

Autofac injecten

Da mich das eben ein paar Minuten gekostet hat, will ich das einmal festhalten.

Folgendes NuGet muss im Projekt inkludiert sein:
Autofac.Extensions.DependencyInjection

HostApplication:

```
[crayon-676703d87e2ec503179254/]
```

WebApplication:

```
[crayon-676703d87e2f4904714379/]
```

Docker auf Windows installieren ohne Docker Desktop

Die DockerCLI und DockerEngine sind Freeware und OpenSource, im Business Bereich muss die Docker Desktop Anwendung erworben werden. Um Docker nutzen zu können, braucht man nicht unbedingt Docker Desktop. Auch wenn Docker Desktop einem bei der Einrichtung vieles einfacher macht. Hier soll eine Schritt für Schritt Anleitung folgen

Alle Schritte nutzen PowerShell oder meine Empfehlung „Windows Terminal Canary“

WSL (Windows-Subsystem für Linux)

Über WSL läuft eine Linux distro unter Windows.

hat man noch kein WSL installiert, kann über diesen Befehl eine Distro ausgewählt werden. Standard ist Ubuntu, ich nutze hier debian.

```
[crayon-676703d87e7ef368940516/]
```

um WSL zu installieren gibt man folgenden Befehl ein

```
[crayon-676703d87e7f3908076581/]
```

Da das Linux System CPU und RAM Ressourcen verbraucht, kann man diese einschränken. Dazu wird eine .wslconfig Datei im Userverzeichnis erstellt

```
[crayon-676703d87e7f4005737246/]
```

In Windows Explorer %UserProfile% eingeben und die erstellte .wslconfig optional bearbeiten:

```
[crayon-676703d87e7f5840547078/]
```

jetzt **wsl** eingeben und wir befinden uns in der Kommandozeile von Linux

Als erstes sollte man die distro aktualisieren

```
[crayon-676703d87e7f7572518950/]
```

Optional: WSL Image verschieben

Es kann Sinnvoll sein, die WSL Images nicht im Standard Verzeichnis zu haben. Daher kann die WSL Image jederzeit umziehen.

1. Name des WSL-Images bestimmen. In meinem Fall ist es „Debian“:

```
[crayon-676703d87e7f8305210094/]
```

2. als tar exportieren:

```
[crayon-676703d87e7fa288923574/]
```

3. In WSL das image lösen:

- [crayon-676703d87e7fb142534639/]
4. 2 Verzeichnisse erstellen. Mein neues Verzeichnis ist D:\Docker\wsl
[crayon-676703d87e7fc688813380/]
 5. WSL herunterfahren:
[crayon-676703d87e7fd702398954/]
 6. Import starten: wsl -import Debian
[crayon-676703d87e7fe717520645/]
 7. Neues Image starten:
[crayon-676703d87e7ff666616144/]
 8. Optional:
[crayon-676703d87e800723965711/]

Docker installieren und einrichten

Zunächst werden benötigte Repositories für das Generieren von Zertifikaten installiert und dann docker selbst. Wichtig: Die Zeilen hier unten so wie die sind nacheinander, Zeile für Zeile kopieren. Nicht mehrere Zeilen miteinander in das Terminal einfügen. Aufforderungen mit Y und Enter bestätigen
[crayon-676703d87e801976609123/]

[crayon-676703d87e802997909811/]

Docker Daemon starten. Wenn der nicht weiter geht, Terminal schließen und neuen Terminal mit **wsl** starten

[crayon-676703d87e804443332115/]

Docker müsste jetzt installiert sein. (optional) Das testen wir, indem wir ein hello-world container laufen lassen:

[crayon-676703d87e805069745465/]

Damit unser aktueller User nicht immer sudo eingeben muss, können wir ihn zu der Gruppe der Docker Administratoren hinzufügen. Evtl. ist die Gruppe bereits vorhanden, dann Meldung ignorieren.

[crayon-676703d87e806482600735/]

Um sicherzustellen, dass Docker bei jedem Neustart läuft, folgende Befehle ausführen:

[crayon-676703d87e807047344472/]

Damit auch der dockerd nach dem Start läuft, muss man ein script einfügen. systemctl kann man mit WSL leider nicht nutzen.

