

Node.js - Laboratorium 11

MongoDB (<https://www.mongodb.com/>)

Program który uruchamia serwer bazy danych z podaniem ścieżki do katalogu:

```
c:/mongodb/bin/mongod --dbpath c:/mongodb-data
```

MongoDB Node.JS Driver (<https://mongodb.github.io/node-mongodb-native/?jmp=docs>)

Reference: <http://mongodb.github.io/node-mongodb-native/3.2/>

API: <http://mongodb.github.io/node-mongodb-native/3.2/api/>

stworzenie instancji klienta MongoDB

```
const { MongoClient } = require('mongodb');

(async () => {
  const dbAddress = 'mongodb://127.0.0.1:27017/';
  const dbName = 'users';
  const mongoOptions = { useNewUrlParser : true };
  const client = await MongoClient.connect(dbAddress, mongoOptions);

  const db = client.db(dbName);
  // ...
})();
```

dodanie użytkownika do kolekcji

```
const { MongoClient } = require('mongodb');

(async () => {
  const dbAddress = 'mongodb://127.0.0.1:27017/';
  const dbName = 'users';
  const mongoOptions = { useNewUrlParser : true };
  const client = await MongoClient.connect(dbAddress, mongoOptions);

  const db = client.db(dbName);

  const result = await db.collection('users').insertOne({ name: 'jan' });
  console.log(result.ops);
})();
```

```
})();
```

zmiana użytkownika nazwy w kolekcji

```
const { MongoClient } = require('mongodb');

(async () => {
  const dbAddress = 'mongodb://127.0.0.1:27017/';
  const dbName = 'users';
  const mongoOptions = { useNewUrlParser : true };
  const client = await MongoClient.connect(dbAddress, mongoOptions);

  const db = client.db(dbName);

  const result = await db.collection('users').updateMany({ name: 'jan' }, {
    $set: { name: 'Adam' } });
  console.log(result.ops);
})();
```

pobieranie wszystkich użytkowników

```
const { MongoClient } = require('mongodb');

(async () => {
  const dbAddress = 'mongodb://127.0.0.1:27017/';
  const dbName = 'users';
  const mongoOptions = { useNewUrlParser : true };
  const client = await MongoClient.connect(dbAddress, mongoOptions);

  const db = client.db(dbName);

  const result = await db.collection('users').find().toArray();
  console.log(result);
})();
```

mongoose (<https://mongoosejs.com/>)

```
const mongoose = require('mongoose');
mongoose.connect('mongodb://localhost:27017/test', { useNewUrlParser: true });

const Cat = mongoose.model('Cat', { name: String });

const kitty = new Cat({ name: 'Zildjian' });
kitty.save().then(() => console.log('meow'));
```

Zadania do wykonania na laboratorium

1. Zainstaluj bazę danych **MongoDB** na swoim komputerze i sprawdź czy działa połączenie do niej tworząc prostą aplikację wykorzystującą pakiet **mongodb**.
2. Wykorzystując bazę danych **MongoDB**, stwórzmy aplikację typu **lista zadań(todos)**.
 - aplikacja powinna pozwolić na dodanie nowego dokumentu do naszej bazy w postaci:

```
{
  label: 'Kupić 5 jaj',
  isCompleted: false
}
```

- pozwolić na zmodyfikowanie opisu jak i zaznaczenie iż dane zadanie zostało już wykonane
- usunięcie wpisu z listy

W tym zadaniu stwórzmy do każdego polecenia oddzielne funkcje i na samym końcu naszej aplikacji wywołajmy przykładowe zastosowanie ich.

3. Dodajmy do zadania 2 implementację **REST API** pozwalające na manipulowanie danymi z poziomu żądań **HTTP**.