МШП 20.35 УНИВЕРСИТЕТ КОД БУДУЩЕГО Минцифры России НАЦИОНАЛЬНЫЕ ПРОЕКТЫ РОССИИ ЦИФРОВАЯ ЭКОНОМИКА

Fullstack-разработка

Mongo
GET-запросы

МШП

Цели урока:

- Подключим БД к серверу
- Покажем данные из БД пользователю сайта

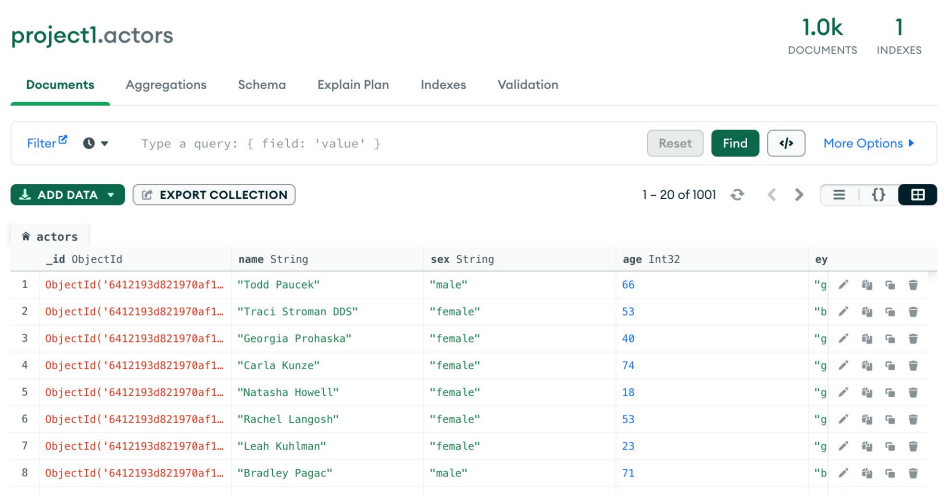
МШП

Теория

Express + Mongo

МШП

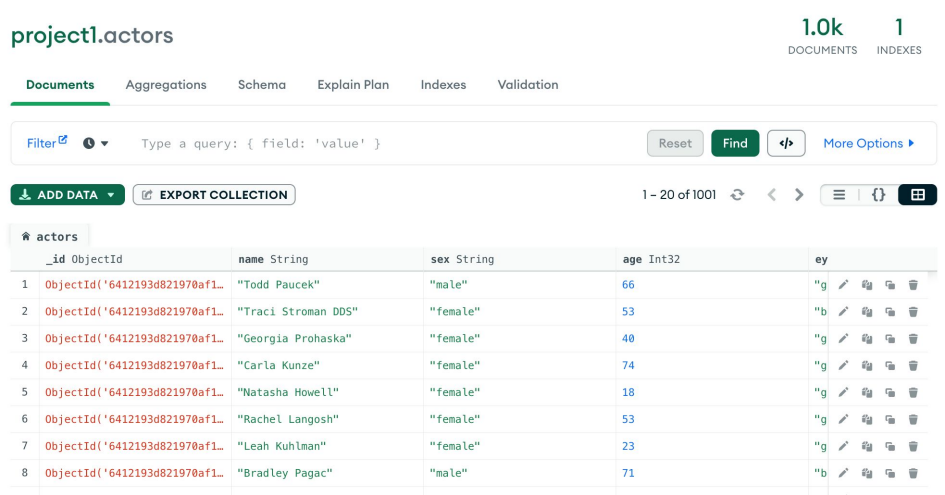
На прошлом уроке мы познакомились с программой **MongoDB Compass**



МШП

Что может делать **Compass**?

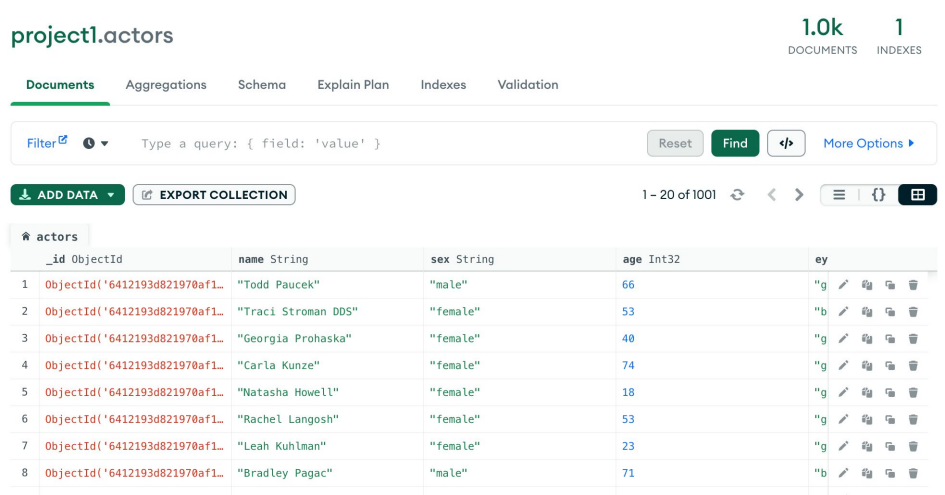
- Устанавливать соединение
- Посылать запросы
- Показывать ответ :)



