

WildFly: Installation und Konfiguration des Application Servers

Vorbemerkung

Wir geben in diesem Dokument Dateinamen im Windows-Format an. Für Linux ersetzen Sie bitte das Trennzeichen \ durch /.

Kommandoskripte sind für Windows i. d. R. `.bat`-Dateien (teilweise sind auch `.cmd` oder `.ps1` vorhanden). Unter Linux nehmen Sie stattdessen `.sh`-Dateien.

Installation und Konfiguration der Server Runtime für das Seminar

TIP

Bei Seminaren, die WildFly benötigen, werden die hier beschriebenen Schritte (Download, Installation, Anpassen der Konfiguration) durch den Aufruf von `mvn` im Verzeichnis `labs` bereits durchgeführt. Der Server steht Ihnen im Verzeichnis `labs\tools\target\wildfly-24.0.0.Final` zur Verfügung. Das Unterverzeichnis `standalone-seminar` enthält die für das Seminar angepasste Serverkonfiguration.

Download und Installation

WildFly kann von <http://wildfly.org/downloads/> heruntergeladen werden. Im Seminar wird die sog. *Jakarta EE Full & Web Distribution* in der Version `24.0.0.Final` genutzt.

Das heruntergeladene File `wildfly-24.0.0.Final.zip` (bzw. `.tgz`) kann an beliebiger Stelle entpackt werden. Dabei entsteht ein neues Verzeichnis namens `wildfly-24.0.0.Final`, das im Rest dieses Dokumentes mit `<wfly_home>` bezeichnet wird.

Erzeugung einer an das Seminar angepassten Konfiguration

Wir nutzen den Server im Seminar im sog. Standalone-Modus. Um die Grundkonfiguration im Verzeichnis `<wfly_home>\standalone` unangetastet zu lassen, legen wir nach dem Download eine Kopie dieses Verzeichnisses mit dem Namen `<wfly_home>\standalone-seminar` an. Darin konfigurieren wir zur Nutzung im Seminar:

- Das Scan-Intervall des Deployment-Scanners wird auf 1 s gesetzt.
- Es wird eine Datasource für die Nutzung einer H2-Datenbank eingerichtet:

- JNDI Name: `java:/jdbc/seminar`
- Connect URL: `jdbc:h2:~/h2/seminar;AUTO_SERVER=TRUE`
- User / Password: `seminar / seminar`.
- Es werden zwei Messaging Destinations `jms/queue/seminar` und `jms/topic/seminar` angelegt (jeweils in `java:/` und `java:jboss/exported/`)
- Die Nutzung von System Properties in Deskriptoren wird erlaubt.
- Die MicroProfile Extensions und Subsysteme für Fault Tolerance, Health, Metrics und OpenAPI werden ergänzt.
- Es werden einige Test- und Demo-User angelegt:

Typ	User	Password	Roles
Administrations-User	<code>admin</code>	<code>admin_123</code>	
Anwendungs-User	<code>anonymous</code>	<code>anonymous_123</code>	<code>guest</code>
Anwendungs-User	<code>hugo</code>	<code>hugo_123</code>	<code>guest demoRole</code>
Anwendungs-User	<code>otto</code>	<code>otto_123</code>	<code>guest</code>

Bei Interesse finden Sie die zur Konfiguration von `standalone-seminar` genutzten CLI-Befehle in der Datei `tools\setup\wildfly\standalone-seminar.cli` sowie die weiteren Konfigurationsdateien in `tools\setup\wildfly\configuration`.

Start und Stopp des Servers

Der Server wird durch den folgenden Befehl gestartet:

```
<wfly_home>\bin\standalone.bat -Djboss.server.base.dir=<wfly_home>\standalone-seminar --server-config=standalone-full.xml
```

Zum Stopp des Servers kann im Server-Fenster `Strg-C` genutzt werden.

TIP

Im Seminar (und auch sonst zur Entwicklung von Software) ist es empfehlenswert, den Server nicht wie gezeigt separat zu starten, sondern ihn in die genutzte IDE zu integrieren und von dort zu kontrollieren.

