

## 1. Installation

---

Dieses Kapitel beschreibt die Installation der Anwendung auf einem beliebigen Betriebssystem.

### 1.1 Voraussetzungen

Die minimalen Voraussetzungen für die Ausführung der Software sind ein von Docker offiziell unterstütztes Betriebssystem sowie eine offiziell unterstützte Plattform. Wir empfehlen unterstützte **Linux Distributionen**, sowie **x86\_64** als Plattform. Hier muss die **Docker Engine** lauffähig installiert sein, um die Ausführung der Anwendung zu ermöglichen (Installationsanleitung: [Linux](#), [Windows](#), [macOS](#)). Zusätzlich wird empfohlen git installiert zu haben ([Git Installationsanleitung](#)).

### 1.2 Installation der Anwendung über git

Wir empfehlen die Installation der Anwendung über **git**. Hierfür sollte man im Terminal, in einer beliebigen shell (empfohlene Referenzshells: bash oder zsh), in den gewünschten Verzeichnis/Ordner für die installation navigieren, was in den meisten shells durch `cd {Pfad zu gewünschtem Installationsordner}` erreicht werden kann. Anschließend wird die Anwendung via `git clone {Pfad zu Repository auf Gitprovider}` installiert.

## 2. Ausführung

---

Dieses Kapitel beschreibt die Ausführung der Software mithilfe von docker auf verschiedensten Betriebssystemen.

### 2.1 Bau eines lauffähigen Docker Image

Wenn die Anwendung korrekt installiert wurde, findet sich in ihr das **Dockerfile**. Anhand von diesem kann man mit **docker**, auf allen unterstützten Betriebssystemen, einfach ein lauffähiges **Docker Image** der Anwendung erstellen. Hierfür sollte im Terminal der Befehl `docker build {Pfad zum Rootverzeichnis der Anwendung (in der das Dockerfile liegt)} -t [Name des Image]` ausgeführt werden (in einigen Systemen kann es nötig sein dies Administrator Berechtigungen auszuführen (in Linux `sudo` vor den Befehl)).

### 2.2 Ausführung des Docker Image

Wenn das Docker Image ohne Fehler erstellt wurde, kann es mit dem Befehl (Referenz Linux mit zsh und bash shell) `docker run -i [Name des Image] < {Pfad zum Inputfile} > {Pfad zum gewünschten Outputfile}` ausgeführt werden. Hier wird das Inputfile durch `< {Pfad zum Inputfile}` in stdin des Containers gelesen, wodurch dieser damit arbeiten kann. Der stdout Output des Containers wird durch `> {Pfad zum gewünschten Outputfile}` in ein gewähltes Outputfile geparkt, dort findet sich dann der berechnete Fahrplan (je nach Betriebssystem sollte hierfür die Dateieindung `.txt` gewählt werden).

---

## 3. Anmerkungen zu diesem Handbuch

---

Dieses Handbuch soll die Installation und Nutzung der Software möglichst jedem Benutzer unabhängig von dessen Wahl des Betriebssystems ermöglichen. Jedoch können wir nicht für die Allgemeingültigkeit dieser Anweisungen garantieren. Wir konnten die Software vor allem auf Linux und Windows Testen, da wir kein MacOS verwenden. Zudem sind die hier gezeigten Befehle alle auf die Linux Shells, welche wir getestet haben (bash und zsh) bezogen, es könnte auf anderen Shells und Betriebssystemen eine andere Syntax erforderlich sein. Falls es zu Problemen kommt empfehlen wir diese Ressourcen: [Linux](#), [Windows](#) und [macOS](#).