Starten der Integrationstests mit IntelliJ (und Docker)

- [Generelles Setup](#)
- [Starten der Server-Integrationstests](#)
- [Starten der Webclient-Integrationstests](#)

Generelles Setup

1. Starten des Server in IntelliJ auf localhost (falls eine anderer Host/IP verwendet wird, muss localhost im weiteren Verlauf ersetzt werden)
2. Importieren des Testnuclets (falls noch nicht geschehen). Beim Import über REST ist kein Richclient nötig.
3. Navigation im Projekt zu

```
nuclos-integrationstests/src/test/groovy/org/nuclos/test
```

Starten der Server-Integrationstests

1. Starten einer *.groovy Datei unter "server" oder des ganzen server Verzeichnis (mit Run oder Debug)
2. Testergebnisse abwarten

Starten der Webclient-Integrationstests

1. Starten des Docker-Containers "selenium/standalone-chrome-debug" auf port 4444
z.B. mit:

```
docker run -d -p 4444:4444 -p 5900:5900 selenium/standalone-chrome-debug
```

2. Eintragen folgenden Hosts in die "/etc/hosts" (Windows: "%windir%\system32\drivers\etc\host")

```
127.0.0.1      host.docker.internal
```

3. Starten des Webclients mit:

```
ng serve --host host.docker.internal
```

4. Einrichten der VM Parameter in IntelliJ mit "Edit Configurations" "JUnit" "Edit configuration templates..."
Beispiel:

```
-ea
-Dselenium.server=http://127.0.0.1:4444/wd/hub
-Dnuclos.server.host=host.docker.internal
-Dnuclos.server.port=8080
-Dnuclos.webclient.host=host.docker.internal
-Dnuclos.webclient.port=4200
-Dnuclos.webclient.screenshots=false
-Dnuclos.failing.videos.path=/home/honk/videos
```

(Im obigen Beispiel wären "nuclos.server.port" (default: 8080) und "nuclos.webclient.host" (default identisch zu "nuclos.server.host") nicht nötig. Die beiden unteren Parameter "nuclos.webclient.screenshots" und "nuclos.failing.videos.path" sind grundsätzlich optional). Ohne "selenium.server" downloaden und starten die Tests einen separaten Docker-Container, der anschließend wieder gelöscht wird.

5. Starten einer *.groovy Datei unter "webclient" oder des ganzen server Verzeichnis (mit Run oder Debug)
6. Testergebnisse abwarten