinzufügen. Am besten verwenden Sie das Gleiche womit auch der Server betrieben wird.



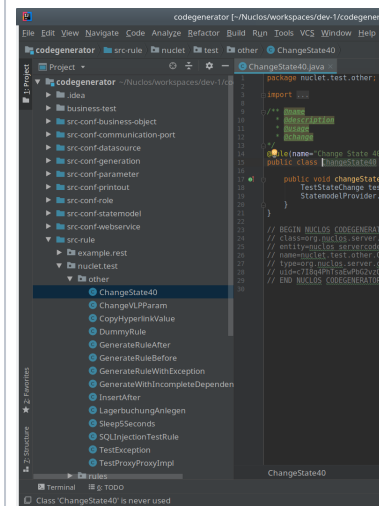
So sollte es dann mit ausgewählten SDK aussehen.



Der Name kann so übernommen werden. Schließen Sie den Wizard mit **Finish** ab.



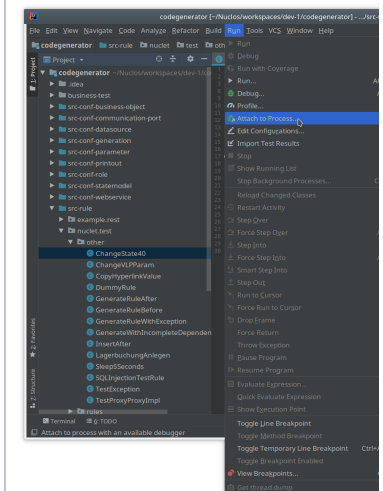
Im Projekt Fenster können Sie nun zu Ihren bereits geladenen Regeln navigieren. Durch den Import und das Erkennen als Maven Projekt sind bereits alle Abhängigkeiten korrekt gesetzt.



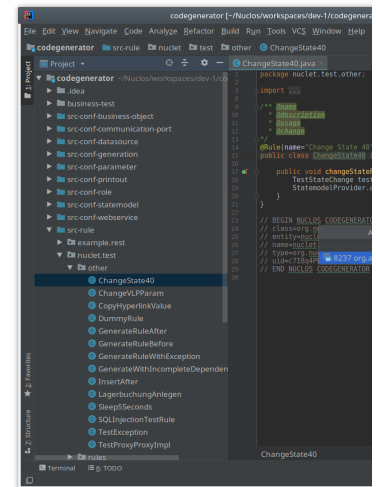
Über das Menü

Run Attach to Process...

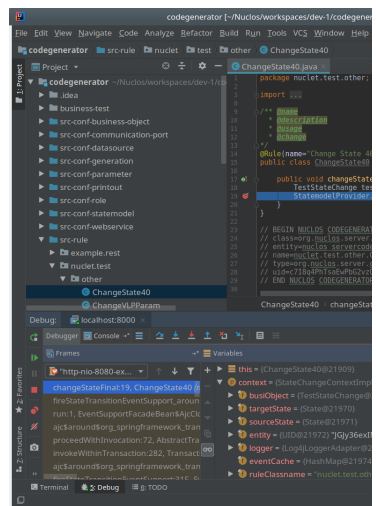
können Sie sich mit dem Server verbinden. Dies ermöglicht einerseits das Debuggen Ihrer Regeln über Breakpoints, als auch Änderungen direkt in die laufenden VM zu übernehmen.



Eine Auswahl mit den laufenden Java Prozessen wird angezeigt. Wählen Sie Ihren Nuclos Server aus. Standardmäßig lautet der Debug Port 8000.



Nun können Sie mit Hilfe von Breakpoints komfortabel Ihre Regeln debuggen.



Über das Menü

Build Build Project...