Wir legen fest, dass wie sudo bei dem aktuellen Benutzer Benutzer gehandhabt werden soll. Nämlich soll für dockerd kein Passwort erfragt werden:

[crayon-676703d87e808752897865/]

ganz unten folgendes hinzufügen und Benutzer gegeben falls anpassen:

[crayon-676703d87e809746170336/]

Jetzt müssen wir die bash bearbeiten, damit nach reboot das Skript ausgeführt wird

[crayon-676703d87e80a283523105/]

mit „Bild runter“-Taste kann man bis ganz unten blättern und dort folgenden Code einfügen. Mit Strg+X, dann Y und Enter wieder raus:

[crayon-676703d87e80b078883790/]

Shell neustarten:

[crayon-676703d87e80d922260099/]

WSL beenden:

[crayon-676703d87e80e705733060/]

und WSL reboot:

[crayon-676703d87e80f988000374/]

Falls das nicht geht, kann ein Windows Job eingerichtet werden, der wsl automatisch bei Windows Start ausführt

[crayon-676703d87e810147224636/]

Unter Aufgabenplanung in Windows diesen Task als „Unabhängig von der Benutzeranmeldung“ konfigurieren. Sonst funktioniert dieser Task nicht

Linux Docker, Windows bekannt machen

Docker läuft nun auf unserer WSL Linux Maschine. Das nützt uns wenig, da wir in PowerShell nicht immer auf die WSL Linux Maschine zugreifen wollen. Wir wollen einfach docker eingeben und damit die docker Instanz auf der Linux Maschine meinen. Daher erstellen wir jetzt die Brücke.

Zuerst müssen wir das in Linux erlauben. Folgender Befehl öffnet einen Editor:

```
[crayon-676703d87e811474926696/]
```

dort mit Strg+V folgendes einfügen:

```
[crayon-676703d87e812772681328/]
```

Mit Strg+X , dann Y und Enter speichern

Mit **exit** beenden wir die WSL Kommandozeile

Jetzt muss unter Windows eine Umgebungsvariable für diesen Docker Host angelegt werden (Terminal muss als Admin laufen):

```
[crayon-676703d87e813218547990/]
```

Docker CLI unter Windows installieren

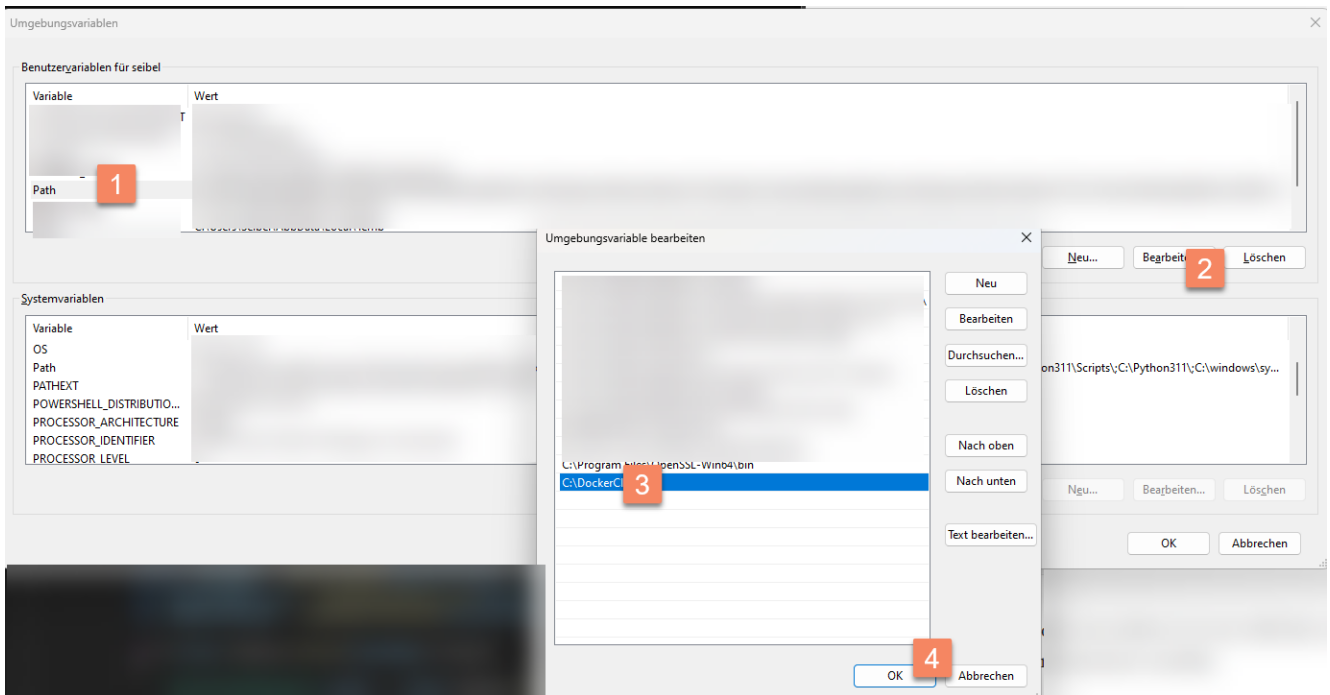
Wenn man bereits choco nutzt, geht das easy mit

```
[crayon-676703d87e814585773112/]
```