Integration des Servers in die IDE

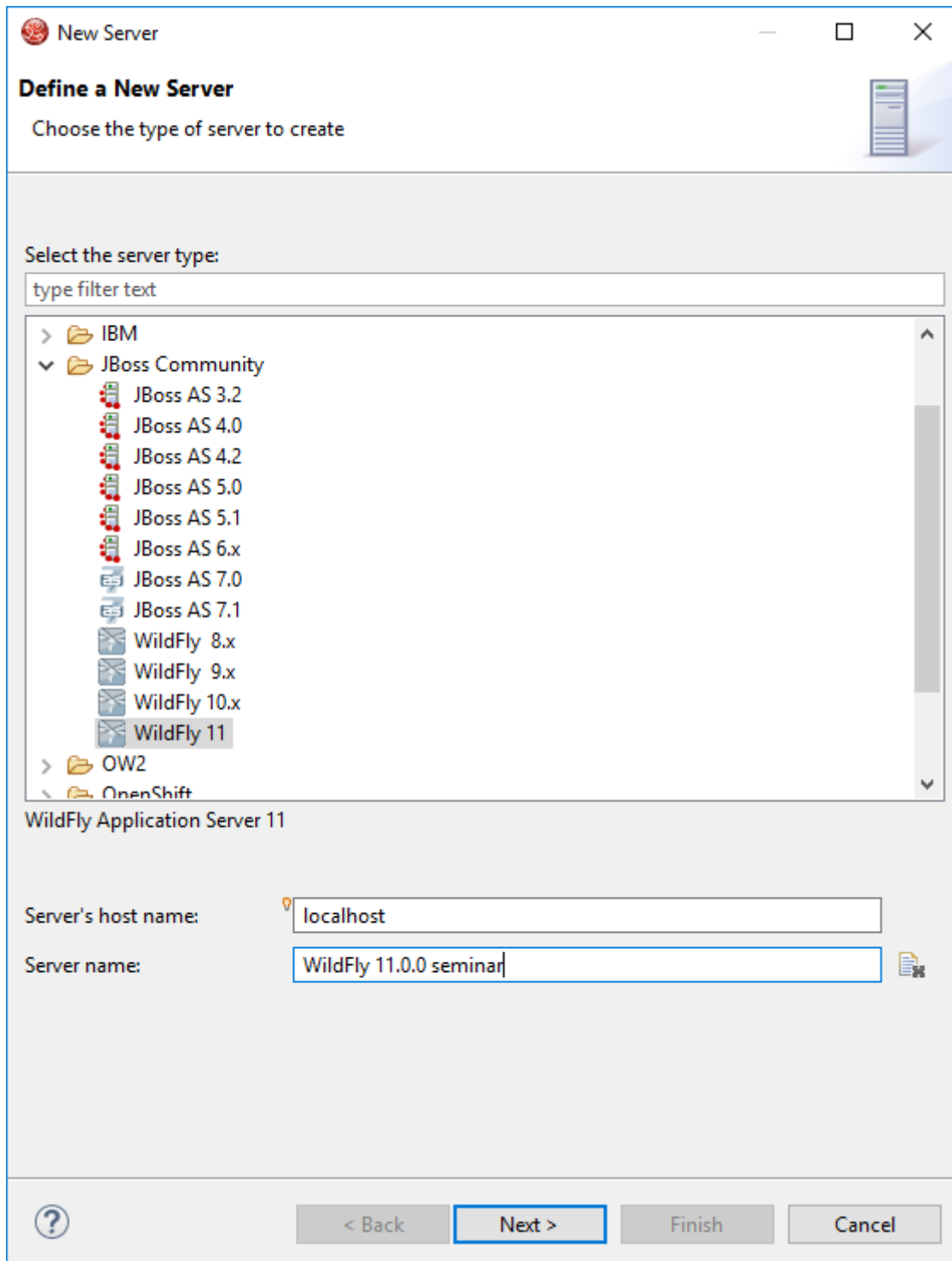
Eclipse

Definition der Server Runtime

- Fügen Sie die View `Servers` Ihrer genutzten Perspektive hinzu. Dazu nutzen Sie den Menüpunkte `Window` → `Preferences` → `Show View` → `Other...` und wählen die View namens `Servers` aus.

- Klicken Sie mit der rechten Maustaste in den freien Bereich der View **Servers** , wählen aus dem Kontextmenü **New** → **Server**, klicken aus dem Ordner **JBoss Community** den Eintrag mit der passenden Version und nennen den neuen Eintrag **WildFly 24.0.0.Final seminar**. Sollte es noch keinen Eintrag für die Version **24.0.0.Final** geben, nehmen Sie den Eintrag mit der höchsten Versionsnummer.

(Die folgenden Screenshots sind auf Basis einer älteren Version entstanden)



- Nach zweimaligem Klick auf **Next** konfigurieren Sie die *JBoss Runtime* mit den folgenden Parametern:
 Name: **WildFly 24.0.0.Final seminar**
 Home Directory: **<wfly_home>**

Runtime JRE: JDK der Version 8 oder 11
Server Base Directory: `standalone-seminar`
Configuration File: `standalone-full.xml`

New Server

JBoss Runtime

WildFly Application Server 11

A JBoss Server runtime references a JBoss installation directory.
It can be used to set up classpaths for projects which depend on this runtime,
as well as by a "server" which will be able to start and stop instances of JBoss.

Name
WildFly 11.0 Runtime

Home Directory [Download and install runtime...](#)
C:\GEDOPLAN\server\wildfly\wildfly-11.0.0.Final [Browse...](#)

Runtime JRE

☒ Execution Environment: JavaSE-1.8 [Environments...](#)

