

Fullstack-разработка

Mongo
Введение

Цель урока:

- Установим нужные программы
- Научимся добавлять тысячи записей БД
- Составим условия поиска по БД

Теория

Mongo

Массив объектов

До сих пор мы хранили данные на сервере в массиве объектов

```
let array = [  
  { ... },  
  { ... },  
  { ... },  
  { ... },  
  { ... }  
];
```

Проблема 1

Данные теряются при перезагрузке сервера.

Это самая очевидная проблема, ведь при разработке сервер перезагружается постоянно, и вы наверняка не раз с этим сталкивались.

Проблема 2

Что будет, если данных очень много?

В таком случае **добавление, удаление и поиск** элемента в массиве будут выполняться очень **медленно**.

Решение очевидно :)

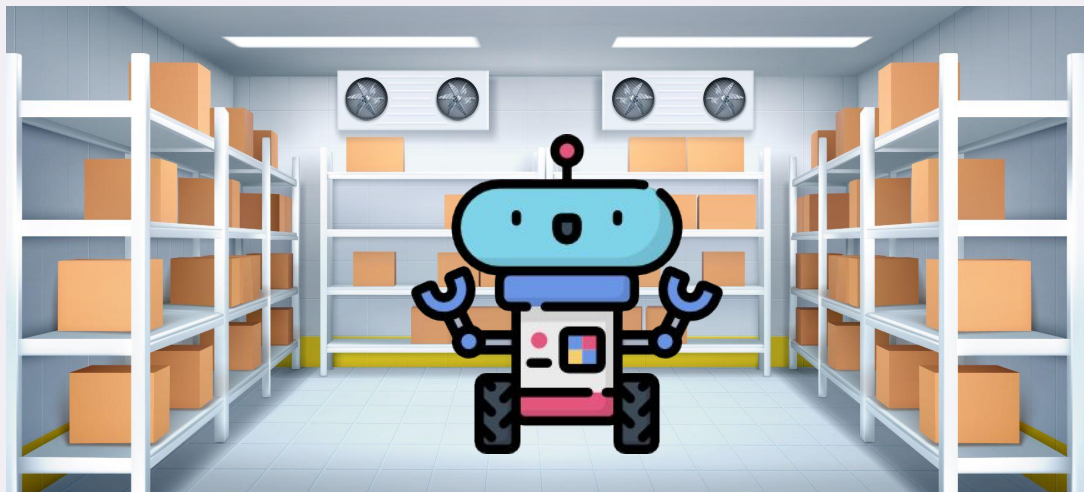
Нужно перенести все данные в БД

База данных – это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе.





- Данные хранятся системно
- Их можно быстро отыскать даже среди миллиона других данных



Кто же управляет таким крутым складом?

СУБД – Система Управления Базами Данных.

Минутка определений



Далее мы (как и весь мир) будем использовать только термин **БД**

Какие бывают БД?

1. MySQL
2. PostgreSQL
3. SQLite
4. MongoDB
5. Redis
6. И другие)



- База данных от Mongo очень проста в использовании
- Подходит для web-разработки
- Не требует изучения нового синтаксиса или языка

Структура MongoDB

Database

chat-app

Collection

chats

users

stickers

Document

```
{
  "name": "Todd Paucek",
  "sex": "male",
  "age": 66
}
```

Документ

```
▼ {  
  ▼ "_id": {  
    "$oid": "6412193d821970af174eaa90"  
  },  
  "name": "Todd Paucek",  
  "sex": "male",  
  "age": 66,  
  "eyes": "green",  
  "hair": "brown",  
  "height": 176,  
  "rating": 4.22  
}
```

Все документы записываются в формате JSON

Документ

```
▼ {  
  ▼ "_id": {  
    "$oid": "6412193d821970af174eaa90"  
  },  
  "name": "Todd Paucek",  
  "sex": "male",  
  "age": 66,  
  "eyes": "green",  
  "hair": "brown",  
  "height": 176,  
  "rating": 4.22  
}
```

Каждый ключ и строковое значение записывается в **двойных кавычках**

Документ

```
▼ {  
  ▼ "_id": {  
    "$oid": "6412193d821970af174eaa90"  
  },  
  "name": "Todd Paucek",  
  "sex": "male",  
  "age": 66,  
  "eyes": "green",  
  "hair": "brown",  
  "height": 176,  
  "rating": 4.22  
}
```

После добавления каждому документу присваивается поле `_id`. Это уникальный номер документа.

