

---

## Додавання даних до MongoDB

У минулій темі було розглянуто деякі базові моменти роботи з MongoDB. Тепер докладніше розглянемо як додавати та отримувати дані з бд.

Щоб додати, ми можемо використовувати різні методи. Якщо потрібно додати один об'єкт, то застосовується метод **insertOne()** . При додаванні набору об'єктів можна використовувати метод **insertMany()** .

### Додавання одного документа

Для додавання одного документа застосовується метод **insertOne()** . Як параметр він приймає документ, що додається. Розглянемо з прикладу:

```
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 user = {name: "Tom", age: 28};
12     const result = await collection.insertOne(user);
13     console.log(result);
14     console.log(user);
15   } catch(err) {
16     console.log(err);
17   } finally {
18     await mongoClient.close();
19   }
20 }
21 run().catch(console.error);
```

Добавляемый документ представляет объект. В данном случае это объект

```
1 {name: "Tom", age: 23};
```

То есть в коллекцию `users` будет добавляться документ, который имеет для свойства `name` и `age`, которые хранят соответственно строку `"Tom"` и число `23`.

После добавления получаем результат операции и выводим его на консоль. Также выводим на консоль объект добавленный `user`, чтобы посмотреть какие изменения произошли с ним после добавления в бд.

Результат работы программы:

```
c:\node\mongoapp>node app
{
  acknowledged: true,
  insertedId: new ObjectId("6112a80ca8a25160d53d161a")
}
{name: "Tom", age: 23, _id: new ObjectId("6112a80ca8a25160d53d161a")}
```

Здесь мы можем увидеть результат операции, в частности, с помощью свойства `result.insertedId` получить идентификатор добавленного объекта. Кроме того, мы видим, кроме начальных свойств документ `user` после добавления в БД получили дополнительное свойство `_id` - это уникальный идентификатор документа, который присваивается сервером при добавлении.

## Добавление нескольких объектов

Теперь используем метод `insertMany()`. Добавим набор объектов и для этого изменим файл приложения `app.js`:

```
1 const MongoClient = require("mongodb").MongoClient;
2
3 const url = "mongodb://127.0.0.1:27017/";
4 const mongoClient = new MongoClient(url);
5
6 const users = [{name: "Bob", age: 35} , {name: "Alice", age: 21}, {name: "Tom", age: 23}];
7
8 async function run() {
9   try {
10     await mongoClient.connect();
11     const db = mongoClient.db("usersdb");
12     const collection = db.collection("users");
13     const results = await collection.insertMany(users);
14     console.log(results);
15     console.log(users);
16   } catch(err) {
```

```
17         console.log(err);
18     } finally {
19         await mongoClient.close();
20     }
21 }
22 run().catch(console.error);
```

Как и `insertOne`, метод `insertMany()` в качестве параметра принимает добавляемые данные - массив объектов, а в качестве результата возвращает результат операции. При удачном добавлении добавленные объекты из массива `users` аналогичным образом получают свойство `_id`.

Запустим приложение:

```
c:\node\mongoapp>node app
{
  acknowledged: true,
  insertedCount: 3,
  insertedIds: {
    '0': new ObjectId("6377874e1e1a085c14fcd12f"),
    '1': new ObjectId("6377874e1e1a085c14fcd130"),
    '2': new ObjectId("6377874e1e1a085c14fcd131")
  }
}
[
  {
    name: 'Bob',
    age: 35,
    _id: new ObjectId("6377874e1e1a085c14fcd12f")
  },
  {
    name: 'Alice',
    age: 21,
    _id: new ObjectId("6377874e1e1a085c14fcd130")
  },
  {
    name: 'Tom',
    age: 45,
    _id: new ObjectId("6377874e1e1a085c14fcd131")
  }
]
```