
Отримання даних у MongoDB

Метод find

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

```
1  const MongoClient = require("mongodb").MongoClient;
2
3  const url = "mongodb://127.0.0.1:27017/";
4  const mongoClient = new MongoClient(url);
5
6  async function run() {
7    try {
8      await mongoClient.connect();
9      const db = mongoClient.db("usersdb");
10     const collection = db.collection("users");
11     const results = await collection.find().toArray();
12     console.log(results);
13
14   } catch(err) {
15     console.log(err);
16   } finally {
17     await mongoClient.close();
18   }
19 }
20 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:

```
1  const MongoClient = require("mongodb").MongoClient;
2
3  const url = "mongodb://127.0.0.1:27017/";
4  const mongoClient = new MongoClient(url);
5
6  async function run() {
7    try {
8      await mongoClient.connect();
9      const db = mongoClient.db("usersdb");
10     const collection = db.collection("users");
11     const results = await collection.find({name: "Tom"}).toArray();
12     console.log(results);
13
14   } catch(err) {
15     console.log(err);
16   } finally {
17     await mongoClient.close();
18   }
19 }
20 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:

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

Получение одного объекта и метод `findOne`

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

```
1 const MongoClient = require("mongodb").MongoClient;
2
3 const url = "mongodb://127.0.0.1:27017/";
4 const mongoClient = new MongoClient(url);
5
6 async function run() {
7   try {
8     await mongoClient.connect();
9     const db = mongoClient.db("usersdb");
10    const collection = db.collection("users");
11    const result = await collection.findOne({name: "Tom"});
12    console.log(result);
13
14    }catch(err) {
```

```

15     console.log(err);
16   } finally {
17     await mongoClient.close();
18   }
19 }
20 run().catch(console.error);

```

[Назад](#) [Зміст](#) [Вперед](#)



ALSO ON METANIT.COM

Компилятор GCC. Первая программа ...

2 месяца назад · 3 коммент...

Компилятор GCC. Первая программа на языке Си на Windows, набор ...

Группировка в CollectionView

месяц назад · 3 коммента...

Группировка в CollectionView в приложении на .NET ...

Измерение производительности ...

2 месяца назад · 1 коммент...

Измерение производительности приложения на языке ...

Отпра

2 меся

Отпра
Core s

0 Комментариев

1 Войти ▾

G

Начать обсуждение...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS ?

Имя



Поделиться

Лучшие

Новые

Старые

Прокомментируйте первым.

Подписаться

Privacy

Не продавайте мои данные