Сейчас мы будем работать в программе MongoDB Compass

project1.actors 1.0k DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter  Type a query: { field: 'value' } Reset Find </> More Options

ADD DATA EXPORT COLLECTION 1 - 20 of 1001 < > ≡ { } 🗑

★ actors

	_id ObjectId	name String	sex String	age Int32	ey
1	ObjectId('6412193d821970af1...')	"Todd Paucek"	"male"	66	"g"    
2	ObjectId('6412193d821970af1...')	"Traci Stroman DDS"	"female"	53	"b"    
3	ObjectId('6412193d821970af1...')	"Georgia Prohaska"	"female"	40	"g"    
4	ObjectId('6412193d821970af1...')	"Carla Kunze"	"female"	74	"g"    
5	ObjectId('6412193d821970af1...')	"Natasha Howell"	"female"	18	"g"    
6	ObjectId('6412193d821970af1...')	"Rachel Langosh"	"female"	53	"g"    
7	ObjectId('6412193d821970af1...')	"Leah Kuhlman"	"female"	23	"g"    
8	ObjectId('6412193d821970af1...')	"Bradley Pagac"	"male"	71	"b"    

Практика на весь урок

1. Установим MongoDB
2. Создадим документы
3. Настроим поиск

Теория

Поисковые запросы

Поиск по одному условию

В поле **Filter** на странице коллекции можно записать условие для поиска нужного документа.

```
{ ключ: значение }
```

Найденный результат называют **выборкой**.

В данном случае в **выборку** попал один документ, у которого ключ **name** совпал с поиском.

The screenshot shows a web interface for searching a collection. At the top, there is a search bar with the label "Filter" and a dropdown arrow. The filter text is "{name: 'Todd Paucek'}". To the right of the filter are buttons for "Reset", "Find", and a code icon, followed by a link "More Options". Below the search bar are two buttons: "ADD DATA" and "EXPORT COLLECTION". To the right of these buttons is a status indicator "1 - 1 of 1" and navigation icons. The main area displays a single document in a code-like format:

```
_id: ObjectId('6412193d821970af174eaa90')
name: "Todd Paucek"
sex: "male"
age: 66
eyes: "green"
hair: "brown"
height: 176
rating: 4.22
```

Поиск по нескольким условиям

Если условий для поиска несколько, они записываются через запятую

```
{ ключ1: значение1, ключ2: значение2 }
```

В результат выборки здесь попало 127 документов, для которых выполняются **оба условия одновременно**.

Filter



{sex: "male", eyes: "green"}

Reset

Find



More Options ▶

ADD DATA ▼

EXPORT COLLECTION

1 – 20 of 127



Поиск по диапазону

Задача: сколько пользователей в базе данных в возрасте от 16 до 20 лет?

```
{ age: { $gte: 16, $lte: 20 } }
```

В Mongo существуют свои операторы для сравнения чисел:

- **\$gte** – greater than or equal to – больше либо равно
- **\$lte** – less than or equal to – меньше либо равно

Используй их, чтобы искать данные в диапазоне.

Комбинируй с другими запросами:

[Filter](#)

```
{ sex: "male", eyes: "green", age: { $gte: 16, $lte: 20 } }
```

[Reset](#)[Find](#)[More Options ▶](#)

Сортировка

Поле **Sort** отвечает за сортировку выборки.

- **{age: 1}** отсортирует от меньшего возраста к большему
- **{age: -1}** наоборот, от большего возраста к меньшему

Filter 

 ▼

{eyes: "green"}

Reset

Find



Less Options ▼

Project

{ field: 0 }

Sort

{age: 1}

MaxTimeMS 60000

Collation

{ locale: 'simple' }

Skip 0

Limit 5

Лимит

Поле **Limit** ограничивает выборку по количеству документов.

Например, из 1000 найденных и отсортированных документов мы увидим только первые 5.

Filter 
 ▼

{eyes: "green"}

Reset

Find



Less Options ▼

Project

{ field: 0 }

Sort

{age: 1}

MaxTimeMS 60000

Collation

{ locale: 'simple' }

Skip 0

Limit 5

Цель урока:

- ✓ Установили нужные программы
- ✓ Научились добавлять тысячи записей БД
- ✓ Составили условия поиска по БД

Домашнее задание