

Installationsanleitung für Server

na17b

Inhaltsverzeichnis

1	Hinweise	1
2	Vorbereitung des Servers	1
2.1	Webserver	1
2.1.1	Apache	1
2.1.2	Via Node-Skript	2
2.2	Printservice	2
2.3	QuitStore	2
3	Erstellen des Quelltextes	2
3.1	Konfiguration	2
3.2	Kompilieren	3
3.3	Bereitstellen des Quelltexts	3

1 Hinweise

Das Repository sollte direkt vom Server aus geklont werden, auf dem später die Software laufen soll. Auch wenn der Quelltext für die Webseite prinzipiell überall kompiliert und übertragen werden kann, so befinden sich im Repository jedoch einige Skripte, welche direkt auf dem Webserver ausgeführt werden müssen und ansonsten also ohnehin kopiert werden müssten. Dazu muss jedoch `nodejs` installiert sein.

```
sudo apt install nodejs
```

2 Vorbereitung des Servers

Die Software besteht serverseitig aus mehreren Komponenten. Dies ist zum einen der Webserver, welcher den Quelltext der Webanwendung zur Verfügung stellt, sowie ein Printservice, welcher übermittelte Daten in eine PDF-Datei umwandelt und zum Download anbietet, und zu guter Letzt der QuitStore, welcher die von den Anwendern eingegebenen Daten abspeichert.

2.1 Webserver

Damit Clients auf die Anwendung zugreifen können, muss diese von einem Webserver angeboten werden. Hierzu gibt es mehrere Möglichkeiten.

2.1.1 Apache

Dieser kann in allen gängigen Linux-Distributionen aus den offiziellen Paketquellen installiert werden. Unter Ubuntu/Debian/Raspbian geschieht dies zum Beispiel folgendermaßen.

```
sudo apt install apache2
```

2.1.2 Via Node-Skript

Sollte obiges keine Option sein, befindet sich im Ordner `frontend/src` ein Skript namens `server.js`. Dieses bindet im Normalfall auf Port 8080 und wird via `nodejs` ausgeführt. Dies geschieht jedoch erst, nachdem der Quelltext kompiliert wurde.

```
node server.js
```

2.2 Printservice

Der Printservice verwendet Python3, um ein tex-Template auszufüllen und dieses anschließend zu kompilieren. Die dafür notwendigen Pakete lassen sich wie folgt installieren.

```
sudo apt install texlive texlive-lang-german texlive-doc-de \
texlive-latex-extra python3
```

Um diesen zu starten, genügen folgende Befehle von der Wurzel des Repositories aus.

```
cd Projekt/src/renderPDF
python3 server.py
```

2.3 QuitStore

Für die Installation und das Starten des QuitStores via Docker liegt ein Skript bereit, dieses muss lediglich ausgeführt werden.

```
cd Projekt
chmod +x quitstore.sh
./quitstore.sh
```

Alternativ dazu wurde auch ein npm-Skript angelgt, welches den QuitStore bei Bedarf installiert und startet.

```
cd Projekt/src/frontend
npm run quit
```

3 Erstellen des Quelltextes

Nachdem nun alle Dienste laufen, fehlt noch der eigentliche Quelltext der Webapp. Zur Steigerung der Performance muss dieser nach der Konfiguration kompiliert werden und durchläuft dabei mehrere Optimierungsphasen.

3.1 Konfiguration

Die folgenden Schritte beziehen sich auf `Projekt/src/frontend` als Wurzelverzeichnis.

Zunächst muss der Pfad angepasst werden, unter dem die Anwendung später zu erreichen sein wird. Soll sie zum Beispiel unter der URL `my.domain.de/thw` erreichbar sein, so lautet der Pfad `/thw`. In den meisten Fällen wird dies jedoch schlicht `/` sein. Die Option hierfür befindet sich in

`config/index.js` und lautet `assetsPublicPath` im Unterpunkt `build` und sollte sich im Bereich um Zeile 53 befinden.

Als zweites muss der Anwendung der Name des Servers mitgeteilt werden, auf dem Printservice und QuitStore laufen. Die Option hierfür befindet sich in den Dateien `PrintServiceAdapter.js` sowie `QuitStoreAdapter.js` im Unterordner `src/api` und heißt in beiden Fällen `url`.

Zum Schluss befinden sich im Ordner `src/config` noch einige Konfigurationsdateien, welche das Verhalten der Anwendung und damit das Nutzererlebnis beeinflussen. Diese können nach Belieben angepasst werden, sind aber bereits mit sinnvollen Standardwerten belegt.

3.2 Kompilieren

Nachdem die Konfiguration abgeschlossen ist, müssen noch die Abhängigkeiten des Quelltextes heruntergeladen werden. Dies vermeidet lästige Downloads von externen Webseiten beim Aufruf der Anwendung, welche das Laden stark verzögern oder sogar unmöglich machen können.

```
cd Projekt/src/frontend  
npm install
```

Dieser Vorgang dauert je nach Internetverbindung mehrere Minuten, da etwa 300MB an Paketen heruntergeladen werden müssen. Anschließend kann der Quelltext kompiliert werden.

```
npm run build
```

Die fertigen Dateien befinden sich nun im Ordner `Projekt/src/frontend/dist`.

3.3 Bereitstellen des Quelltexts

Wurde als Webserver Apache gewählt, so genügt es, den **Inhalt** des Ordners `dist`, also die Datei `index.html` sowie den Ordner `static` zu kopieren. Diese landen nun beispielsweise direkt im Wurzelverzeichnis von Apache, unter Debian und Derivaten also in `/var/www/html`.

Wurde kein Webserver installiert, so muss die Datei `server.js` auf den Server ausgeführt werden. Zu beachten ist dann aber, dass die Anwendung nicht unter Port 80 erreicht werden kann, sondern stets Port 8080 in die Adressleiste des Browsers mit eingegeben werden muss.