

# Go-Backend-API mit Postgres-Datenbank und Docker-Compose zur Notendurchschnittsberechnung

Abdelrahman Shahin

31.03.2023

# 1 Einführung und Motivation

Im Rahmen der Veranstaltung "Weiterführende Programmiersprachen" habe ich mich entschieden, ein Projekt mit der Programmiersprache Go zu realisieren. Ziel des Projekts ist es, das Erlernen der Sprache durch die praktische Umsetzung eines realen Anwendungsfalls zu vertiefen.

Als Anwendungsfall habe ich mich für die Berechnung von Notendurchschnitten entschieden. Hierbei will ich eine Go-basierte Backend-API entwickeln, die eine Postgres-Datenbank nutzt, um Noten zu speichern und auszuwerten. Durch den Einsatz von Docker-Compose will ich zudem eine leicht deploybare und skalierbare Infrastruktur schaffen.

Mit diesem Projekt möchte ich nicht nur meine Kenntnisse in Go und Backend-Entwicklung erweitern, sondern auch Erfahrungen mit modernen Entwicklungsprozessen wie Continuous Integration und Deployment sammeln.

## 2 Technische Grundlagen

### 2.1 Go

Go ist eine Open-Source-Programmiersprache, die von Google entwickelt wurde. Sie zeichnet sich durch ihre Effizienz, einfache Syntax und integrierte Unterstützung für Multithreading aus. Um Go auf Ihrem Computer zu installieren, gehen Sie wie folgt vor: Gehen Sie auf die offizielle Go-Website: <https://go.dev/doc/install>. Laden Sie die aktuelle Version von Go für Ihr Betriebssystem herunter. Öffnen Sie die heruntergeladene MSI-Datei und folgen Sie den Anweisungen zur Installation von Go. Standardmäßig wird Go in den Ordner "Program Files" oder "Program Files (x86)" installiert. Sie können den Installationspfad bei Bedarf ändern. Nach der Installation müssen Sie alle offenen Befehlsfenster schließen und erneut öffnen, damit Änderungen an der Umgebung, die durch den Installer vorgenommen wurden, am Befehlsfenster reflektiert werden. Um sicherzustellen, dass Sie Go erfolgreich installiert haben, können Sie die folgenden Schritte ausführen: Klicken Sie im Startmenü von Windows. Geben Sie in das Suchfeld "cmd" ein und drücken Sie die Eingabetaste. Geben Sie im Befehlsfenster, das angezeigt wird, den folgenden Befehl ein: `go version`. Wenn der Befehl die installierte Version von Go ausgibt, haben Sie Go erfolgreich installiert. Nach der Installation von Go können Sie die Go-Entwicklungsumgebung aufsetzen, um mit der Entwicklung von Go-Programmen zu beginnen. Dazu können Sie Visual Studio Code mit der Go-Erweiterung verwenden.

## 2.2 Postgres-Datenbank

Um PostgreSQL auf einem Windows-System zu installieren, müssen die folgenden Schritte ausgeführt werden:

1. Unter der offiziellen PostgreSQL-Website <https://www.postgresql.org/download/windows/> müssen den Anweisungen für Windows befolgt werden.
2. Installationsprogramm starten und den Anweisungen auf dem Bildschirm folgen, um die Installation abzuschließen.
3. Die Option pgAdmin4 auswählen, um pgAdmin4 für die Verwaltung der Datenbank zu installieren.

Nachdem PostgreSQL erfolgreich installiert wurde, kann ein Datenbankserver auf eingerichtet werden. Hierfür wurde pgAdmin4 benutzt:

1. In der linken Navigationsleiste wird auf „Servers/PostgreSQL/Databases“ geklickt.
2. Durch einen Rechtsklick auf „Databases“ wird eine neue Datenbank erstellt oder im Terminal wird der Befehl „CREATE DATABASE <datenbankname>“ ausgeführt. Standardmäßig wird eine Datenbank namens "postgres" erstellt.
3. Standardmäßig läuft der Server auf Localhost auf Port 5432.

## 2.3 Docker-Compose

Laden Sie die aktuelle Version von Docker für Windows von der offiziellen Website herunter: <https://www.docker.com/products/docker-desktop> Führen Sie die heruntergeladene Installationsdatei aus, um Docker Desktop auf Ihrem Computer zu installieren. Während der Installation müssen Sie möglicherweise Ihre Anmeldinformationen eingeben, um administrative Berechtigungen zu erhalten. Sobald die Installation abgeschlossen ist, starten Sie Docker Desktop, indem Sie auf das Docker-Symbol in der Windows-Taskleiste doppelklicken. Bestätigen Sie, dass Sie die Konfiguration ändern möchten, wenn Sie dazu aufgefordert werden, und legen Sie die Ressourcen für Docker Desktop fest. Wenn Docker Desktop erfolgreich gestartet wurde, öffnen Sie eine neue Befehlszeile und führen Sie den folgenden Befehl aus, um sicherzustellen, dass Docker ordnungsgemäß installiert wurde: **docker -version** Docker Desktop enthält bereits Docker Compose, daher müssen Sie nichts weiter tun, um Compose zu installieren. Sie können jedoch den folgenden Befehl ausführen, um sicherzustellen, dass Docker Compose ordnungsgemäß installiert wurde: **docker-compose -version** Wenn Docker Compose erfolgreich installiert wurde, wird die Version von Compose angezeigt.

