

Laboratorium 10

Model danych w bazie danych MongoDB

Cele

- Zapoznanie się z dokumentową bazą danych MongoDB
- Przygotowanie środowiska pracy
- Opracowanie modelu danych typu dokumentowego.
- Zapisanie danych w bazie MongoDB.

Dane

- Dane pogodowe pochodzące z IMGW
- Dane astronomiczne
- Dane administracyjne

Program

Napisz program, który odczytuje dane i zapisuje je w postaci dokumentów JSON w bazie MongoDB. Zmodyfikuj program liczący statystyki tak, aby pobierał dane z MongoDB.

Wskazówki

Wskazówki dotyczące instalacji i podstawowych testów bazy MongoDB.

Instalacja

Linux	<pre>apt-get install mongodb-org yum install mongodb-org</pre>
Mac / Windows	https://www.mongodb.com/try/download/community

Instalacja z użyciem Docker Desktop

1. Instalacja Docker Desktop - <https://hub.docker.com/editions/community/docker-ce-desktop-windows>
2. Pobranie kontenera MongoDB:

```
docker pull mongo
```

3. Utworzenie woluminu na dane:

```
docker volume create --name=mongodata
```

4. Uruchomienie kontenera:

```
docker run --name mongodb -v mongodata:/data/db -d -p 27017:27017 mongo
```

5. Podłączenie do bazy z poziomu aplikacji klienckiej:

```
mongodb://localhost:27017/admin
```

6. Podłączenie do bazy z poziomu aplikacji klienckiej z podaniem nazwy użytkownika:

```
mongodb://admin:password@localhost:27017/admin
```

Wersja on-line

MongoDB Atlas (MongoDB as a service) - <https://www.mongodb.com/try>

Aplikacja kliencka

- MongoDB Compass - <https://www.mongodb.com/products/compass>
- MongoDB for VS Code – Visual Studio Code extension.

Test

Nawiązanie połączenia z bazą MongoDB z poziomu aplikacji w języku Python:

```
import pymongo
import json

connection = pymongo.MongoClient("mongodb://localhost")
db = connection.baza
bdot = db.bdot
...
connection.close
```

Przepisanie modelu BDOT10k do bazy MongoDB, np.

```
bdot.insert_one({'gmlid': 'PL.PZGIK.BDOT10k.BUCMA.04.1549', ...})
```

Przykład reprezentacji obiektu klasy OT_BUCM_A w JSON

```
{
  'klasa': 'OT_BUCM_A ',
  'gmlid': 'PL.PZGIK.BDOT10k.BUCMA.04.636',
  'idIIP': {
    'lokalnyId': '2CE79A9C-1876-6A1D-E053-CC2BA8C00F96',
    'przestrzenNazw': 'PL.PZGIK.994.BDOT10k',
    'wersjaId': '2015-02-10T00:00:00'
  },
  'czyObiektBDOO': 'false',
  'x_kod': 'BUCM01',
  'x_skrKarto': {
    'nil': true,
    'nilReason': 'inapplicable'
  },
  'x_katDoklGeom': 'dokladny',
  'x_zrodloDanychG': 'ortofotomapa',
  'x_zrodloDanychA': 'pomiarTerenowy',
  'x_katIstnienia': 'eksploatowany',
  'x_rodzajReprGeom': 'maksymalnyZasieg',
  'x_uzytkownik': 'Uzytkownik04.xml',
  'x_aktualnoscG': '2015-02-10',
  'x_aktualnoscA': '2015-02-10',
  ...
}
```

Operacje na danych w aplikacji klienckiej MongoDB

Dodanie obiektu do kolekcji torun (operacja INSERT)

```
db.torun.insertOne(
{
  'klasa': 'OT_BUCM_A ',
  'gmlid': 'PL.PZGIK.BDOT10k.BUCMA.04.636',
  'idIIP': {
    'lokalnyId': '2CE79A9C-1876-6A1D-E053-CC2BA8C00F96',
    'przestrzenNazw': 'PL.PZGIK.994.BDOT10k',
    'wersjaId': '2015-02-10T00:00:00'
  },
  'czyObiektBDOO': 'false',
  'x_kod': 'BUCM01',
  ...
})
```

