

Deployment:

Sie können die Software auch einfach auf unserem Server testen, hierfür haben wir Ihnen folgende Daten bereitgestellt:

IP-Adresse: 185.233.106.149

Benutzername: admin@example.com

Passwort: !Kantinerado2024!

Hinweis: Dieses Profil hat damit sämtliche Berechtigungen.

Benutzername: kantinenmitarbeiter@example.com

Passwort: !Kantinerado2024!

Hinweis: Dieses Profil ist als Kantinenmitarbeiter angelegt und hat die damit verbundenen Berechtigungen.

Benutzername: hungernde@example.com

Passwort: !Kantinerado2024!

Hinweis: Dieses Profil ist als Hungernder angelegt und hat die damit verbundenen Berechtigungen.

Vorraussetzungen:

- Das gesamte Github-Repository muss bereits vollständig heruntergeladen sein. (<https://github.com/Zensurversuch/Kantinerado>)
- Python installiert (<https://www.python.org/downloads/>)
- Docker installiert (<https://docs.docker.com/get-docker/>)
- Node.js bzw. NodePackageManager (npm) installiert (<https://nodejs.org/en/download>)

Vorgehen:

Der Vorgang des Deployments ist durch einen Befehl vollständig:

1. Gehen Sie in den Ordner (/Development/Local-Environment/ 2. Gehen Sie in die Console und führen Sie folgenden Befehl aus:

○ `docker-compose up --build`

Hinweis: Sofern Sie das Deployment lokal durchführen möchten, müssen Sie in dem Ordner /Development/Local-Environment eine ".env" mit folgendem Inhalt erstellen:

POSTGRES_PW=<Passwort>s
JWT_SECRET_KEY=9e7fG3q8!\$aPz&5rTb@KuYwXmCvF1s
ANGULAR_ENVIRONMENT=development

Datenbank aufsetzen mit Backup

Im Ordner WebApp/Database finden Sie das die Datenbankbackup DummyDatabase.sql. Dieses muss in die Datenbank geladen werden, während die Anwendung läuft und solange die Datenbank noch leer (nicht vollständig initialisiert) ist. Hierfür kann man ein beliebiges Datenbanktool nehmen. Dieses Backup initialisiert aber nur die Tabellen und legt Nutzer mit zugehörigen Tabellen an. Es wird sonst alles leer gelassen. Ein Beispiel wie man ein Backup lädt ist in unserer README auf der Startseite des Repos.

Hinweis: Eine Testdatenbank, mit vielen Daten, kann durch WebApp/Database/DummyDatabasePresentation.sql initialisiert werden.

Initialisierung mit Python:

Wenn etwas am Datenbankmodell geändert wurde, muss man ein Python Skript ausführen, mit welchem das Datenbankmodell in die Datenbank geladen wird. Hierfür gibt es die Skripte „Web-App/Database/initialize_db_local.py“ und „Web-App/Database/initialize_db_server.py“. Das lokale Skript muss man ausführen, wenn man die Datenbank auf seiner lokalen Entwicklungsumgebung initialisieren will und das Server Skript, wenn man eben eine DB auf einem Remote Server initialisieren will (hier muss dann die IP im Skript angepasst werden).

Hinweis: Wenn man über diesen Weg eine ganz leere Datenbank initialisiert muss man einen admin Benutzer erstellen, das Problem ist, dass dies eigentlich nur ein Admin kann. Der Weg, um das zu machen erstellt man einfach einen Nutzer über Registrieren und ändert hier in der Datenbank dann die Rolle von hungernde zu admin. Dann ist dieser Benutzer ein Admin welcher alles Weitere machen kann. Außerdem müssen Allergien von Hand in der Datenbank angelegt werden

Der sauberste Weg dies zu machen ist über eine virtuelle Python Umgebung, sodass die Packages nicht global auf dem Rechner abgespeichert werden. Wenn gewünscht kann man dies auch ohne virtuelle Umgebung machen.

Es ist wichtig das man bevor man das Python Skript ausführt das man in den Database Ordner navigiert und dieses dann dort per Konsole ausführt, sodass die Abhängigkeiten die auf ein anderes Verzeichnis verweisen richtig gezogen werden.

1. Erstellen Sie eine virtuelle Python Umgebung
 - Windows: `python -m venv .venv`
2. Stellen Sie den Interpreter der virtuellen Umgebung auf Python (venv)
 - Windows: `.venv\Scripts\python.exe`

3. Aktivieren Sie die Virtuelle Umgebung.

- Windows: `venv\Scripts\activate`

4. Installieren Sie die verschiedenen Bibliotheken über den Package Installer Python (pip):

- Windows: `pip install -r "requirements.txt"`

5. Initialisieren der Tabellen durch das Python Script für Lokal: ○

- Windows: `python intialize_db_local.py`

Initialisieren der Tabellen durch das Python Script für Server:

- Windows: `python initialize_db_server.py`

Hinweis: Passen Sie hierfür die Adresse im Script an.