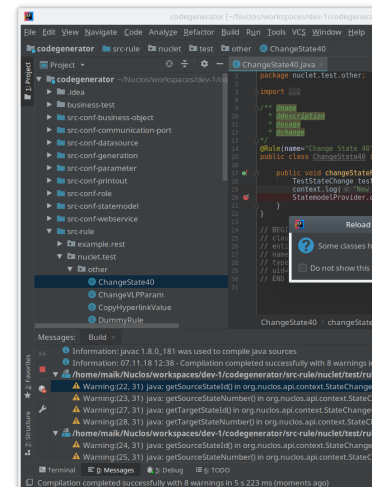
wird eine Neukompilierung der Regeln angestoßen.

Nach Bestätigen der Rückfrage (siehe Screenshot) werden die geänderten Klassen ohne Umwege sofort an die laufende VM übermittelt. Dies geschieht über den Debug Port.

Die Kompilierung schlägt mit folgender Meldung fehl?

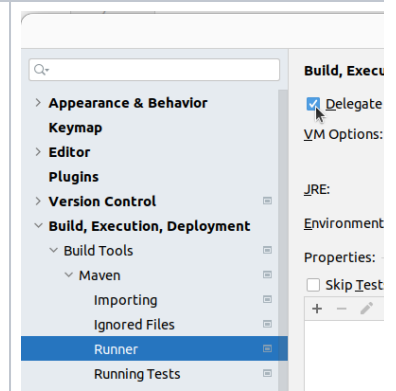
Error: java: JDK isn't specified for module 'codegenerator'

Dann müssen Sie das SDK neu setzen. Lesen Sie [hier](#) wie Sie am besten vorgehen.



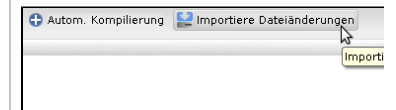
Um Konflikte mit dem Nuclos Compiler zu vermeiden (automatisch erzeugte Klassen werden nicht gefunden) sollte die Option "Delegate IDE build/run actions to Maven" aktiviert werden.

Eine andere Lösung ist, die *Automatische Kompilierung* in Nuclos zu deaktivieren, dies erfordert aber ein Neuladen der Regeln vom Dateisystem (siehe nächster Schritt).



Alternativ (ab Nuclos 4.53) können Sie über den Button **Importiere Dateiänderungen** (Ergebnisliste der Regeln) Änderungen an Dateien direkt eingelesen werden.