## 3 Architektur

### 3.1 Überblick

Die Architektur des vorliegenden Projekts besteht aus einer Go-Backend-API und einer Postgres-Datenbank, die in Docker-Containern ausgeführt werden. Die API stellt Endpunkte zur Verfügung, um Daten von der Datenbank abzurufen und zu verarbeiten. Diese Architektur ermöglicht eine hohe Skalierbarkeit und Portabilität, da die Anwendung in einem einzigen Docker-Compose-Cluster bereitgestellt werden kann. Die API wird mit Go entwickelt, da Go eine performante und einfach zu erlernende Sprache ist, die ideal für den Aufbau von Webanwendungen geeignet ist. Die Postgres-Datenbank wird verwendet, da sie eine robuste und skalierbare Lösung für die Speicherung von Daten bietet. Docker wird verwendet, um die Anwendung zu containerisieren und damit die Entwicklung und Bereitstellung zu vereinfachen. Durch die Verwendung von Docker-Compose kann die gesamte Anwendung mit nur wenigen Befehlen ausgeführt und getestet werden. Docker-Compose ermöglicht auch die einfache Skalierung der Anwendung, indem zusätzliche Instanzen der API hinzugefügt werden können, um die Last zu bewältigen.

### 3.2 Datenbank-Modell

Das Datenbankmodell der Anwendung besteht aus einer einzigen Tabelle namens "Noten". Diese Tabelle enthält eine Spalte für die ID der Note, eine Spalte für die zugehörige ID des Studenten, eine Spalte für die zugehörige ID des Kurses sowie eine Spalte für die Punktzahl der Note. Die Tabellenspalten wurden so entworfen, dass sie die Anforderungen der Anwendung erfüllen, nämlich die Speicherung von Notendaten für Studenten und Kurse sowie die Möglichkeit, den Durchschnitt der Noten eines bestimmten Kurses oder eines bestimmten Studenten zu berechnen.

### 3.3 API-Endpunkte

Die API stellt mehrere Endpunkte bereit, um die Daten in der Postgres-Datenbank zu verwalten und zu verarbeiten. Die wichtigsten Endpunkte sind: GET /users - Ruft eine Liste aller Benutzer ab POST /users - Erstellt einen neuen Benutzer GET /users/id - Ruft einen bestimmten Benutzer mit der angegebenen ID ab PUT /users/id - Aktualisiert einen bestimmten Benutzer mit der angegebenen ID DELETE /users/id - Löscht einen bestimmten Benutzer mit der angegebenen ID

GET /courses - Ruft eine Liste aller Kurse ab POST /courses - Erstellt einen neuen Kurs GET /courses/id - Ruft einen bestimmten Kurs mit der angegebenen

ID ab PUT /courses/id - Aktualisiert einen bestimmten Kurs mit der angegebenen ID DELETE /courses/id - Löscht einen bestimmten Kurs mit der angegebenen ID

GET /notes - Ruft eine Liste aller Noten ab POST /notes - Erstellt eine neue Note GET /notes/id - Ruft eine bestimmte Note mit der angegebenen ID ab PUT /notes/id - Aktualisiert eine bestimmte Note mit der angegebenen ID DELETE /notes/id - Löscht eine bestimmte Note mit der angegebenen ID

GET /users/id/courses - Ruft eine Liste aller Kurse eines bestimmten Benutzers mit der angegebenen ID ab GET /courses/id/notes - Ruft eine Liste aller Noten eines bestimmten Kurses mit der angegebenen ID ab Die HTTP-Methoden, die für jeden Endpunkt verwendet werden, sind GET, POST, PUT und DELETE. Diese Endpunkte ermöglichen es den Benutzern, Benutzer, Kurse und Noten zu erstellen, zu aktualisieren, abzurufen und zu löschen und alle Kurse und Noten eines bestimmten Benutzers oder Kurses abzurufen.

## **4 Implementierung**

### **4.1 API-Implementierung**

In diesem Abschnitt

### **4.2 Datenbank-Integration**

## **5 Deployment**

### **5.1 Docker-Compose-Datei**

### **5.2 Deployment in der Cloud**

## **6 Fazit**

### **6.1 Zusammenfassung**

### **6.2 Ausblick**