Übung

Aufgabe 0: Installation MongoDB

Installiere MongoDB auf deinem Rechner.
Zur Überprüfung ob du bereits MongoDB installiert hast kannst du mit

mongod --version

Die aktuelle Version ausgeben
db version v3.0.7
git version: 6ce7cbe8c6b899552dadd907604559806aa2e9bd

Starte daraufhin deinen MongoDB Server auf Port 27017 und überprüfe ob du dich mit dem Build-In Mongo-Client darauf konnektieren kannst.

Aufgabe 1: Connect mit Node

Lade dir mithilfe des npm-Paketmanager das Modul mongoDB herunter:

```
npm install mongodb --save
```

Lege nun eine neue JavaScript Datei *mongo.js* an in der Du eine Verbindung mithilfe des Moduls zu deinem MongoDB Servers aufbauen kannst.

Aufgabe 2: Find Documents

Erstelle mithilfe des Build-In Clients die Collection Students.
Lege folgende Documents für die Collection an:

[
    {name: “Martina”, age: “25”, status: “extern”, semesters: 8},
    {name: “Daniel”, age: “11”, status: “intern”, semesters: 2},
    {name: “Uta”, age: “13”, status: “extern”, semesters: 7},
    {name: “Chris”, age: “28”, status: “extern”, semesters: 11},
    {name: “Pascal”, age: “18”, status: “inter”, semesters: 3}
]

Implementiere nun mithilfe nodeJS und dem mongodb-Modul folgende Abfragen:
(Lasse dir das Ergebnis mit einem `console.log(result)` ausgeben

1. Zeige alle Datensätze der Collection *Students*
2. Zeige alle Datensätze der Collection *Students* deren status == extern ist
3. Zeige alle Datensätze der Collection *Students* deren Alter > 20 ist
4. Erstelle eine Abfrage welches folgendes Ergebnis liefert:
   {name: “Daniel”, age: “11”}

Aufgabe 3: Insert Document

Implementiere nun mithilfe nodeJS und dem mongodb-Modul folgende Insertions:

1. Lege folgenden Datensatz in die Collection *Student* an:

{name: “Jascha”, age: “12”, status: “intern”, semesters: 11},

2. Lege folgende Datensätze auf einmal die Collection *Student* an:

[
    {name: “Michael”, age: “25”, status: “intern”, semesters: 1},
    {name: “Patrick”, age: “11”, status: “extern”, semesters: 2},
    {name: “Michele”, age: “27”, status: “extern”, semesters: 7}
]

Überprüfe mithilfe dem Build-In Client ob die Datensätze erfolgreich angelegt wurden.

Aufgabe 3: Update Document

Update folgende Datensätze (als Selektor soll hierbei die _id dienen) mithilfe nodeJS und dem mongodb-Modul:

{name: “Michael”, age: “25”, status: “intern”, semesters: **1**}
-> {name: “Michael”, age: “25”, status: “intern”, semesters: **2**}
{name: “Patrick”, age: “**11**”, status: “extern”, semesters: 2}
-> {name: “Patrick”, age: “**13**”, status: “extern”, semesters: 2}

Aufgabe 4: DeleteDocument
Lösche folgende Datensätze (als Selektor soll hierbei der name dienen) mithilfe nodeJS und dem mongodb-Modul:

{name: “Uta”, age: “13”, status: “extern”, semesters: 7},
{name: “Michele”, age: “27”, status: “extern”, semesters: 7}

Aufgabe 5: Aggregation
Implementiere nun mithilfe nodeJS und dem mongodb-Modul folgende Aggregationen:

1. Errechne das durchschnittliche Alter (*age)* für Studenten des *status: “intern”* und des *status: “extern”*-Gruppen
2. Summiere die Semesteranzahl für die Studenten mit dem Status intern und dem Status extern