Projektdokumentation

ChatOne

Nina Krenzer
Ida Dörffel
Laura Schröppel
Melissa Baur

Link zum GitHub-Repository: https://github.com/ninakrenzer/ChatOne

Vorstellung

In diesem Projekt haben wir eine Chat-Anwendung programmiert, die mit einer Redis Datenbank verknüpft ist. Nach dem Starten der Anwendung kommt man auf einen „Login-Bereich“. Dort kann man sich mit einem gewünschten Namen anmelden. Im Anschluss daran, wird man zur eigentlichen Chat-Anwendung weitergeleitet und hat nun die Möglichkeit mit anderen Nutzern zu chatten.

Auswahl der Datenbank / Vorteile der Datenbank

Bei der Datenbank haben wir uns für die No-SQL-Datenbank Redis entschieden. Hierbei spielten insbesondere die Leistung, die Echtzeit-Verarbeitung, die Skalierbarkeit, die geringe Latenzzeit und die Unterstützung vieler Datentypen eine Rolle.

Da Redis eine In-Memory-Datenbank ist, werden die Daten im Arbeitsspeicher abgespeichert was zu einer hohen Leistung durch die schnelle Lese- und Schreibgeschwindigkeit führt. Da dies für die Chat-Anwendung viel gebraucht wird, stellte dieser Punkt eine zentrale Rolle bei der Auswahl dar. Auch die mögliche Echtzeit-Verarbeitung von Daten war ein Grund uns für Redis zu entscheiden, da es beim Chatten mit anderen Personen wichtig ist, Änderungen der Daten sofort zu erhalten, um keine Missverständnisse durch Zeitverzögerungen entstehen zu lassen. Auch die geringe Latenzzeit trägt dazu bei, die Antwortzeiten möglichst kurz zu halten. Ein weiterer Vorteil von Redis ist die gute Skalierbarkeit. Dadurch, dass es horizontal skalierbar ist und Cluster- und Replikationsfunktionen unterstützt, kann es ideal auf mehrere Server verteilt werden, was somit auch für eine hohe Verfügbarkeit verantwortlich ist. Da Redis zusätzlich viele verschiedene Datentypen speichern kann, ist es ideal für die Speicherung von Chat-Verlaufsdaten und Benutzerinformationen.

All die genannten Punkte sind Vorteile der No-SQL Datenbank Redis im Gegensatz zu SQL-Datenbanken.

Aufbau und Funktion des Projekts

Für die Chatanwendung haben wir Socket.io und einen Redis-Server verwendet, der in einem Docker-Container läuft. Redis speichert die Nachrichten und Socket.io ist die Verbindung zwischen Server und Client. Der Server läuft auf dem Port 5000 und der Redis-Container auf Port 6379.

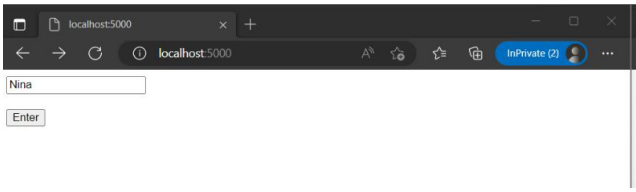
Im Folgenden wird die Programmcode erklärt:

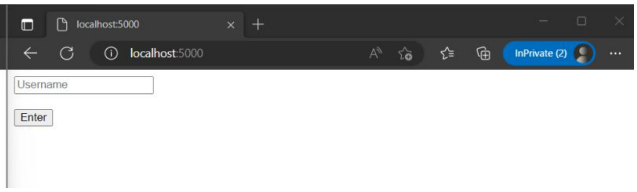
Die index.ejs ist die erste Seite, die einem angezeigt wird, wenn man die Anwendung startet. Hier trägt man den gewünschten Benutzernamen ein, den man im Chat verwenden möchte.