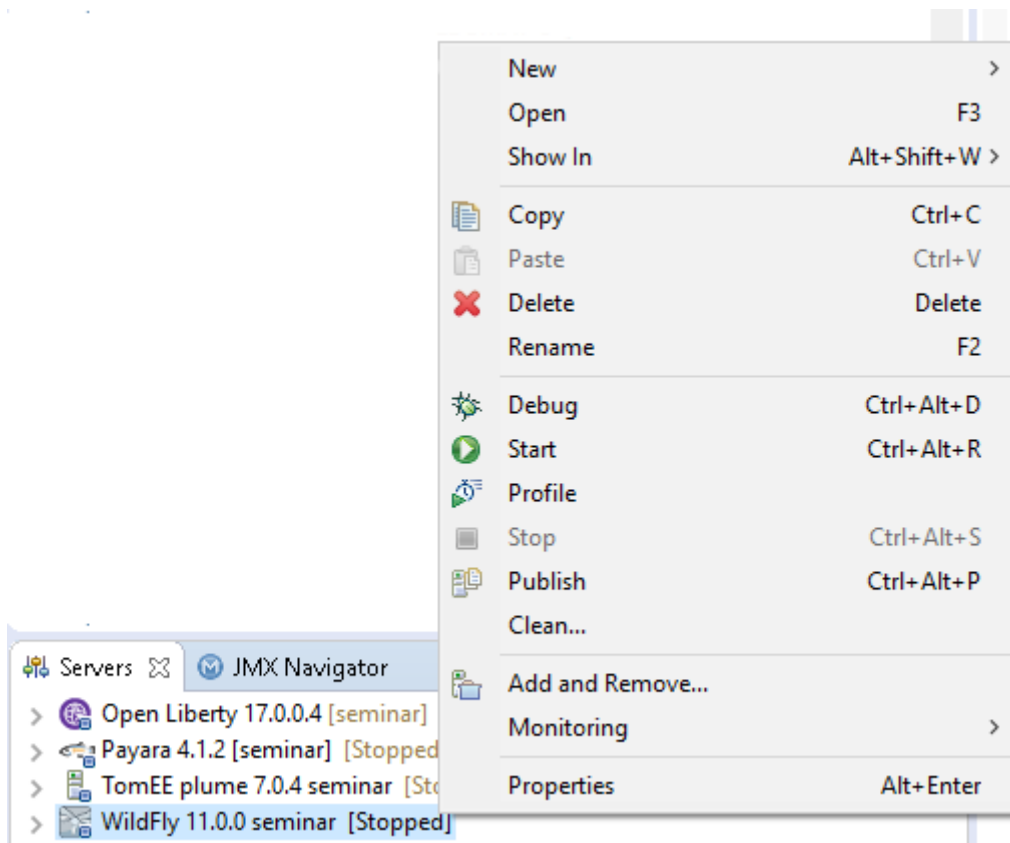
☐ Alternate JRE: jdk1.8.0_144 [Installed JREs...](#)

Server base directory: standalone-seminar [Browse...](#)

Configuration file: standalone-full.xml [Browse...](#)

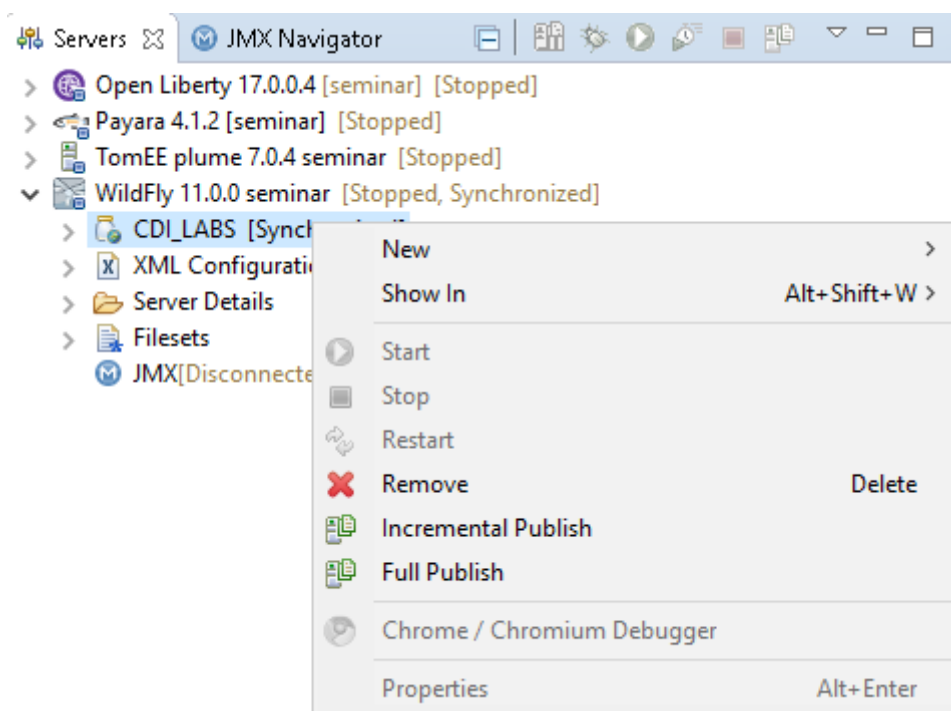
[?<](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

- Nach Abschluss des Konfigurationsdialogs mit **Finish** erscheint ein entsprechender Eintrag in der View **Servers**. Nach einem Rechtsklick darauf kann der Server gestartet (und später auch wieder gestoppt) werden.



Deployment von Anwendungen

Anwendungen können per Drag-and-Drop in den Server gebracht werden. Dazu ziehen Sie das gewünschte Projekt aus der View **Package Explorer** (oder **Projekt Explorer**) auf den Servereintrag in der View **Servers**. Die Anwendung erscheint dann dort eingerückt unterhalb des Servereintrags und kann mit einem Rechtsklick erneut deployt (**Full Publish**) oder wieder entfernt werden (**Remove**).



