

Installationsanleitung

Virtual Device Pool Manager (VDPM)

Installationsanleitung	1
Mindestanforderungen.....	2
Hinweis zur Verwendung von SSL.....	2
Klonen des VDPM GitHub Repository.....	2
Installieren der Abhängigkeiten.....	2
Setup der Applikation	3
Starten in der Entwicklungsumgebung.....	4
Building.....	4
Zusätzliche Voraussetzungen für die Produktionsumgebung.....	4
Produktionsumgebung starten	6

Mindestanforderungen

Die Applikation wurde mit folgenden Versionen entwickelt & getestet:

- MariaDB 10.11.2 <https://mariadb.org/>
- Node.js 18.15.0 <https://nodejs.org/en/>
- NPM 9.5.0 <https://www.npmjs.com/>

Hinweis zur Verwendung von SSL

Die Applikation unterstützt die Verwendung von SSL. Dazu muss ein gültiges Zertifikat und ein privater Schlüssel im PEM-Format vorhanden sein. Diese müssen im Verzeichnis `certs` abgelegt werden. Die Dateinamen müssen `cert.pem` und `key.pem` lauten.

Das Repository beinhaltet ein selbst signiertes Zertifikat für die Verwendung unter «localhost». Damit die Applikation mit diesem Zertifikat genutzt werden kann müssen diese ggf. zuerst im Browser akzeptiert werden. Öffne dazu die URL `https://localhost:[PORT]` und ersetze beim Aufruf im Browser [PORT] mit dem von dir definierten API-Port aus dem Setup Prozess, standardmässig wird der Port 3000 verwendet. Der Browser wird eine Warnung anzeigen, dass das Zertifikat nicht vertrauenswürdig ist. Klicke auf "Erweitert" und anschliessend auf "Zertifikat ansehen". Anschliessend kannst du auswählen, das dem Zertifikat vertraut werden soll.

Beim Öffnen des GUI unter (Standart-) Port 8080 zeigt ebenfalls den Hinweis das diesem Selbstsignierten Zertifikat nicht automatisch vertraut werden kann. Wiederhole die oben genannten Schritte.

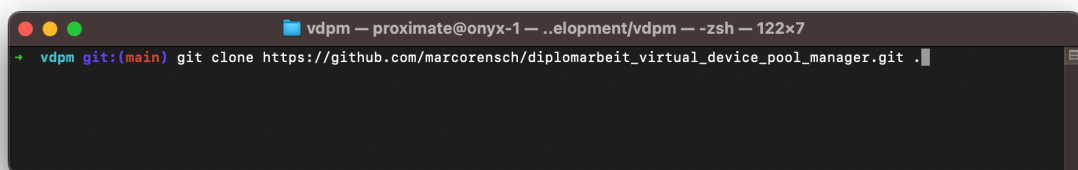
Hinweis: Das Vorgehen kann je nach Browser variieren.

Für den produktiven Einsatz müssen eigene SSL-Zertifikate genutzt werden.

Klonen des VDPM GitHub Repository

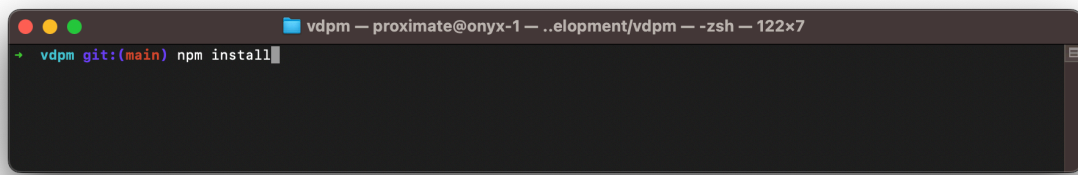
Das GitHub Repository zum Projekt kann über die folgende URL geklont werden:

https://github.com/marcorensch/diplomarbeit_virtual_device_pool_manager.git



Installieren der Abhängigkeiten

Um die Projektabhängigkeiten zu installieren kann der Befehl `npm install` im Projektverzeichnis ausgeführt werden. Es werden automatisch alle Abhängigkeiten für Setup, Server & Client installiert.

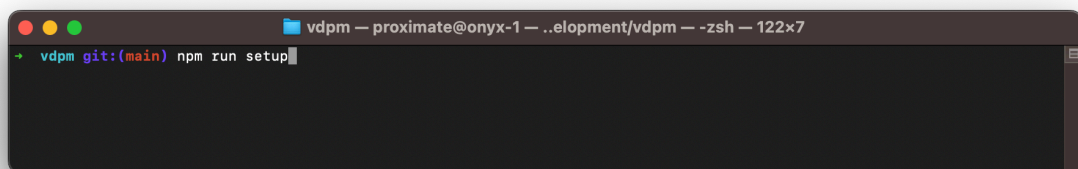


```
vdpm — proximate@onyx-1 — ..elopment/vdpm — -zsh — 122x7
+ vdpm git:(main) npm install
```

Nach der Installation kann geprüft werden ob alle Ordner (client, server, setup, certs) vorhanden sind. Ist dies der Fall kann mit dem Setup begonnen werden.