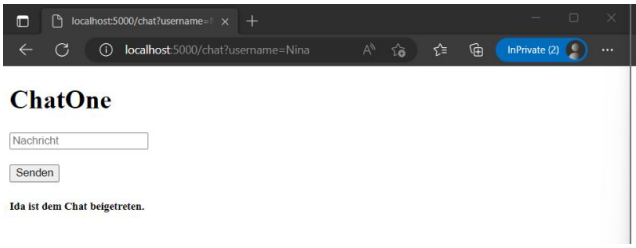
In der Datei chat.ejs haben wir ein Eingabefeld, den Sende-Button und zwei Container, um die Nachrichten und neu beigetretene Nutzer des Chats anzuzeigen. Im unteren Teil der Datei wird die Verbindung zum Socket.io-Server hergestellt. Die Funktion „emitData“ ist für das Senden der Nachricht zuständig und leitet diese über den Socket an den Server weiter. Der Socket-Listener „message“ empfängt die eingehenden Nachrichten vom Server und stellt sie auf dar. Socket-Listener „joined“ ist dagegen für zuständig zu benachrichtigen, wenn neue Nutzer dem Chat beitreten.

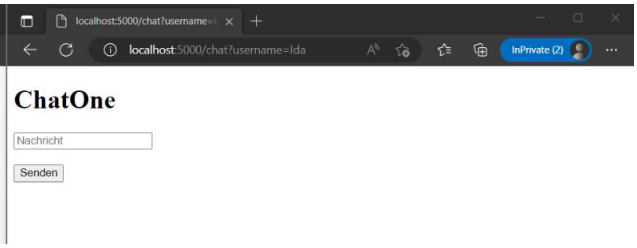
In der index.js haben wir einen Node.js-Server der Express verwendet. Hier haben wir die Verbindung zur Redis-Datenbank. Die Funktion „sendMessage“ ruft die Nachrichten von Redis ab und sendet sie über den Socket an den Client. Der Socket-Listener „connection“ behandelt neue Socket-Verbindungen. Wenn also ein Client verbunden ist werden die Nachrichten über die schon oben erklärte Funktion „sendMessage“ an den Client gesendet. Der Socket-Listener „message“ nimmt die Nachrichten vom Client auf, speichert sie in der Datenbank und leitet sie über den Socket an alle Clients weiter.

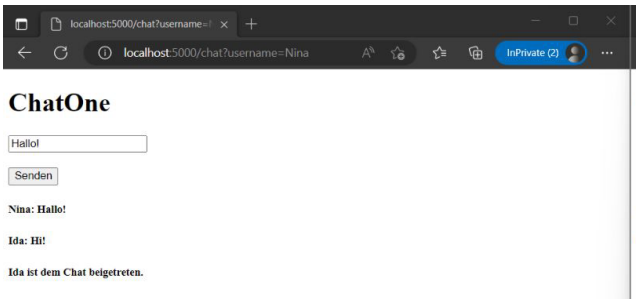
Anlage:

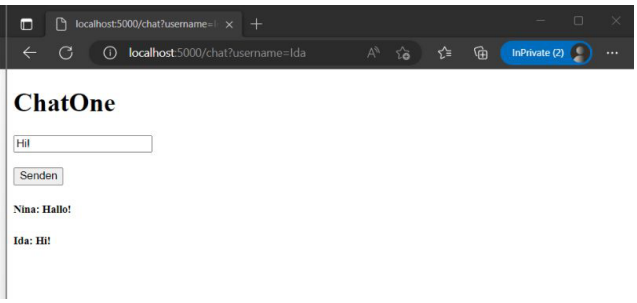

localhost:5000
Nina
Enter


localhost:5000
Username
Enter


localhost:5000/chat?username=Nina
ChatOne
Nachricht
Senden
Ida ist dem Chat beigetreten.


localhost:5000/chat?username=Ida
ChatOne
Nachricht
Senden


localhost:5000/chat?username=Nina
ChatOne
Hallo!
Senden
Nina: Hallo!
Ida: Hi!
Ida ist dem Chat beigetreten.


localhost:5000/chat?username=Ida
ChatOne
Hi!
Senden
Nina: Hallo!
Ida: Hi!