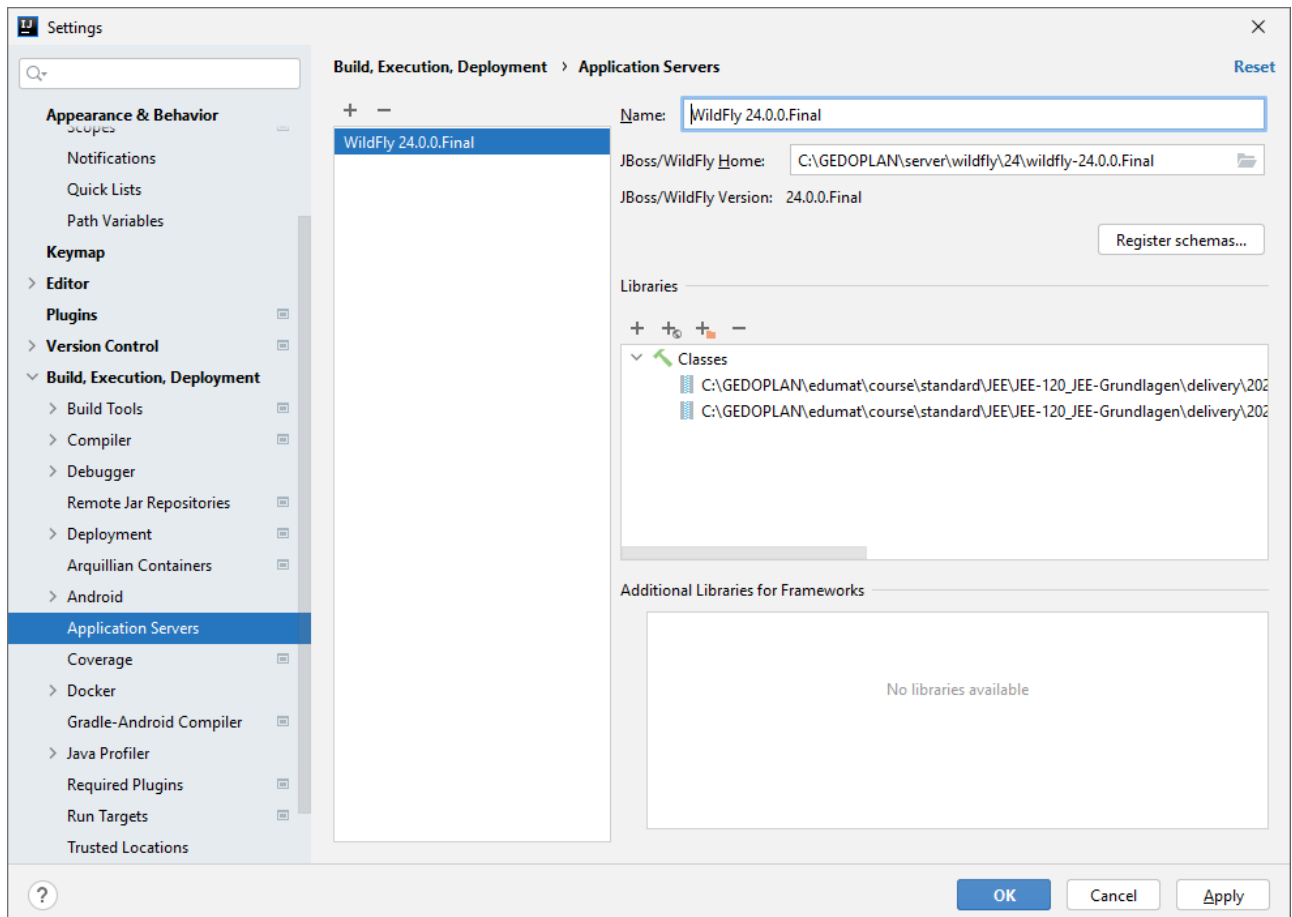
IntelliJ IDEA

NOTE

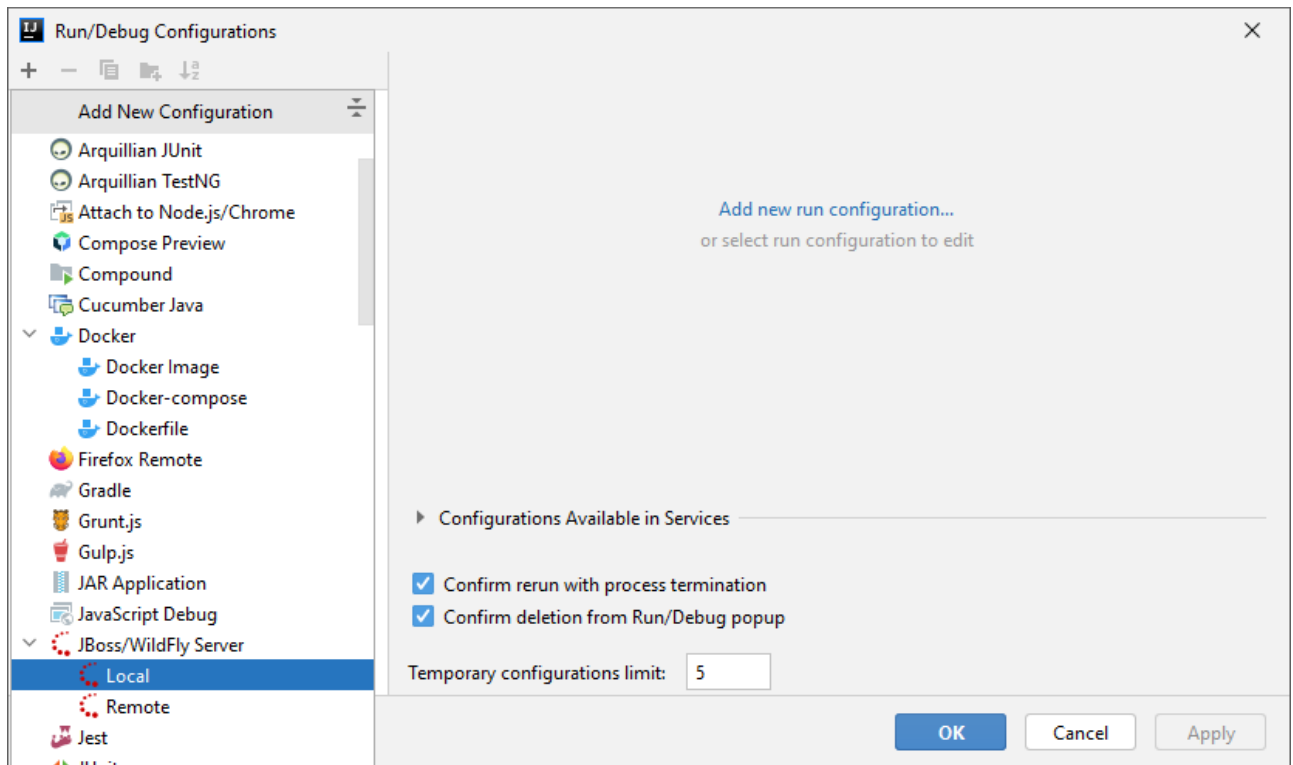
Die kostenlose *Community Edition* reicht für Enterprise-Anwendungen leider nicht aus. Sie benötigen die kostenpflichtige *Ultimate Edition*.

