Vorbereitungsauftrag - Datenbanksysteme

|  |  |
| --- | --- |
| Datum | 26.01.2024 |
| NDK | Web Engineer NDK HF |
| Modul | Web-Technologien Aufbau |
| Thema | MongoDB |
| Plattform | Selbststudium |

**Buch Pro MERN Stack / Kapitel 6: Reading from MongoDB**

|  |  |
| --- | --- |
| **Aufgabe** | **Lösung** |
| 1. We are saving the connection in a global variable. What happens when the connection is lost? Stop the MongoDB server and start it again to see what happens. Does the connection still work? | Eine globale Variable zu verwenden ist Usanz. Das connection-Objekt ist ein connection-Pool, welches automatisch den besten Weg wählt:   * eine bestehende TCP-Verbindung wiederverwenden * eine neue Verbindung aufsetzen bei einem Absturz * etc. |
| 1. Shut down the MongoDB server, wait for a minute or more, and then start the server again. Now, refresh the browser. What happens? Can you explain this? What if you wanted a longer period for the connection to work even if the database server is down? Hint: Look up the connection settings parameters at [http://mongodb.github.io/node-mongodb-native/3.1/reference/connecting/connection-settings/](Http://mongodb.github.io/node-mongodb-native/3.1/reference/connecting/connection-settings/). | Wenn die Datenbank nur für eine kurzen Zeitraum (weniger als 30 Sekunden) nicht verfügbar ist, dann verbindet sich der Driver wieder, sobald die Datenbank wieder verfügbar ist.  Wenn die Datenbank länger nicht verfügbar ist, wird eine Fehlermeldung zurückgegeben. In diesem Fall ist ein Neustart des Servers notwendig.  Bei den 30 Sekunden handelt es sich um einen Default-Intervall, welcher geändert werden kann (connection settings: *reconnectTries* oder *reconnectInterval*). |
| 1. We used *toArray()* to convert the list of issues into an array. What if the list is too big, say, a million documents? How would you deal with this? Hint: Look up the documentation for the MongoDB Node.js driver’s *Cursor* at <http://mongodb.github.io/node-mongodb-native/3.1/api/Cursor.html>. Note that the *find()* method returns a *Cursor*. | Eine Möglichkeit wäre, das Ergebnis einzuschränken mit *limit()*. Oder mit *skip()* könnte festgelegt werden, ab wann die Liste gestartet werden soll.  Sofern der Client mit grossen Listen umgehen kann, aber nicht viel Speicherplatz im Server verschwendet werden soll, könnte *hasNext()* und *next()* verwendet werden, um die Ergebnisse zu streamen. |