Setup der Applikation

Stelle vor dem Start des Setup Prozesses sicher das Deine Datenbank (MariaDB oder MySQL) gestartet & erreichbar ist. Starte den Setup Prozess mit dem Befehl `npm run setup` direkt aus dem Projektverzeichnis heraus.



```
vdpm — proximate@onyx-1 — ..elopment/vdpm — -zsh — 122x7
+ vdpm git:(main) npm run setup
```

Der Setup Prozess stellt dir nun die folgenden Fragen um die Applikation einzurichten. In Klammern findet sich jeweils der Standardwert.

- Server API host
 - o Die IP Adresse auf welchem die Server API gestartet wird, es wurde lediglich die Verwendung als «localhost» resp. 127.0.0.1 getestet.
- Server API port
 - o Der Port über welcher die API erreichbar sein wird
- Use SSL
 - o Ob der Server mit SSL Unterstützung gestartet werden soll. Ein Self-Signed SSL Zertifikat für «localhost» befindet sich im Projektverzeichnis «certs» und kann bei Bedarf ersetzt werden.
- Database host
 - o Hostadresse der MariaDB Datenbank
- Database port
 - o Port über welche die MariaDB erreichbar ist
- Database user
 - o Benutzer für die Anmeldung in der Datenbank
- Database password
 - o Passwort für den zuvor genannten User
- Create test database
 - o Für die Tests wird eine Test-Datenbank genutzt / benötigt
- Test database host
 - o Hostadresse der MariaDB Datenbank die für Tests verwendet werden soll
- Test database port
 - o Port über welche die Test Datenbank erreichbar ist

- Test database user
 - Datenbank-Benutzer der für die Tests verwendet werden soll
- Test database password
 - Das Passwort für den zuvor genannten User auf der Test-Datenbank
- JWT expiration time
 - Normale Session-Dauer eines Users
- JWT refresh expiration time
 - Erweiterte Session-Dauer eines Users in welcher kein erneuter Login nötig ist
- Admin password
 - Das Passwort für den ersten Administrator der Applikation, wird kein Passwort definiert so generiert das Skript ein Passwort

Nachdem die letzte Frage beantwortet wurde startet der Konfigurationsskript, installiert die nötige Datenbanken, Tabellen und speichert benötigte Inhalte in die Datenbank. Sollte hier ein Fehler angezeigt werden stelle bitte sicher das deine Datenbank erreichbar ist und du die korrekten Benutzerdaten für die Verbindung mit der Datenbank hinterlegt hast. Sollte die Installation fehlschlagen wird dir der Skript weiterführende Informationen zum Fehler anzeigen.

```

vdpm — proximate@onyx-1 — ..elopment/vdpm — -zsh — 122x24
Done!
Executing 15_tbl_device_documents.sql
Done!
Executing 16_tbl_weblinks.sql
Done!
Executing 20_tbl_guides.sql
Done!
Executing 21_tbl_guide_slides.sql
Done!
Executing 22_tbl_guides_devices.sql
Done!

***** User created! *****

Username: administrator
Password: 
*****

-----
You can now using npm run devStart to start your server locally.
-----

→ vdpm git:(main)

```

Starten in der Entwicklungsumgebung

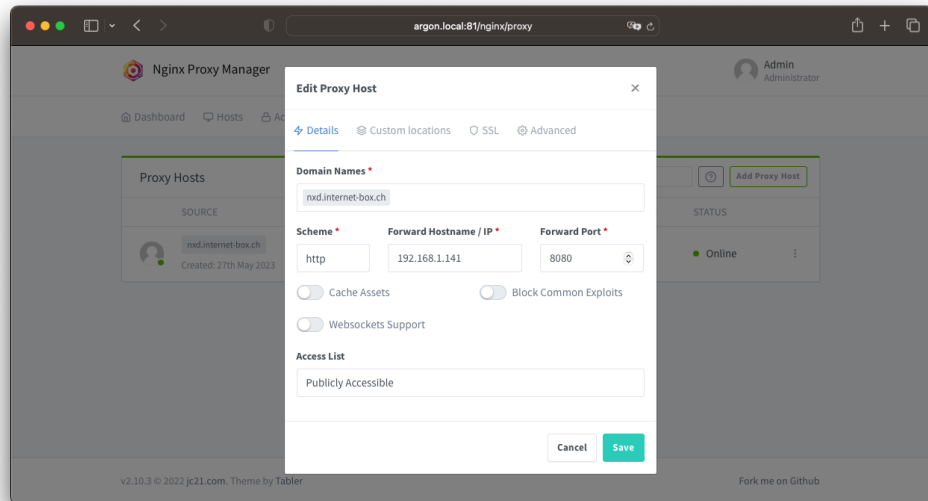
Die Applikation kann nun mit dem Befehl `npm run devStart` aus dem Projektverzeichnis heraus gestartet werden.

Building

Die Applikation (Client) kann über den Befehl `npm run build` für die Produktion gebaut werden. Durch den Aufruf des Skripts wird der Vue-Client gebaut und im Verzeichnis «./client/dist» abgelegt.