Definition der Server Runtime

- Fügen Sie in **File > Settings (Ctrl-Alt-S)** im Punkt **Build, Execution, Deployment > Application Servers** einen **JBoss/WildFly Server** hinzu. Sie können ihn im Dialog passend benennen und müssen als **JBoss/WildFly Home** das Verzeichnis **<wfly_home>** eintragen:



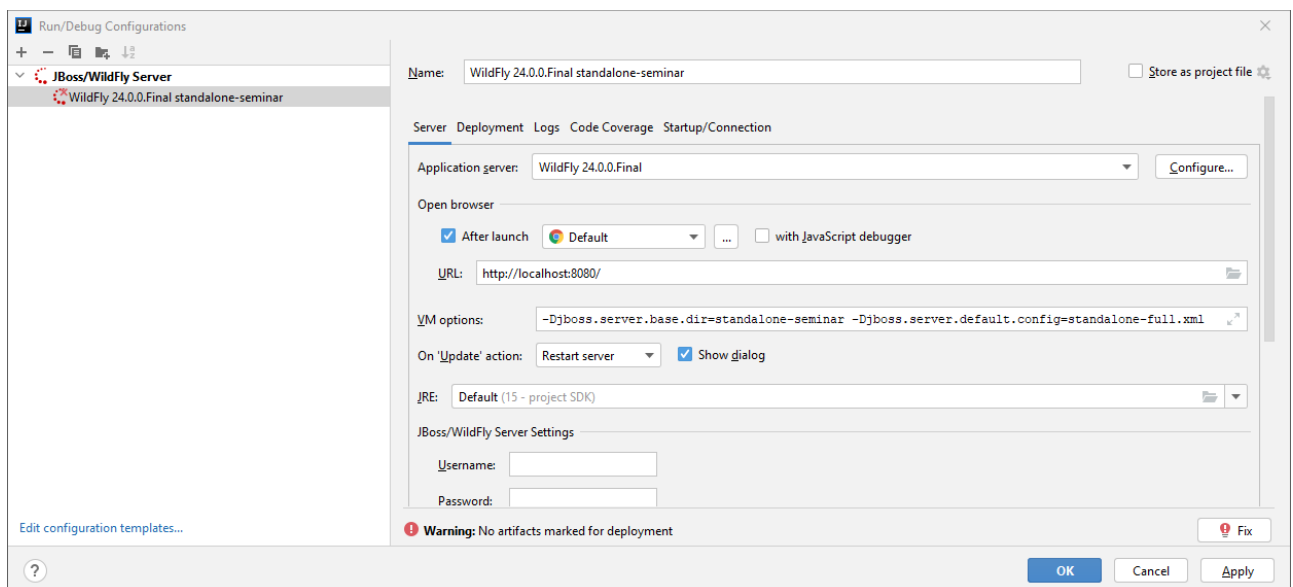
- Nun erzeugen Sie in **Run > Edit Configurations** eine Start-Konfiguration für unseren lokalen Seminar-Server:



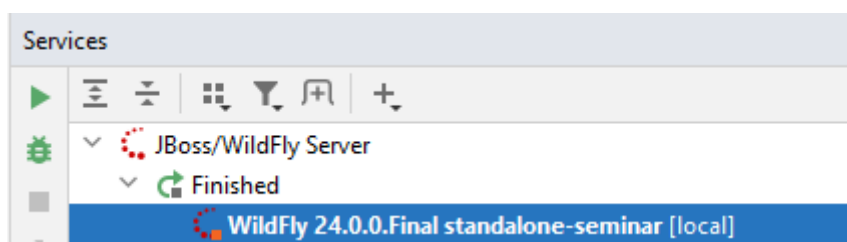
Auch hier können Sie einen passenden Namen eintragen. Um unsere an das Seminar angepasste Konfiguration des Servers zu nutzen, tragen Sie im Eingabefeld **VM Options** den folgenden Wert ein:

`-Djboss.server.base.dir=standalone-seminar`
`-full.xml`

`-Djboss.server.default.config=standalone`

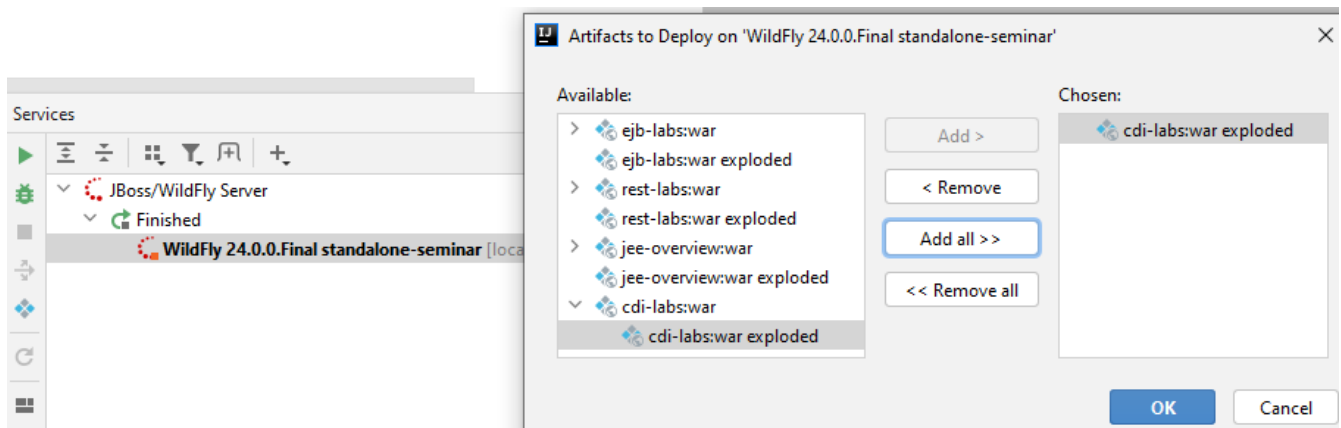


- Der Server erscheint in der View **Services** und kann dort mit den Icons links gestartet und gestoppt werden:



Deployment von Anwendungen

Durch Rechtsklick auf den Server können Deployment-**Artifacts** hinzugefügt werden:



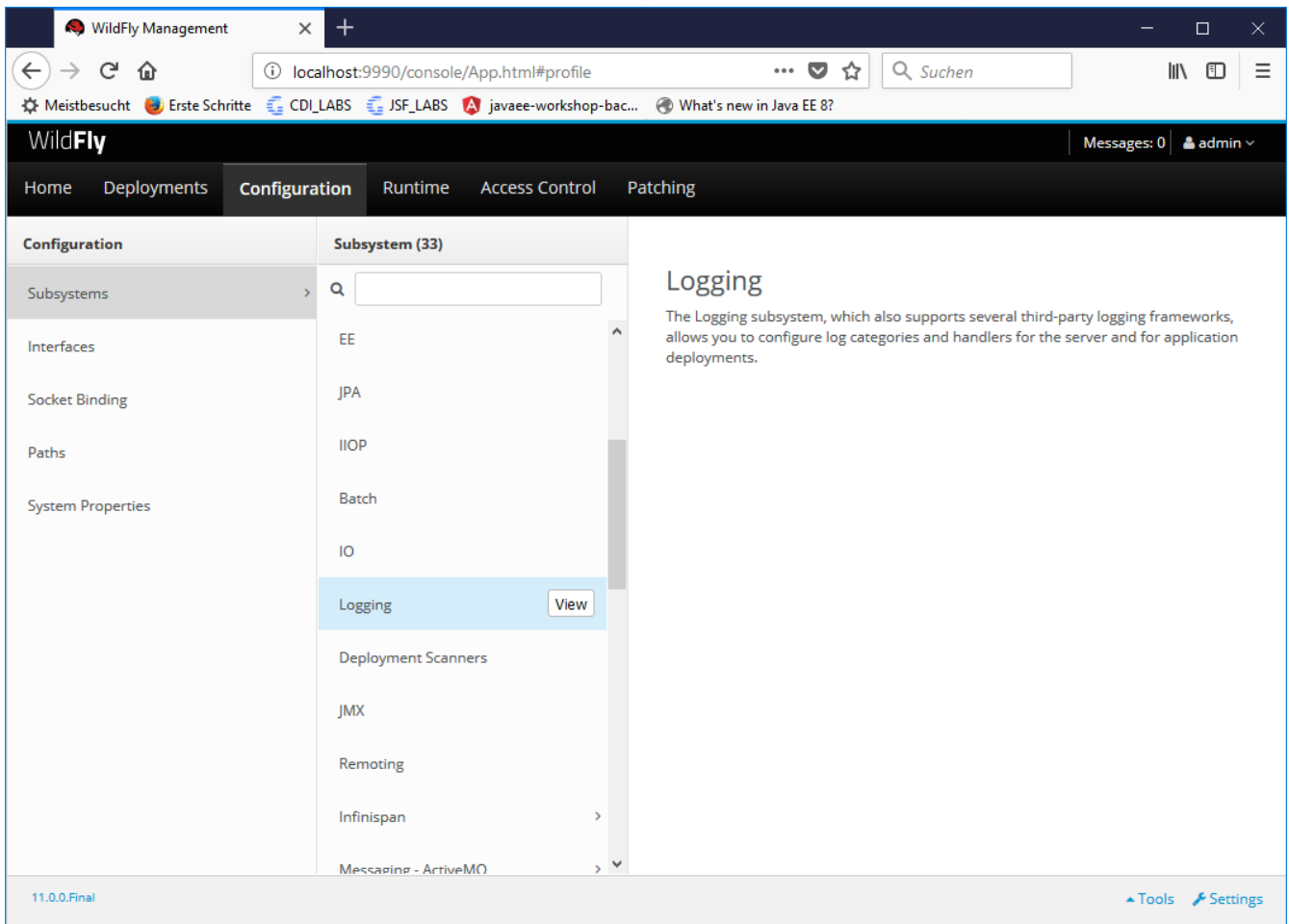
Diese können dann wiederum mit Rechtsklicks gestartet, redeployt, gestoppt und entfernt werden.