Dies ist z.B. hilfreich, wenn der Server in einer VM (oder Docker Container) betrieben wird und die automatische Überwachung von Dateiänderungen nicht greift, da die Dateien möglicherweise auf einem "Share" Laufwerk abgelegt werden (im Falle von Docker z.B. auf dem volume `/opt/nuclos/home/data/codegenerator`)



```
2018-11-07 12:39:29,621 DEBUG [SQLLogger] - SELECT t.INTID, t.DATCREATED, t.STRCREATED, t.DATCHANGED, t.STRCHANGED, t.INTVERSION, t.STRname, t.STRNUCLOSSYSTEMID
=40004318]
2018-11-07 12:39:29,626 DEBUG [SQLLogger] - SELECT t.INTID, t.DATCREATED, t.STRCREATED, t.DATCHANGED, t.STRCHANGED, t.INTVERSION, t.STRname, t.STRNUCLOSSYSTEMID
=40004318]
2018-11-07 12:39:43,142 DEBUG [SQLLogger] - SELECT t.INTID, t.DATCREATED, t.STRCREATED, t.DATCHANGED, t.STRCHANGED, t.INTVERSION, t.STRname, t.STRNUCLOSSYSTEMID
=40004318]
2018-11-07 12:39:43,158 DEBUG [SQLLogger] - SELECT t.INTID, t.DATCREATED, t.STRCREATED, t.DATCHANGED, t.STRCHANGED, t.INTVERSION, t.STRname, t.STRNUCLOSSYSTEMID
=40004318]
2018-11-07 12:39:43,159 DEBUG [SQLLogger] - SELECT t.INTID, t.DATCREATED, t.STRCREATED, t.DATCHANGED, t.STRCHANGED, t.INTVERSION, t.STRname, t.STRNUCLOSSYSTEMID
=40004318]
2018-11-07 12:39:43,160 DEBUG [SQLLogger] - SELECT t.INTID, t.DATCREATED, t.STRCREATED, t.DATCHANGED, t.STRCHANGED, t.INTVERSION, t.STRname, t.STRNUCLOSSYSTEMID
=40004318]
2018-11-07 12:39:43,161 DEBUG [SQLLogger] - SELECT NEXTVAL('IDFACTORY')
=1]
2018-11-07 12:39:43,161 DEBUG [SQLLogger] - INSERT INTO T_UD_G0 STATEHISTORY (INTID, DATCREATED, STRCREATED, DATCHANGED, STRCHANGED, INTVERSION, STRUID_T_MD_S
=40004326, 2018-11-07 12:39:43,161, nuclos, 2018-11-07 12:39:43,161, nuclos, 1, P2mVnmKlnaTlRhpfeF, 40004318]
2018-11-07 12:39:43,161 DEBUG [SQLLogger] - SELECT t.INTID, t.DATCREATED, t.STRCREATED, t.DATCHANGED, t.STRCHANGED, t.INTVERSION, t.STRUID_T_MD_STATE, t.INTIC
=40004326]
2018-11-07 12:39:43,162 DEBUG [SQLLogger] - UPDATE FAC0 TESTSTATECHANGE SET DATCHANGED = ?, STRCHANGED = ?, INTVERSION = ?, STRname = ?, STRNUCLOSSYSTEMID = ?
=2018-11-07 12:39:43,162, nuclos, 7, 9, TS1811#000000, false, P2mVnmKlnaTlRhpfeF, 40004318]
2018-11-07 12:39:43,163 DEBUG [SQLLogger] - SELECT t.INTID, t.DATCREATED, t.STRCREATED, t.DATCHANGED, t.STRCHANGED, t.INTVERSION, t.STRname, t.STRNUCLOSSYSTEMID
=40004318]
2018-11-07 12:39:43,166 INFO [nuclet.test.other.ChangeState40] - New log message!
```

Ein kleiner Test für eine zusätzliche Ausg

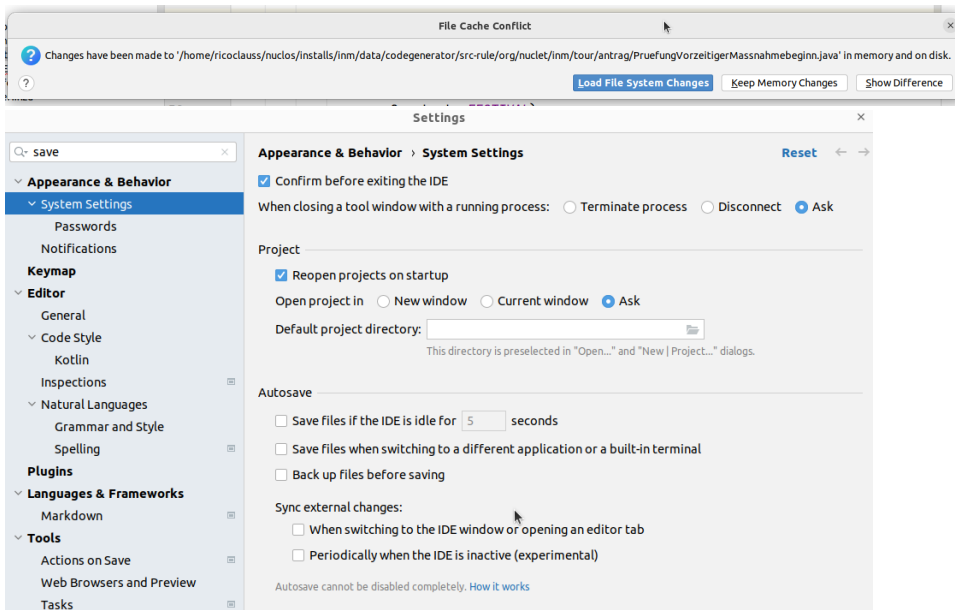
Die Möglichkeit zu debuggen und die Code Completion einer IDE wird Ihre Produktivität deutlich steigern. Probieren Sie es aus!