Zusätzliche Voraussetzungen für die Produktionsumgebung

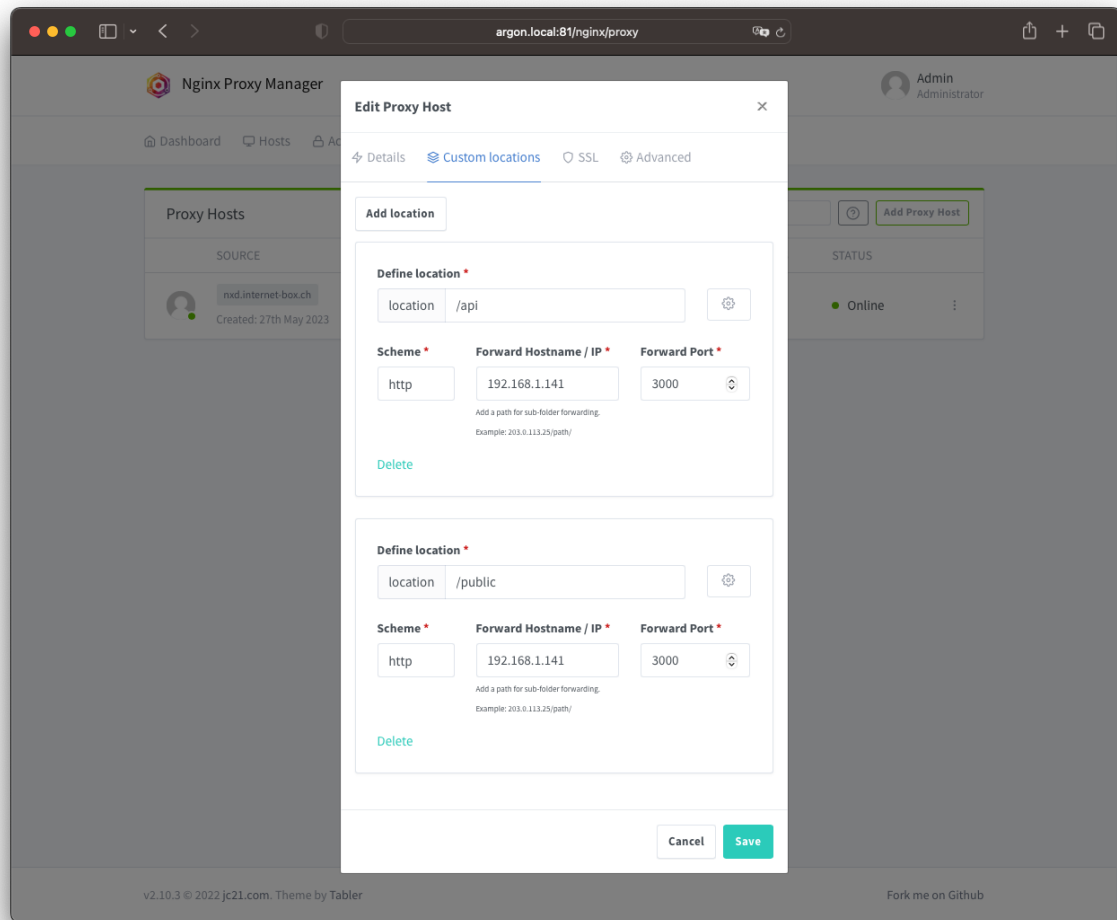
Damit die Applikation in einer Produktiv-Umgebung genutzt werden kann wird ein NGINX-Server benötigt. An dieser Stelle wird nicht detailliert auf die Konfiguration eines NGINX-Servers eingegangen. Damit die Client-Oberfläche angesteuert werden kann muss in der Konfiguration der (Standard-)Port «8080» verwendet werden.



Für den entsprechenden Host müssen im Proxy zudem die folgenden Locations definiert werden:

Location (Pfad)	Port
/api	3000 (Serverport)
/public	3000 (Serverport)

Bei der Verwendung des NGINX Proxy Managers kann eine Konfiguration ohne SSL zum Beispiel wie folgt aussehen:



Die oben gezeigte Konfiguration routet Anfragen für die Pfade /api sowie /public an den Server mit der IP 192.168.1.141 unter Port 3000 weiter. Wird ein alternativer Port (nicht 3000) beim Setup verwendet müssen die Ports entsprechend angepasst werden.

Produktionsumgebung starten

Wenn der Client gebaut wurde, kann im Projektverzeichnis mittels dem Beispiel Skript `npm run start` die Produktionsumgebung gestartet werden.

```

vdpm — npm run start — npm — node < npm run start __CFBundleIdentifier=com.apple.Terminal TMPDIR=/var/folders/yp/0...
[+] vdpm git:(main) npm run start
> virtual_device_pool_manager@1.0.0 start
> conc -n="server,client" -c="green,blue" "cd server && cross-env USE_SSL=false npm run start" "node serve.mjs"

[client] Client available on port 8080
[server]
[server] > server@1.0.0 start
[server] > cross-env NODE_ENV=production node server.mjs
[server]
[server] Server listening on port 3000 without SSL

```

Die Konsole zeigt nach dem erfolgreichen Start über welche Ports Client & Server erreichbar sind.