Dodanie wielu obiektów do kolekcji torun (operacja INSERT)

```
db.torun.insertMany(  
  [  
    {  
      'klasa': 'OT_BUCM_A',  
      'gmlid': 'PL.PZGIK.BDOT10k.BUCMA.04.636',  
      'idIIP': { 'lokalnyId': '2CE79A9C...' },  
    },  
    {  
      'klasa': 'OT_BUCM_A',  
      'gmlid': 'PL.PZGIK.BDOT10k.BUCMA.04.637',  
      'idIIP': { 'lokalnyId': '2CE79A9C...' },  
    },  
    {  
      'klasa': 'OT_BUCM_A',  
      'gmlid': 'PL.PZGIK.BDOT10k.BUCMA.04.638',  
      'idIIP': { 'lokalnyId': '2CE79A9C...' },  
    }  
  ]  
)
```

Wyszukanie obiektów w kolekcji torun (operacja SELECT * FROM ...)

```
db.torun.find()
```

Zliczenie liczby wyników (operacja SELECT COUNT(*) FROM ...)

```
db.torun.find().count()
```

+

Wyszukanie obiektu o podanej klasie (operacja SELECT * FROM ... WHERE klasa = 'OT_BUCM_A')

```
db.torun.find({'klasa': 'OT_BUCM_A'})
```

Usunięcie obiektu o podanym gmlid

```
db.torun.deleteOne({'gmlid': 'PL.PZGIK.BDOT10k.BUCMA.04.636'})
```

Usunięcie wszystkich obiektów klasy OT_BUCM_A

```
db.torun.deleteMany({'klasa':'OT_BUCM_A'})
```

MongoDB + Python

+Przykład wyczytania obiektu o podanym gmlid

(baza danych nazywa się bdot10k, kolekcja nazywa się torun)

```
import pymongo
connection = pymongo.MongoClient("mongodb://localhost")
db = connection.bdot10k
torun = db.torun
query = {'gmlid': 'PL.PZGIK.BDOT10k.BUCMA.04.1540'}
doc = torun.find_one(query)
print(doc)
connection.close()
```

Zapisanie jednego dokumentu (INSERT)

```
db.torun.insert_one({'gmlid': , 'PL.PZGIK.BDOT10k.BUCMA.04.1540'})
```

Zapisanie wielu dokumentów (INSERT)

```
db.torun.insert_many([{'klasa': , 'OT_BUCM_A'} for i in range(2)])
```

Pobranie jednego dokumentu (SELECT)

```
db.torun.find_one({'gmlid': , 'PL.PZGIK.BDOT10k.BUCMA.04.1540'})
```

Pobranie wielu dokumentów (SELECT)

```
db.torun.find({'klasa': , 'OT_BUCM_A'})
```

Odnosiniki

Dokumentacja bazy danych MongoDB	https://docs.mongodb.com/manual/
Podręcznik "MongoDB Documentation Release 3.0.4"	http://web.cs.wpi.edu/~cs585/s17/Books/Books-PDF/MongoDB-manual.pdf
Docker overview	https://docs.docker.com/get-started/overview/
Docker Desktop	https://www.docker.com/products/docker-desktop
Docker Docs	https://docs.docker.com/
Docker 101 Tutorial	https://www.docker.com/101-tutorial
Docker Desktop for Windows user manual	https://docs.docker.com/desktop/windows/
MongoDB on Docker Hub	https://hub.docker.com/_/mongo
Docker and MongoDB	https://www.mongodb.com/compatibility/docker
Wprowadzenie do JSON (w3schools)	https://www.w3schools.com/js/js_json_intro.asp
Portal json.org	https://www.json.org/json-en.html
JSON Formatter & Validator	https://jsonformatter.curiousconcept.com/
JSON Editor Online	https://jsoneditoronline.org/
Podręcznik "Understanding JSON Schema"	http://json-schema.org/understanding-json-schema/UnderstandingJSONSchema.pdf