МШП

Вывод 1

Compass это всего лишь **клиент** для базы данных **MongoDB**

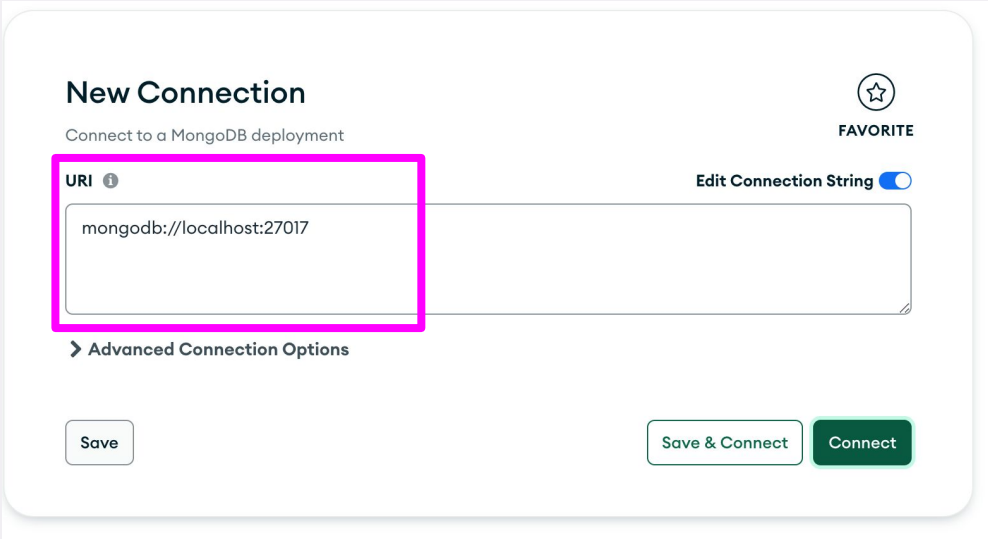


МШП

Вывод 2

На самом деле **MongoDB это сервер!**

У него и URL есть:

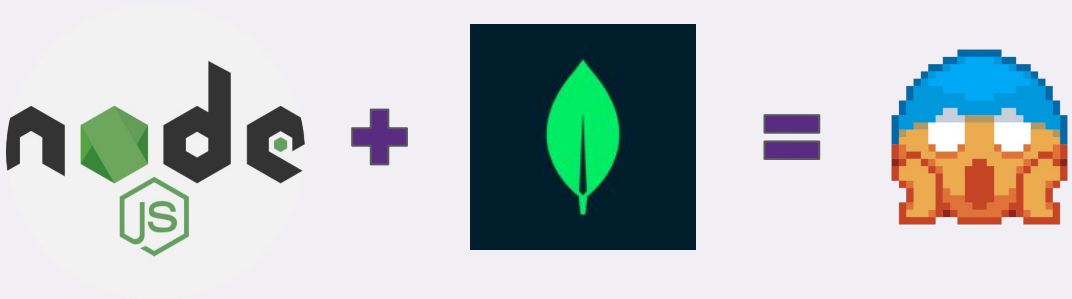


МШП

Если **Express** это сервер....

И **MongoDB** это сервер....

Как они будут взаимодействовать???

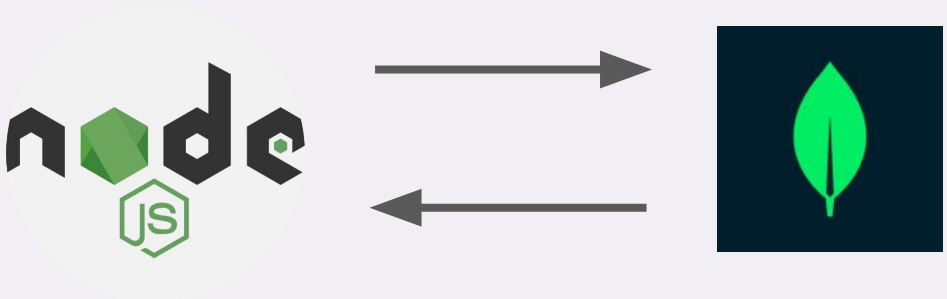


МШП

Ответ есть!