Ansonsten hier herunterladen:

https://download.docker.com/win/static/stable/x86_64/

und unter einem Verzeichnis z.B. C:\DockerCli ablegen. Nun muss dieses Verzeichnis in den Umgebungsvariablen unter Path eingetragen werden



über folgenden Befehl müsste Docker nun aufrufbar sein
[crayon-676703d87e815312210485/]

Man muss jetzt wissen, dass es nun 3 Ebenen gibt.

Windows -> WSL Linux -> Docker Container.

WSL hat für Windows automatisch ein mount in dem Ordner /mnt/ eingerichtet

Möchte man nun, dass der Container auf das Windows Verzeichnis C:\Temp bindet, so muss man stattdessen /mnt/c/Temp eingeben

Referenz:

Docker and WSL2 without Docker Desktop | by Romain Bruyère | Medium

files folders – Change the storage location of a WSL2 – Super User

How to automatically start the Docker daemon on WSL2 – NillsF blog

Portainer installieren mit SSL Zertifikat

Portainer über docker installieren

[crayon-676703d87e816300073513/]

! Funktioniert noch nicht. Das Zertifikat wird vom Browser nicht anerkannt !

Zertifikat auf der Linux WSL erzeugen und das Zertifikat nach D:\Docker kopieren:

[crayon-676703d87e817598296132/]

dotNet Docker veröffentlichen

1. Wenn ein eigener nuget Server mit http verwendet wird, dann muss dort wo die sln liegt, eine NuGet.config angelegt werden:

[crayon-676703d87ecb6984143009/]

2. die csproj bearbeiten:

[crayon-676703d87ecc3788398053/]

3. Docker Desktop muss vorinstalliert sein und mit wsl2 laufen

[crayon-676703d87ecc7336369383/]

4. Https Zertifikat für Entwicklung erstellen:
<https://learn.microsoft.com/en-us/aspnet/core/security/docker-https?view=aspnetcore-8.0#running-pre-built-container-images-with-https>

[crayon-676703d87ecca776007771/]

Docker

Docker Desktop (AMD64)

Docker Desktop: The #1 Containerization Tool for Developers | Docker

In der Regel reicht auch nur die cli
[crayon-676703d87efa6093992911/]
und Rancher Desktop
[crayon-676703d87efac691357034/]

Portainer

<https://docs.portainer.io/start/install-ce/server/docker/wsl>
[crayon-676703d87efae417002680/]
WebAdmin: <https://localhost:9443/>

Microsoft Sql Server

microsoft/mssql-server – Docker Image | Docker Hub
[crayon-676703d87efb0405864477/]
Passwort setzen nicht vergessen

Im Container, unter volumes als bind, ein Verzeichnis einrichten, wo die backups liegen und wo die volumes liegen sollen

RabbitMQ

[crayon-676703d87efb2959458119/]

WebAdmin: <https://localhost:15672/>

Benutzername: guest

Password: guest

Hashicorp/Vault

[crayon-676703d87efb5966907305/]

hashicorp/vault – Docker Image | Docker Hub

Weboberfläche: <http://localhost:8200/>

Logging Structured Message Templates selbst verwenden

Um selbst Strukturierte Messages Templates wie im Logging zu verwenden, kann man folgende Annotations nutzen:

[crayon-676703d87f255895327557/]

Discord Installation „a fatal javascript error occurred“

Das war die einzige Lösung, die funktionierte, um Discord zu installieren über choco:

I have a problem with discord, and its no the logo. :
r/discordapp (reddit.com)

Dazu das Powershell-Script. Ausprobiert und getestet.
[crayon-676703d87f3c9265108472/]

Jupyter unter Windows installieren

Mit Jupyter (Julia + Python) kann man interaktive Notebooks erstellen.

Ein Notebook besteht aus Cells, die entweder als Text (markdown) oder Code (Python, etc.) nacheinander erstellt werden können.

Der Code kann im Notebook ausgeführt werden und visuell dargestellt werden. Variablen können Code-Zelle übergreifend genutzt werden.

