Отримання даних у MongoDB

Метод find

Для отримання даних із колекції застосовується метод **find()** :

```
const MongoClient = require("mongodb").MongoClient;

const url = "mongodb://127.0.0.1:27017/";
const mongoClient = new MongoClient(url);

async function run() {
    try {
        await mongoClient.connect();
        const db = mongoClient.db("usersdb");
        const collection = db.collection("users");
        const results = await collection.find().toArray();
        console.log(results);

    }catch(err) {
        console.log(err);
    } finally {
        await mongoClient.close();
    }
}
run().catch(console.error);
```

Метод find повертає спеціальний об'єкт **FindCursor** , і щоб отримати всі дані цього об'єкта викликається метод **toArray()** . І якщо ми запустимо програму, то побачимо всі раніше додані дані:

```
c:\node\mongoapp>node app
[
    {
        _id: new ObjectId("6112a80ca8a25160d53d161a"),
        name: "Tom",
```

```
        age: 23
    }, {
        _id: new ObjectId("6112a80873c40dedd6d837c"),
        name: "Bob",
        age: 35
    },
    {
        _id: new ObjectId("6112a80873c40dedd6d837d"),
        name: "Alice",
        age: 21
    },
    {
        _id: new ObjectId("6112a80873c40dedd6d837e"),
        name: "Tom",
        age: 45
    }
]

c:\node\mongoapp>
```

Фильтрация документов

С помощью метода `find()` мы можем дополнительно отфильтровать извлекаемые документы. Например, нам надо найти всех пользователей, у которых имя - Tom:

```
const MongoClient = require("mongodb").MongoClient;

const url = "mongodb://127.0.0.1:27017/";
const mongoClient = new MongoClient(url);

async function run() {
    try {
        await mongoClient.connect();
        const db = mongoClient.db("usersdb");
        const collection = db.collection("users");
        const results = await collection.find({name: "Tom"}).toArray();
        console.log(results);

    }catch(err) {
        console.log(err);
    } finally {
        await mongoClient.close();
    }
}
run().catch(console.error);
```

В качестве параметра в find передается объект, который устанавливает параметры фильтрации. В частности, что свойство name должно быть равно "Tom".

```
c:\node\mongoapp>node app
[
    {
        _id: new ObjectId("6112a80ca8a25160d53d161a"),
        name: "Tom",
        age: 23
    },
    {
        _id: new ObjectId("6112a80873c40dedd6d837e"),
        name: "Tom",
        age: 45
    }
]

c:\node\mongoapp>
```

Фильтрация по нескольким критериям

Мы можем устанавливать дополнительные критерии фильтрации, например, добавим фильтрацию по возрасту:

```
const results = await collection.find({name: "Tom", age: 28}).toArray();
console.log(results);
```

Получение одного объекта и метод findOne

Метод **findOne()** працює аналогічно, тільки дозволяє отримати один документ:

```
const MongoClient = require("mongodb").MongoClient;

const url = "mongodb://127.0.0.1:27017/";
const mongoClient = new MongoClient(url);

async function run() {
    try {
        await mongoClient.connect();
        const db = mongoClient.db("usersdb");
        const collection = db.collection("users");
        const result = await collection.findOne({name: "Tom"});
        console.log(result);

    }catch(err) {
```

```
        console.log(err);
    } finally {
        await mongoClient.close();
    }
}
run().catch(console.error);
```

Назад Зміст Вперед