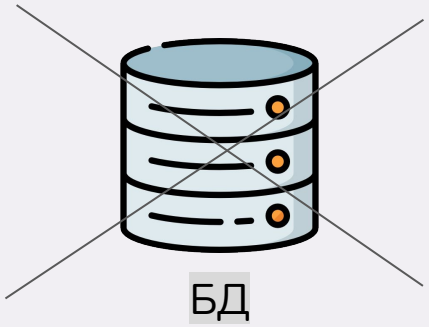
Во взаимодействии Express-Mongo **Express** выступит как **клиент и будет делать запросы.**

MongoDB будет в ответ отдавать запрошенные данные.



МШП

Сервер без БД



handlebars

array

index.hbs

index.html

localhost:3000/

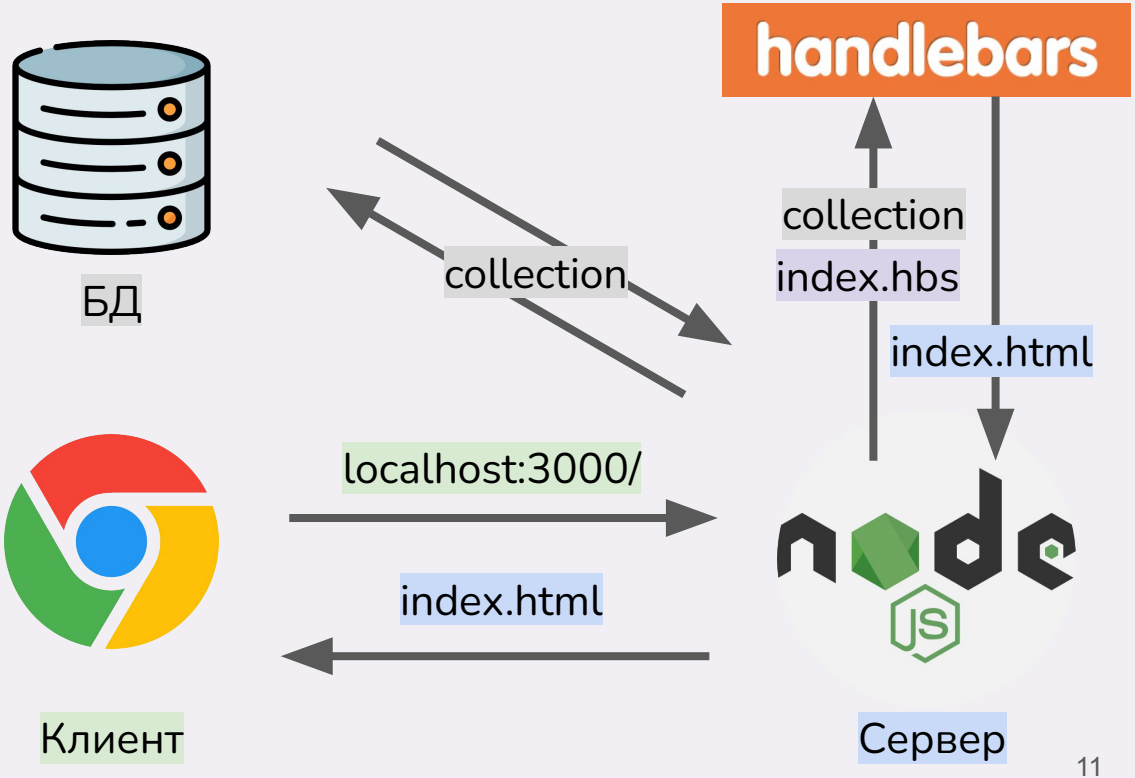
index.html

Клиент

Сервер

МШП

Сервер с БД



МШП

Итоги

- Express выступает как сервер для браузера
- Express выступает как клиент для БД

МШП

Практика

Подключение к БД

МШП

Подключение к БД

Чтобы Express смог подключиться к БД, нам нужно сделать следующее:

1. Проверить, что **MongoDB** запущен
2. Узнать адрес сервера **MongoDB**
3. Установить npm пакет **mongoose**
4. Добавить код подключения в **index.js**

МШП

Установка пакета

Mongoose это специальный пакет для работы с MongoDB. Сегодня мы сделаем первый шаг к его изучению.