Damit der Code ausgeführt werden kann, wird ein Kernel benötigt, der wie ein Interpreter den Code verarbeitet und ausgibt.

Um Jupyter nutzen zu können, muss dieser zunächst installiert werden.

Installation Jupyter

Jupyter selbst benötigt Python und für die Installation den Python package manager, welcher in Python inkludiert ist.

```
[crayon-676703d87f559293842202/]
```

Über den Befehl

```
[crayon-676703d87f55e276481339/]
```

wird der JupyterLab Server lokal gestartet. Dieser ist erreichbar unter `http://127.0.0.1:8888`

In der Konsole ist auch der Token hinterlegt, den man für die Ersteinrichtung benötigt.

Editoren

JupyterLab Web

Über `http://127.0.0.1:8888` kommt man auf die Schaltzentrale von JupyterLab.

Dort kann man auch die Notebooks erstellen und bearbeiten

Azure Data Tool

My experience working with notebooks in Azure Data Studio – Data on Wheels – Kristyna Ferris & Steve Hughes (wordpress.com)

JupyterLab unterstützt von Haus aus nur Python als Kernel. Weitere Kernels müssen nachinstalliert werden.

Leider gibt es kein Kernel für SQL. Aber Azure Data Tools hat ein eingebautes Kernel für SQL Server.

Die Nutzung inkl. Copilot ist wirklich komfortabel.

VS Code

Für VsCode müssen die Jupyter Extensions heruntergeladen werden.

Über Strg + Shift + P kann eine neue Jupyter Datei erstellt werden.

Oben rechts in der Ecke wählt man „Select Kernel“ aus und gibt als Adresse `http://127.0.0.1:8888` und das Passwort ein.

Danach kann man die Kernels aus JupyterLab nutzen

nteract

nteract ist ein weiterer Editor um Jupyter Notebooks zu erstellen. Dieser kann mit CodeMirror und Monaco arbeiten.

nteract: write your next code-driven story.

