Cheatsheet 165

# Shell

## Lokal mongo starten

Mongosh

## Origin mongo starten

mongosh "mongodb+srv://cluster0.ostytiv.mongodb.net/" --apiVersion 1 --username dbAdmin

# Backup

## Import

mongorestore --uri="<URI>" --db=<DBNAME> --username=<USER> <INPUT>

mongorestore --db=formum ./forum

## Export

mongodump --uri="<URI>" --db=<DBNAME> --username=<USER> --out=<OUTPUT>

mongodump --db=restaurant --out=restaurant-export-local

# Read

## Liste alle Restaurants mit italienischer Küche (**Italian**) auf.

db.restaurants.find({"cuisine": "Italian"})

## Finde das Restaurant mit dem Namen 7B Bar.

db.restaurants.find({"name": "7B Bar"})

## Finde alle Restaurants, welche den Begriff Steakhouse im Namen tragen.

db.restaurants.find({ "name": { $regex: /Steakhouse/ } })

## Finde alle Restaurants, welche die Postleitzahl 11219 besitzen.

db.restaurants.find({ "address.zipcode": "11219" })

## Es gibt ein Restaurant, welches in einer Bewertung mehr als 100 Punkte (score) erhalten hat. Mit welchem Befehl kann es gefunden werden?

db.restaurants.find({ "grades.score": { $gt : 100 }})

## Es gibt ein Restaurant, welches in einer Bewertung mehr als 100 Punkte (score) erhalten hat. Mit welchem Befehl kann es gefunden werden?

db.restaurants.find({$and:[{"cuisine": "Irish"},{"borough": "Queens"}]})

## Finde alle Restaurants in Brooklyn, welche entweder Backwaren (Bakery) oder Burgers (Hamburgers) anbieten.

db.restaurants.find({$and:[{$or:[{"cuisine": "Hamburgers"}, {"cuisine": "Bakery"}]},{"borough": "Brooklyn"}]})

## Es sollen ausschliesslich die Namen aller Restaurants ausgegeben werden.

db.restaurants.find({},{"name": true,"\_id": false})

## Finde alle Restaurants, welche Seafood anbieten. Es soll der Name und die Adresse ausgegeben werden.

db.restaurants.find({"cuisine":"Seafood"},{"name": true,"address.street":true,"\_id": false})

## Die Anzahl an Restaurants in Manhattan soll ermittelt werden.

db.restaurants.countDocuments({"borough": "Manhattan"})

## Finde alle Restaurants, welche Seafood anbieten und keine Bewertung unter 5 haben.

db.restaurants.find({$and:[{"cuisine":"Seafood"}, {"grades.score":{$not:{$lte:5}}}]})

## Stelle die Restaurants aus der vorherigen Aufgabe mit Name und Adresse dar. Alle anderen Informationen sollen nicht angezeigt werden.

db.restaurants.find({$and:[{"cuisine":"Seafood"}, {"grades.score":{$not:{$lte:5}}}]},{"name":true,"address":true,"\_id":false})

# Update

## Das Restaurant mit dem Namen Bully’S Deli hat den Besitzer gewechselt und soll neu Herbies Snacks heissen.

db.restaurants.updateOne({"name": "Bully'S Deli"}, { $set: { "name": "Herbies Snacks"} } )

## Das Restaurant bietet unter neuer Führung Fastfood an. Passe die Küche an.

db.restaurants.updateOne({"name": "Herbies Snacks"}, { $set: { "cuisine": "Fastfood"} } )

## Der neue Besitzer hat eine Chance auf einen Neuanfang: Ersetze die bisherigen Bewertungen durch ein leeres Array.

db.restaurants.updateOne({"name": "Herbies Snacks"}, { $set: { "grades": []} } )

## Füge für jedes Restaurant ein neues Feld für den Ländercode ****countrycode**** ein. Da sich bisher alle Restaurants in den USA befinden, wird der Ländercode ****US**** eingetragen

db.restaurants.updateMany({}, { $set: { "countrycode": "US"} } )

## Manche Restaurants möchten ihre Öffnungszeiten anzeigen. Dazu sollen in ein Array **hours** die Öffnungszeiten für eine Woche eingetragen werden, beginnend bei Montag. Die Öffnungszeiten werden als Objekt mit den Feldern **from** und **to** als String eingetragen. Ergänze diese Öffnungszeiten für das Restaurant **Glorious Food**:

|  |  |  |
| --- | --- | --- |
| **Tag** | **From** | **To** |
| Mo | 07:00 | 22:00 |
| Di | 12:00 | 22:00 |
| Mi | 07:00 | 22:00 |
| Do | 07:00 | 22:00 |
| Fr | 07:00 | 23:30 |
| Sa | 10:00 | 23:30 |
| So | 14:00 | 21:00 |

db.restaurants.updateOne({"name":"Glorious Food"},{$set:{

"hours":{

"Mo":{ "From":"07:00", "To":"22:00" },

"Di":{ "From":"12:00", "To":"22:00" },

"Mi":{ "From":"07:00", "To":"22:00" },

"Do":{ "From":"07:00", "To":"22:00" },

"Fr":{ "From":"07:00", "To":"23:30" },

"Sa":{ "From":"10:00", "To":"23:30" },

"So":{ "From":"14:00", "To":"21:00" }

}}})

## Zusätzliche Informationen sollen ein noch besseres Bild des Restaurants geben. In einem neuen Objekt namens **misc** sollen die zusätzlichen Informationen gespeichert werden. Ergänze diese Informationen für das Restaurant **Glorious Food**:

| **Bezeichnung** | **Beschreibung** | **Datentyp** | **Wert** |
| --- | --- | --- | --- |
| hasWifi | WLAN-Zugang | Boolean | true |
| parking | Anzahl Parkplätze | Integer | 15 |
| webpage | Webseite | String | gloriousfood.com |
| phone | Telefonnummer | String | +1 123-456-7890 |

db.restaurants.find({$set:{ "misc":{ "hasWifi":true, "parking": 15, "webpage":"gloriousfood.com", "phone":"+1 123-456-7890" }

## Im Objekt **misc** wird der Zeitstempel der Eröffnung ergänzt. Im Feld **opening** wird für das Restaurant **Glorious Food** das Datum 1. August 2022, welches um 7:00 Uhr ortszeit geöffnet hat, eingetragen. Wichtig: Beachte die Zeitzone.

db.restaurants.updateOne({"name":"Glorious Food"},

{$set:{"misc.opening": new Date("2022-08-01T07:00:00-0400")}})