Konfiguration des Logging-Systems

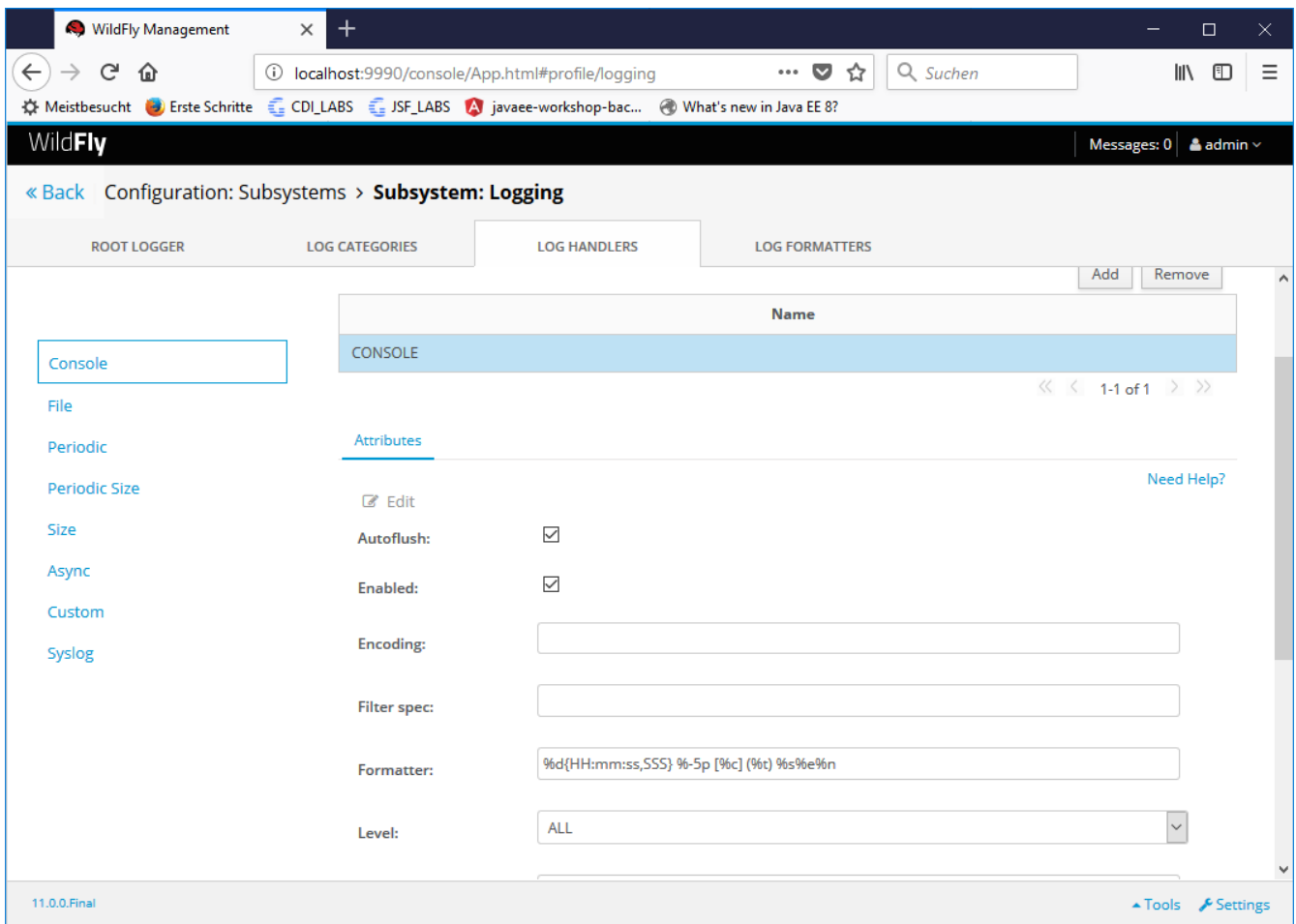
Die Protokollausgaben des Servers erscheinen in der View **Console** (und zusätzlich noch in einer Log-Datei). Im Auslieferungszustand werden nur Meldungen mit einem Schwellwert von **INFO** oder höher angezeigt.