Выполни в терминале VSCode команду:

```
npm install mongoose
```

МШП

Установка соединения

- Добавь эти строчки после других настроек сервера.
- Скопируй ссылку для подключения из MongoDB Compass.
- Укажи **имя базы данных,** к которой подключаешься.

index.js

```
let mongoose = require('mongoose');

mongoose.connect('mongodb://127.0.0.1:27017/project2');
```

МШП

Теория

Mongoose

МШП

Через пакет **Mongoose** можно делать всё то же самое, что и в **Compass:**

1. **Искать по коллекции**
2. Добавлять новые документы
3. Удалять документы

В этом нам помогут **Схема** и **Модель**

МШП

Допустим, у нас есть коллекция книг **books**:

```
_id: ObjectId('641572c5943deb7ae93889ad')
title: "Aliquid repellendus impedit illum praesentium."
year: 1983
isRead: true
```

```
_id: ObjectId('641572c5943deb7ae93889ae')
title: "Eius."
year: 1934
isRead: false
```

```
_id: ObjectId('641572c5943deb7ae93889af')
title: "Debitis alias architecto rerum voluptate."
year: 1913
isRead: false
```

МШП

Схема документа

Schema описывает структуру полей документа в коллекции: **название поля** и **тип данных**.

Например, это схема для коллекции **books:**

```
let schema = new mongoose.Schema({
    title: String,
    year: Number,
    isRead: Boolean
});
```

index.js

_id указывать не нужно!

МШП

Модель коллекции

Model представляет собой саму **коллекцию.**

Название модели соответствует названию коллекции в единственном числе.

Для создания модели нужна **схема.**

```
let schema = new mongoose.Schema({
    title: String,
    year: Number,
    isRead: Boolean
});

let Book = mongoose.model('book', scheme);
```

МШП

Схема нужна, чтобы mongoose правильно обрабатывал документы коллекции с учётом типов данных.

Модель нужна, чтобы делать запросы к коллекции.

МШП

Model.find

Метод **find** делает запрос к БД и получает **массив объектов** всех документов в коллекции.

```
app.get(`/`, async function (req, res) {
    let data = await Book.find();
    res.send(data[0]);
})
```

Ответ сервера:


```
{"_id":"641572c5943deb7ae93889ad","title":"A
liquid repellendus impedit illum
praesentium.","year":1983,"isRead":true}
```

МШП

async / await

А это что за новшества такие?

```
app.get(`/`, async function (req, res) {
    let data = await Book.find();
    res.send(data[0]);
})
```

Знакомьтесь: **асинхронный код!**

МШП

async / await

Асинхронный код – это код, который возвращает результат через неизвестное время.

```
app.get(`/`, async function (req, res) {
    let data = await Book.find();
    res.send(data[0]);
})
```

Например, БД может выполнить один запрос за 1 миллисекунду, а другой за 2 секунды.

МШП

async / await

Функции, в которых выполняется асинхронный код, помечаются словом **async**.

```
app.get(`/`, async function (req, res) {
    let data = await Book.find();
    res.send(data[0]);
})
```

Метод **find** работает асинхронно, поэтому мы будем **дожидаться** его выполнения через ключевое слово **await**.

МШП

Не делай так!

Если мы забудем использовать ключевые слова **async** и **await**, то в переменной **data** не будет данных коллекции :(

Ведь мы были нетерпеливы и отправили клиенту данные до того, как сами получили их.

```
app.get(`/`, function (req, res) {
    let data = Book.find();
    res.send(data); // пусто :(
})
```

МШП

Model.find

Метод find работает точно так же, как поле **Filter в MongoDB.**

Можно искать по нескольким условиям и диапазону чисел:

```
Actor.find({
    eyes: 'green',
    sex: 'female',
    age: {$gte: 20, $lte: 30}
});
```

МШП

Model.sort

Итоговый результат поиска легко отсортировать по полю:

```
Actor.find({
    eyes: 'green',
    sex: 'female',
    age: {$gte: 20, $lte: 30}
}).sort({
    rating: -1
})
```

МШП

Model.limit

После сортировки выборку можно **обрезать**, оставив в ней только первые 5 элементов.

```
Actor.find({
    eyes: 'green',
    sex: 'female',
    age: {$gte: 20, $lte: 30}
}).sort({
    rating: -1
}).limit(5);
```

МШП

Model.findOne

Метод find всегда возвращает массив.
Но что, если хочется найти только один единственный уникальный объект?

```
app.get(`/`, async function (req, res) {
    let book = await Book.findOne({
        title: 'Властелин колец'
    });
    res.send(book);
})
```

Метод **findOne** решает именно эту задачу, возвращая искомый объект.

МШП

Как использовать Mongoose?

- Создай схему
- Создай модель
- Используй методы модели для поиска по коллекции:
  - find
  - findOne
  - sort
  - limit

МШП

Практика

Страница с товарами
Фильтр товаров
Страница товара

МШП

Цели урока:

- ✓ Подключим БД к серверу
- ✓ Покажем данные из БД пользователю сайта

МШП

Домашнее задание