Traintastic aus dem Quellcode erstellen

Diese Seite beschreibt die Erstellung von Traintastic unter Debian Trixie und Raspian OS Trixie. Um Traintastic aus dem Quellcode zu erstellen, sind ein paar wenige Voraussetzungen nötig. Als erstes muss Du die Build Tools installieren.

Installation der Build Tools

Du kannst die Tools manuell installieren, oder Dir dieses kleine Script erstellen.

```
#!/bin/bash
#
# This script installs all requirements to build traintastic
#
# 2026-04-24 Tom, DL7BJ
#
sudo apt install mc # optional
sudo apt install vim # optional
sudo apt install cmake
sudo apt install build-essential libboost-all-dev libsystemd-dev
sudo apt install libarchive-dev
sudo apt install liblua5.4-dev
sudo apt install qt6-base-dev qt6-tools-dev qt6-tools-dev-tools
sudo apt install qt6-svg-dev
sudo apt install qt6-websockets-dev
sudo apt install qt6-image-formats-plugins
exit 0
```

Download des Quellcodes

Nach dem Du die Build Tools installiert hast, benötigst Du den Quellcode. Das Traintastic Repository hat ein paar Submodules, so dass Du diese am besten gleich mit herunterlädst. Es ist sinnvoll, für den Quellcode ein Verzeichnis zu erstellen, wie *dev* oder *projects*. Dann wechselt Du in dieses Verzeichnis und erstellst einen Klon der Quelltexte von Github.

```
git clone –recurse-submodules https://github.com/traintastic/traintastic.git
```

Wenn Du z,B. das Verzeichnis *dev* verwendet hast, findest Du dort nun ein Verzeichnis *traintastic*. Dies ist das Verzeichnis, welches ich im weiteren als $TRAINTASTIC_PATH bezeichne. Du kannst diesen Verzeichnisnamen in eine Umgebungsvariable exportieren:

```
export TRAINTASTIC_PATH=$HOME/dev/traintastic
```

Erstellen des Traintastic Servers

Für die Erstellung des Servers wechselt Du in das Verzeichnis *$TRAINTASTIC_PATH/server*. Um die Build Umgebung zu initialisieren, rufst Du diesen Befehl auf, der nur einmalig benötigt wird (außer Du möchtest alles noch mal neu bauen).

```
cmake -S . -B build
```

If the build directory $TRAINTASTIC_PATH/server/build exists, you can type these

```
cmake –build build –target traintastic-server
```

Auf einem Raspberry Pi 4 dauert dies ein paar Stunden. Du kannst es aber massiv beschleunigen, in dem Du angibst, das cmake alle Prozessorkerne verwenden soll. Das geht mit der Option 'j'.

```
cmake –build build –target traintastic-server -j4
```

Auf einem PC kannst Du natürlich noch mehr Kerne angeben, wenn diese vorhanden sind.

Erstellen des Clients

Für die Erstellung des Clients musst Du in das Verzeichnis *$TRAINTASTIC_PATH/client* wechseln. Wenn noch kein *build* Verzeichnis existiert, musst Du die Build-Umgebung initialisieren, wie schon beim Server.

```
cmake -S . -B build
```

Danach kannst Du den Client bauen.

<code bash> cmake –build build –target traintastic-client -j4 </code

Erstellen der Anleitung

Verzeichnisse

The *data directory* is the location where Traintastic server stores all its data, such as: settings, worlds, logfile, backups. If you don't install Traintastic, only build it, then the default location for Linux is:

`$HOME/.config/traintastic-server`

Without installation you must create the directory for the language files manually. Normally Traintastic uses for the language files `/opt/traintastic/tranlations` You can create there some directories, but you have has normal user no write access on /opt, so you must use `sudo`

`sudo mkdir -p /opt/traintastic` `sudo mkdir -p /opt/traintastic/bin` `sudo mkdir -p /opt/traintastic/icons` `sudo mkdir -p /opt/traintastic/lncv` `sudo mkdir -p /opt/traintastic/manual` `sudo mkdir -p /opt/traintastic/translations`

Set the owner and the group of this directory to your username: `chown -R tom:tom /opt/traintastic` Copy all the tranlastion files from the Traintastic source `cp -a $TRAINTASTIC_PATH/shared/translations/* /opt/traintastic/translations`

From:
https://isnix.de/ - **It's boring when it works!**

Permanent link:
https://isnix.de/doku.php?id=modellbahn:software:traintastic:buildfromsource

Last update: **2026-05-09 15:34**