Änderungen daran können am einfachsten mit der Web-Anwendung *WildFly Management* erfolgen. Um sie nutzen zu können, wird ein administrativer User benötigt. Bei einem für das Seminar vorkonfigurierten Server ist bereits ein User **admin** mit dem Passwort **admin_123** vorhanden. Sollte das nicht der Fall sein, öffnen Sie bitte ein Kommandofenster im Verzeichnis `<wfly_home>\bin` und starten das Kommando `add-user.bat -sc ..\standalone-seminar\configuration -u admin -p admin_123`. Damit wird der o. a. Administrations-User in die Konfigurationsdatei `<wfly_home>\standalone-seminar\configuration\mgmt-users.properties` eingetragen.

Nun können Sie (bei laufendem Server) die Web-Anwendung *WildFly Management* mit einem Web-Browser Ihrer Wahl unter <http://localhost:9990> aufrufen, sich mit dem User **admin** anmelden und den Menüpunkt **Configuration** → **Subsystems** → **Logging** auswählen.



Nach Klick auf **View**, **LOG HANDLERS** und **CONSOLE** ändern Sie das **Level** von **INFO** auf **ALL**.



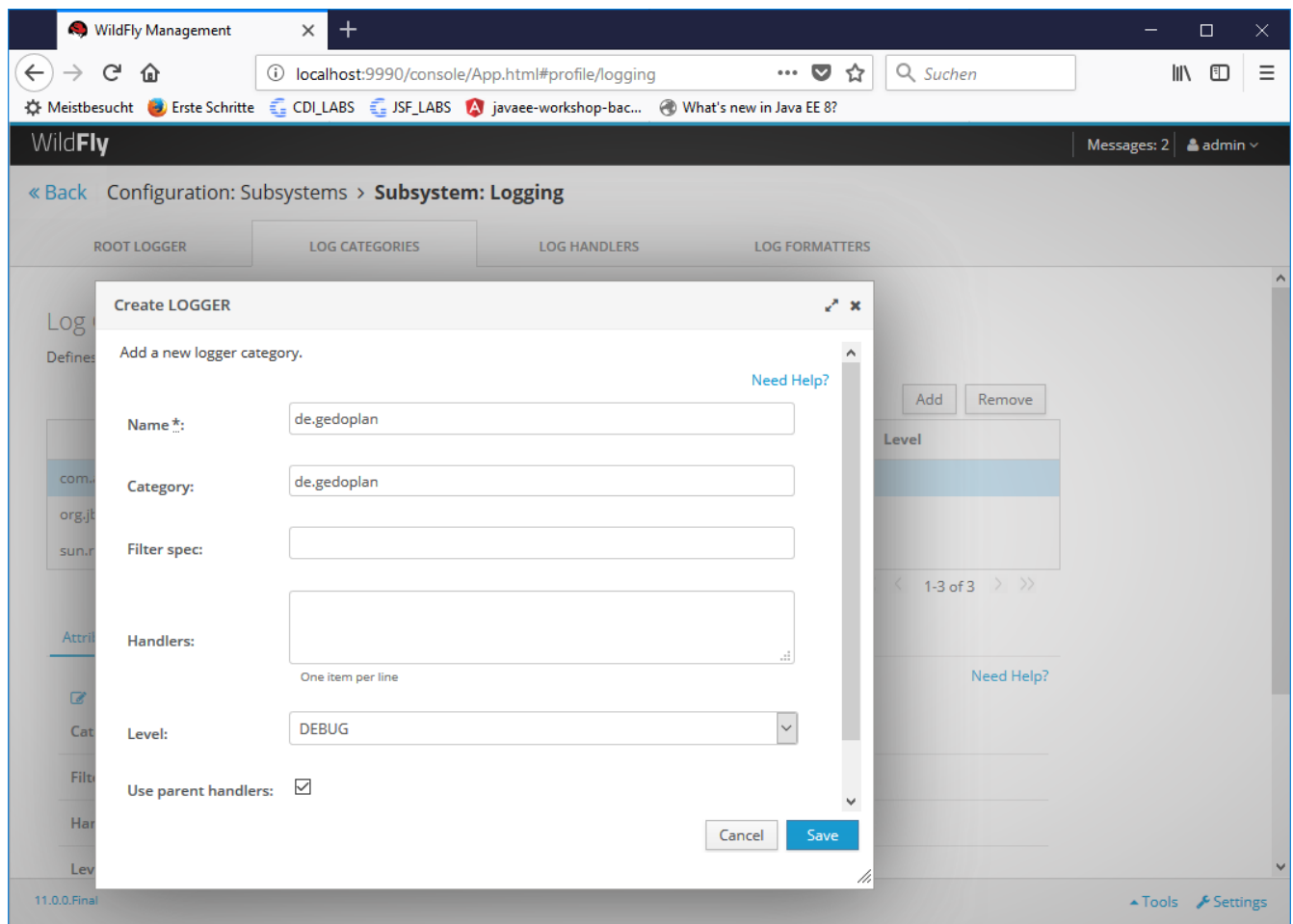
Schließlich legen Sie in **LOG CATEGORIES** mit Hilfe des Buttons **Add** einen neuen Logger mit folgenden Parametern an:

Name: **de.gedoplan**

Category: **de.gedoplan**

Level: **DEBUG**

Use parent handlers: [x]



Analog können Sie weitere Logger anlegen oder auch bestehende Einträge modifizieren. Die Änderungen werden sofort aktiv - auch ohne Neustart des Servers.