

NoSQL



FERIT

Osnove MongoDB-a

Laboratorijska vježba 3

Postavljanje okruženja

1. Kreirajte direktorij na vašem računalu za ovu vježbu
2. Unutar tog direktorija, ubacite datoteku pod nazivom docker-compose.yml (s Merlina).
3. Otvorite direktorij u Visual Studio Code okruženju
4. Otvorite terminal, pozicionirajte se u direktorij gdje ste spremili datoteku

```
services:
  mongodb:
    image: mongo:6
    container_name: nosql-mongo-lab3
    ports:
      - "27017:27017"
    environment:
      - MONGO_INITDB_ROOT_USERNAME=root
      - MONGO_INITDB_ROOT_PASSWORD=example
    volumes:
      - ./mongo-data:/data/db
      - ./restaurants.json:/restaurants.json
  mongo-express:
    image: mongo-express
    container_name: nosql-mongo-express
    ports:
      - "8081:8081"
    environment:
      - ME_CONFIG_MONGODB_ADMINUSERNAME=root
      - ME_CONFIG_MONGODB_ADMINPASSWORD=example
      - ME_CONFIG_MONGODB_SERVER=mongodb
    depends_on:
      - mongodb
```

Postavljanje okruženja

5. Pokrenite MongoDB (bazu) i Mongo Express (web sučelje) naredbom:
`docker-compose up -d`
6. Provjerite radi li sve ispravno naredbom `docker ps`. Trebali biste vidjeti dva containera: `nosql-mongo-lab3` i `nosql-mongo-express`
7. Pripremiti dokument *restaurants.json* (s Merlina) u isti folder. Uvesti podatke iz json dokumenta u bazu „lv3” naredbom:

`docker exec -i nosql-mongo-lab3 mongoimport --username root --password example --authenticationDatabase admin --db lv3 --collection restaurants --type json --file /restaurants.json`
8. Spajanje na MongoDB pomoću mongosh:
`docker exec -it nosql-mongo-lab3 mongosh --username root --password example`
9. Koristiti bazu „lv3”: `use lv3`

Postavljanje okruženja

10. Nakon spajanja na mongosh moguće je upućivati naredbe kroz komandnu liniju

11. Spajanje na Mongo Express (web GUI):

- Otvorite web preglednik (Chrome, Firefox...) i idite na adresu: <http://localhost:8081>
- Koristite sljedeće podatke za spajanje:
 - **Username:** admin
 - **Password:** pass
- Kliknite na „View” kod baze „lv3” i potom „View” na kolekciji „Restaurants”. Ovdje možete vizualno pregledati podatke u bazi.

Zadatak 1: Osnovna navigacija i čitanje podataka

a) Navigacija:

- Prikažite sve baze podataka na serveru.
- Provjerite na kojoj se bazi trenutno nalazite.
- Prikažite sve kolekcije unutar test baze.

b) Dohvaćanje svih dokumenata:

- Napišite upit koji dohvaća **sve** restorane iz kolekcije restaurants. Zbog preglednosti, dodajte .limit(5) na kraj upita da se ispiše samo prvih 5.

c) Filtriranje po točnoj vrijednosti (Equality Match):

- Pronađite sve restorane koji se nalaze u gradskoj četvrti (borough) "Manhattan".
- Pronađite sve restorane koji poslužuju kuhinju (cuisine) tipa "Italian".

Zadatak 1: Osnovna navigacija i čitanje podataka

d) Filtriranje po polju unutar ugniježđenog dokumenta (Dot Notation):

- Pronađite sve restorane koji se nalaze na poštanskom broju (address.zipcode) "10075".

e) Filtriranje pomoću operatora za usporedbu:

- Pronađite sve restorane koji su u barem jednoj ocjeni (grades.score) dobili ocjenu **veću od 90**.
- Pronađite sve restorane koji su dobili ocjenu **manju ili jednaku 5**.

f) Filtriranje pomoću logičkih operatora:

- Pronađite sve restorane koji poslužuju "Italian" kuhinju **ILI** se nalaze u četvrti "Brooklyn".
- Pronađite sve restorane u četvrti "Manhattan" koji poslužuju "American" kuhinju **I** imaju barem jednu ocjenu veću od 70.

Zadatak 2: Projekcija, ažuriranje i brisanje

a) Projekcija (Projection):

- Napišite upit koji pronalazi sve restorane u četvrti "Queens", ali vraća **samo** polja name, cuisine i borough.
- Napišite isti upit, ali ovoga puta **isključite** polje _id iz rezultata.

b) Ažuriranje jednog dokumenta (updateOne):

- Pronađite restoran s imenom "Tu-Lu'S Gluten-Free Bakery".
- Pomoću updateOne i operatora \$set, promijenite mu vrijednost polja cuisine u "Bakery".
- Dodajte mu novo polje chef_name s vrijednošću "Ana Anić" koristeći \$set.
- Pronađite ga opet da provjerite je li promjena uspješna.

Zadatak 2: Projekcija, ažuriranje i brisanje

c) Ažuriranje više dokumenata (updateMany):

- Napišite upit koji **svim** restoranima koji poslužuju "American" kuhinju dodaje novo polje `inspection_due` s vrijednošću `true`.
- Provjerite na nekoliko američkih restorana je li se polje dodalo.

d) Brisanje jednog dokumenta (deleteOne):

- Restoran s `restaurant_id` "40356018" se zatvorio.
- Pronađite i obrišite taj restoran koristeći `deleteOne`.
- Pokušajte ga ponovno pronaći da potvrdite da je obrisani.

Zadatak 3: Sortiranje i paginacija

a) Sortiranje (sort):

- Prikažite prvih 10 restorana iz četvrti "Bronx", sortiranih abecedno po imenu (name) **uzlazno (A-Z)**.
- Prikažite prvih 10 restorana iz četvrti "Bronx", sortiranih abecedno po imenu **silazno (Z-A)**.

b) Limitiranje i preskakanje (limit i skip):

- Napišite upit koji prikazuje **5 najnovije ocijenjenih restorana** u cijeloj kolekciji. (Pretpostavite da grades.date polje predstavlja datum ocjenjivanja).
- Implementirajte paginaciju: Prikažite restorane sortirane po imenu, ali preskočite prvih 10 i prikazite idućih 10 (tj. restorane od 11. do 20. na listi).

Kraj vježbe:

- Nakon što ste gotovi, možete se odjaviti iz mongosh-a naredbom `exit`.
- Da biste zaustavili i obrisali Mongo Docker containere, u terminalu upišite:
`docker-compose down -v`
- Kreirajte jednu tekstualnu datoteku (`lab3_rjesenja_ime_prezime.txt`) i u nju kopirajte sve naredbe koje ste koristili za rješavanje zadataka, grupirane po broju zadatka.
- *Uploadajte* datoteku s rješenjima na Merlin.