

# Benutzerdokumentation Vokabeln5 Backend

## Einleitung

Die Software stellt das Backend für den Vokabeln5-Trainer bereit. Über HTTP-Anfragen können Nutzer registriert werden, Vokabeln verwaltet und gelernt werden. Weiterhin kann über eine HTML-basierte Seite die Anfragendokumentation eingesehen werden und neue Nutzer registriert werden.

## Installation

### Voraussetzungen:

Die Installation des Vokabeln5-Backend erfordert ein Java JDK ab Version 8 und eine Installation von Gradle.

Gegebenenfalls muss die Datei „gradle.properties“ angepasst werden, um Gradle den Pfad zum installierten JDK mitzuteilen.

Die Installation erfordert weiterhin einen PostgreSQL-Server, dieser kann in der Konfigurationsdatei festgelegt werden.

Der PostgreSQL-Server benötigt mehrere Tabellen:

- Eine Tabelle für Benutzer (Standard: „users“, siehe Abschnitt Konfiguration)
- Eine Tabelle für Vokabeln (Standard: „voc“, siehe Abschnitt Konfiguration)
- Eine Tabelle für den Brute-Force-Schutz (Diese muss „fehlerhafteanmeldung“ heißen)

Diese Tabellen können zum Beispiel mit folgenden SQL-Statements erstellt werden:

Für die **Benutzer-Tabelle** „users“:

```
-- DROP TABLE public.users;

CREATE TABLE public.users
(
    id serial,
    email character varying(256) COLLATE pg_catalog."default" NOT NULL,
    password character varying(256) COLLATE pg_catalog."default" NOT
NULL,
    CONSTRAINT users_pkey PRIMARY KEY (email),
    CONSTRAINT test UNIQUE (id)
)WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.users
    OWNER to postgres;
```

Für die **Vokabel-Tabelle** “voc”:

```
-- Table: public.voc

CREATE TABLE public.voc
(
    user_id integer,
    id serial,
    frage character varying(256) COLLATE pg_catalog."default" NOT NULL,
    antwort character varying(256) COLLATE pg_catalog."default" NOT
NULL,
    sprache character varying(256) COLLATE pg_catalog."default" NOT
NULL,
    phase integer,
    gruppe character varying(256) COLLATE pg_catalog."default" NOT
NULL,
    CONSTRAINT voc_pkey PRIMARY KEY (id),
    CONSTRAINT voc_user_id_fkey FOREIGN KEY (user_id)
        REFERENCES public.users (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT voc_phase_check CHECK (phase < 6)
)

TABLESPACE pg_default;

ALTER TABLE public.voc
    OWNER to postgres;
```

Für den **Brute-Force-Schutz** mit Tabelle “fehlerhafteanmeldung”:

```
-- Table: public.fehlerhafteanmeldung

CREATE TABLE public.fehlerhafteanmeldung
(
    ip inet NOT NULL,
    zeitpunkt timestamp without time zone NOT NULL,
    gesperrtbis timestamp without time zone
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.fehlerhafteanmeldung
    OWNER to postgres;
```

### Konfiguration:

Das Programm kann mit der Datei voc6.cfg konfiguriert werden.

Nach dem Klonen aus dem git-Repository enthält die Datei bereits folgende Beispiel-Konfiguration:

EINSTELLUNG	STANDARDWERT	ERKLÄRUNG
"VOC_PORT":	5433	Port, auf dem das Backend auf HTTP-Anfragen hört
"DB_IP":	"localhost"	IP-Adresse des PostgreSQL-Servers
"DB_PORT":	5432	Port auf den der PostgreSQL-Server hört
"DB_NAME":	""	Name der Datenbank
"T_USERS":	"users"	Name der Benutzer-Tabelle in der Datenbank
"T_VOC":	"voc"	Name der Vokabel-Tabelle in der Datenbank
"T_IP"	"fehlerhafteanmeldung"	Name der Tabelle, in der die IPs stehen, die fehlerhafte Anmeldungen machen.
"DB_USER_NAME":	"postgres"	Benutzername mit dem sich das Programm mit dem PostgreSQL-Server verbindet
"DB_USER_PASSWORT":	"postgres"	Passwort mit dem sich das Programm mit dem PostgreSQL-Server verbindet
"SECRET_KEY"	"MY_SECRET_KEY!"	Passwort für die Generierung von JWTs

### Starten:

Die Installation kann auf mehrere Arten erfolgen: Nach Klonen des git-Repository kann der Quellcode in eine Java-IDE (IntelliJ Idea, Netbeans) importiert werden und von dort kompiliert und ausgeführt werden.

Alternativ kann im Unterordner Vokabeln5Backend01 in der Konsole das Kommando „gradlew.bat run“ (für Windows) bzw. „./gradlew run“ (für Unix) ausgeführt werden, um das Programm zu kompilieren und auszuführen.

Soll eine Konfiguration aus einer Datei gelesen werden, so wird das Programm mit „gradlew.bat run --args='-c voc6.cfg'“ (für Windows) bzw. „./gradlew run --args='-c voc6.cfg'“ (für Unix) gestartet. Natürlich kann man beliebige Konfigurationen, die in der Datei vorhanden sind, in der Konsole manuell eingeben. Diese Konfigurationen sind von höherer Priorität, als die in der Datei, und werden dann übernommen. Für Hilfe bei den Command-line Argumenten kann man die Applikation mit dem Argument „-h“ oder „--help“ ausführen.