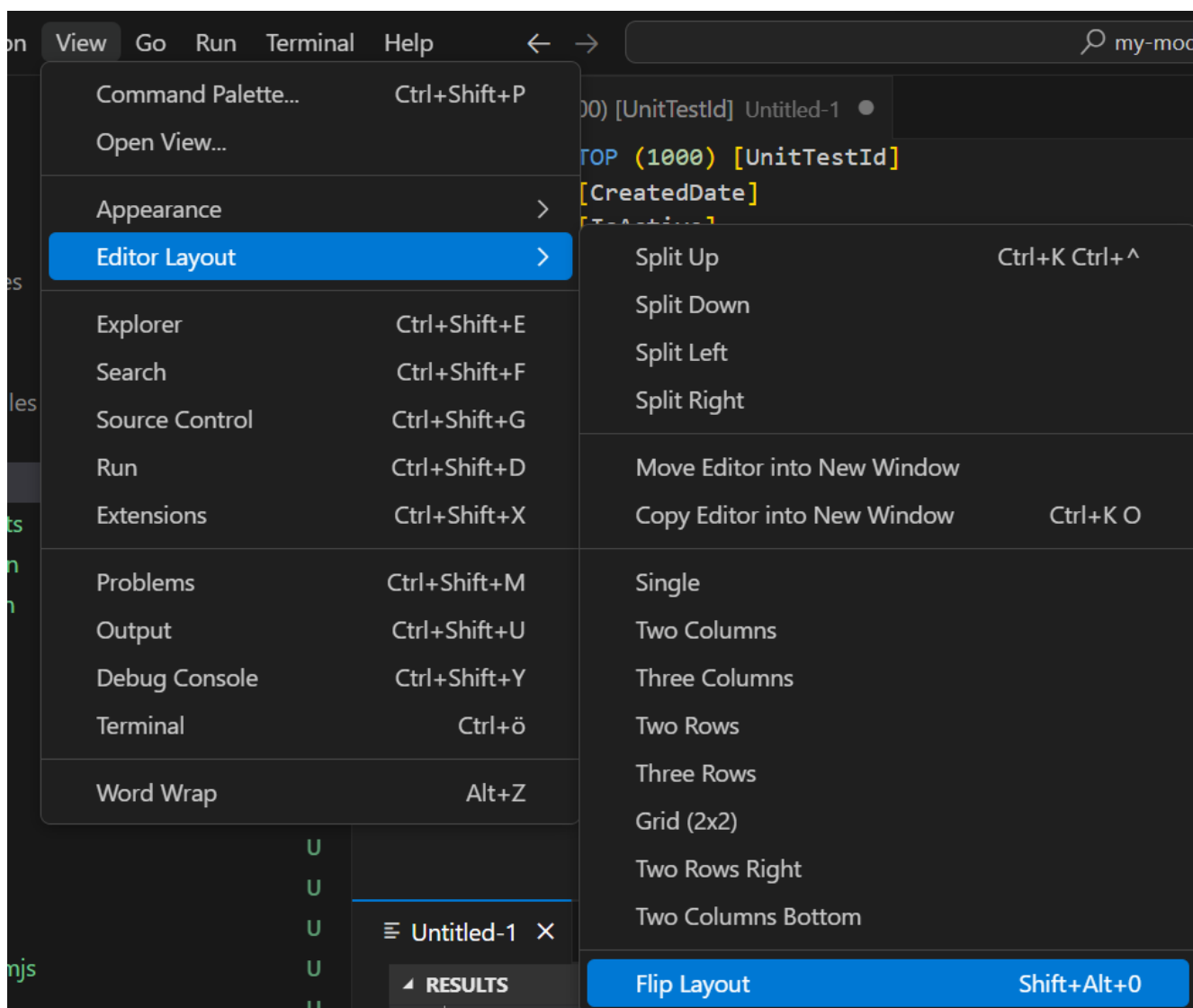
Weitere Kernel

[crayon-676703d87f560000547225/]

.NET Interactive (C#, F#, PowerShell)

[crayon-676703d87f563583326143/]

VS Code SQL Fenster horizontal ausrichten



Cronicle in Rocky Linux installieren

Cronicle

nodeJs über dnf installieren. Nicht über nvm, weil nvm nur für

den jeweiligen User installiert

```
[crayon-676703d87f7b3703610483/]
```

So wie hier beschrieben, manuelle Schritte durchführen.
Wichtig, vorher die aktuelle Version herausfinden und im
Archiv ersetzen

```
[crayon-676703d87f7b8189483241/]
```

Firewall Port freischalten

```
[crayon-676703d87f7b9545800635/]
```

Maschine neustarten

```
[crayon-676703d87f7ba112666243/]
```

rc1one OneDrive

rc1one Konfigurieren

1. zu mnt Verzeichnis bewegen mit cd und ls

2. OneDrive Ordner erstellen:

```
[crayon-676703d87f95e597692117/]
```

3. rc1one installieren

```
[crayon-676703d87f962031001162/]
```

4. Neue rc1one config erstellen

```
[crayon-676703d87f963176069386/]
```

5. n -> New remote

Name: OneDrive

Nach Nummer suchen. In meinem Fall 27 Microsoft OneDrive

client_id mit Enter überspringen, client_secret mit Enter auch

überspringen

1 für Microsoft Cloud Global aussuchen (Evtl. geht auch 3 Germany)

Edit advanced config? -> n

Der nächste Teil ist ein wenig tricky. Denn man benötigt den Microsoft 2FA Secret. Diesen bekommt man nur, wenn man über einen Browser die Anmeldung bestätigt.

Da die Linux Maschine nicht über einen Webbrowser verfügt, muss man dies auf Windows Rechner machen und die rclone config auf die Linux Maschine übertragen.

OneDrive Secret Token besorgen

1. Chocolatey installieren: [Chocolatey Software | Installing Chocolatey](#)

2. rclone installieren:

```
[crayon-676703d87f965602976439/]
```

3. OneDrive authentifizieren.

```
[crayon-676703d87f966625583924/]
```

Es erscheint in PowerShell Konsole der Token:

```
[crayon-676703d87f967765767257/]
```

Dieser Token muss kopiert werden.

In der Linux Maschine sind wir noch beim Konfigurieren.

Die Frage ob ein Browser geöffnet werden soll wird mit n beantwortet und der token dort eingefügt.

4. Choose a number from below, or type in an existing value
-> 1/ OneDrive Personal or Business

5. Found drive „root“ of type „personal“ -> y für yes, default

6. Nächste Frage wieder mit y -> y) Yes this is OK (default)

7. y) Yes this is OK (default)

8. q -> Quit Config.

Die Konfiguration ist abgeschlossen.

OneDrive Storage mounten

Fuse installieren:

```
[crayon-676703d87f968676036191/]
```

mounten

```
[crayon-676703d87f96a838922330/]
```

Es sollte erscheinen [1] und die wie groß das Laufwerk ist.

wenn jetzt zu dem Verzeichnis navigiert wird, kann mit `ls` die Struktur ersehen werden.

Die Dateien sind nicht abgerufen und synchronisiert.