Viel Spaß 😊

Probleme mit der IDE?

Dat eik onfl ikte:	
Bei m glei chz eiti gen Arb eite n in Nuc los und der IDE kan n es sei	

n,
das
s
Nuc
los
ein
e
neu
ere
Ver
sio
n
ein
er
Re
gel
erz
eug
t,
die
die
in
der
IDE
ge
ma
cht
Änd
eru
nge
n
rüc
kgä
ngi
g
ma
che
n
wür
de.
In
de
m
Fall
mu
ss
der
Dat
eik
onfl
ikt
zug
uns
ten
der
in
inte
lliJ
bea
rbei
tete
n
vari
ant
e
auf
gel
öst
wer
den
("K
eep
Me
mor
y
Ch
ang
es"
bzw

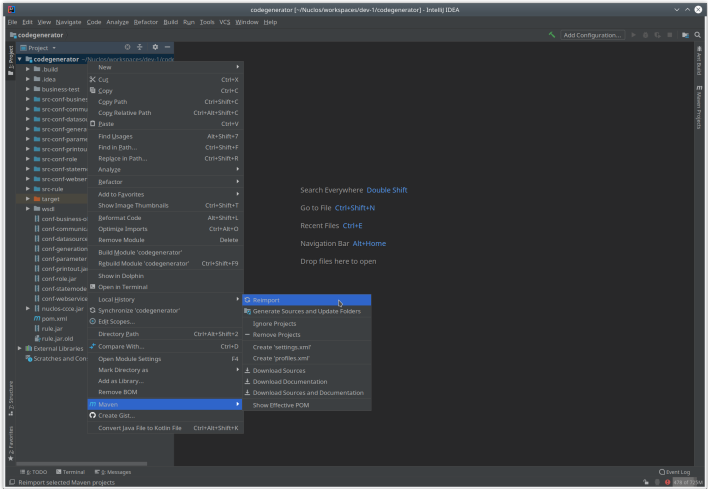


.
Alt+
K).

Um
das
Auft
rete
n
solc
her
Kon
flikt
e
zu
red
uzi
ere
n,
kan
n
die
Aut
osa
ve-
Fun
ktio
n
von
inte
lliJ
weit
geh
end
dea
ktivi
ert
wer
den

.
Änd
eru
nge
n
kön
nen
dan
n
we
nn
nöti
g
per
Str
g+
S
auf
das
Dat
eisy
ste
m
ges
chri
ebe
n
wer
den.

Nicht immer erkennt eine IDE schnell genug das sich Source code geändert hat, in diesem Fall sollte ein Reimport durch Maven Abhilfe schaffen.



Ein typisches Beispiel wäre ein Wechsel des Datenbankschemas mit einem ganz anderen Nucleus, oder nach dem Einspielen eines Datenbanks-Dumps.

Error: java: JDK isn't specified for module 'codegenerator'

Sie erhalten die obige Fehlermeldung beim kompilieren der Regeln?

Dann tragen Sie das JDK erneut ein. Dieser Fehler kann bei einem ganz neuen Projekt oder auch wenn Sie das JDK ändern auftreten.

Gehen Sie wie folgt vor:

Im Menü

File Project Structure...

aufrufen

Das **Project SDK** einmal leeren mit dem Eintrag **<No SDK>** Apply

Und wieder wie hier im Beispiel 1.8 auswählen OK

Im Anschluss sollten Ihre Regeln wieder kompilieren.