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

Fullstack-разработка

Mongo
POST-запросы
Добавление данных

МШП

Цель урока:

Научимся добавлять и удалять данные из БД с помощью **mongoose**

МШП

Разбор ДЗ

А было ли оно?

МШП

Практика

Поиск по товарам

МШП

Теория

Операции с данными

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**

Эти действия доступны через **Модель Коллекции**

МШП

Для демонстрации с нами снова 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

Модель коллекции

У документов в коллекции
всего 3 поля:

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

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

МШП

Создание документа

С помощью **названия модели** мы можем создать **новый документ** для коллекции:

```
app.post(`/create`, async function (req, res) {
    let book = new Book({
        title: "JavaScript: сильные стороны",
        author: "Дуглас Крокфорд",
        isRead: false
    });



})
```

МШП

Model.save

Документ нужно асинхронно сохранить в БД с помощью метода **save**:

```
app.post(`/create`, async function (req, res) {
    let book = new Book({
        title: "JavaScript: сильные стороны",
        author: "Дуглас Крокфорд",
        isRead: false
    });

    await book.save();

    res.send('Дело сделано');
})
```

МШП

Для создания новых документов
всегда используй метод **POST,**
так как он передаёт данные
от клиента к серверу
в **теле запроса**

МШП

Model.save

Часть значений может передаваться **из формы**,
но некоторые можно задавать **по умолчанию:**

```
app.post(`/create`, async function (req, res) {
    let book = new Book({
        title: req.body.title,
        author: req.body.author,
        isRead: false
    });

    await book.save();

    res.send('Дело сделано');
})
```

МШП

Model.deleteOne

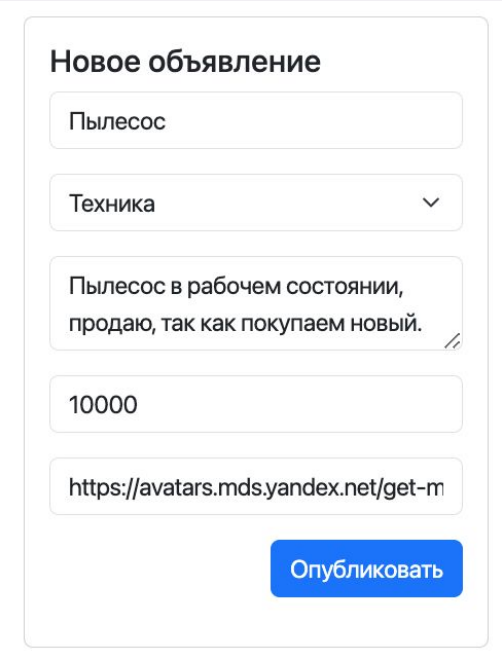
Удаление – это операция, которая использует внутри себя поиск.

Составь условие, и метод **deleteOne** удалит **одну** запись, которая подойдёт под это условие.

```
app.get(`/remove`, async function (req, res) {
    await Book.deleteOne({
        title: 'Властелин колец'
    });
    res.send('Дело сделано');
})
```

МШП

Попробуем добавить новое объявление в новый раздел нашего сайта



МШП

Практика

Создание

Удаление

МШП

Теория

Проверка данных

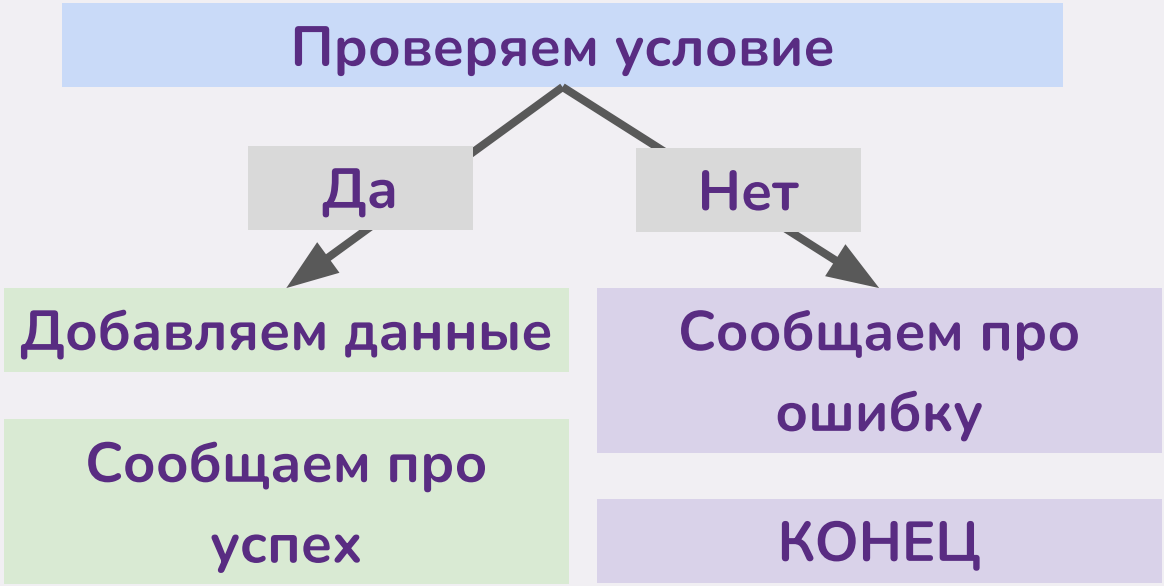
МШП

Минутка тестировщика:

Найдите как можно больше проблем с нашей формой добавления

МШП

Алгоритм проверки



МШП

Алгоритм проверки

return – это выход из кода. Ни одна следующая строчка не выполнится.

```
app.post(`/create`, async function (req, res) {
    if (условие) {
        res.send('Ошибка');
        return;
    }

    let book = new Book({...})
    await book.save();
    res.send('Дело сделано');
})
```

МШП

Нет данных

Подумай, какие данные необходимы,
а какие могут быть пропущены!

```
app.post(`/create`, async function (req, res) {
    let title = req.body.title;

    if (!title) {
        res.send('Ошибка');
        return;
    }
    ...
})
```

МШП

Дубли

Наверняка в твоей коллекции будут **уникальные поля.** Тогда стоит перед добавлением проверить, нет ли уже такого документа, с помощью метода **Model.exists**

```
app.post(`/create`, async function (req, res) {
    let title = req.body.title;
    let data = await Book.exists({title: title});

    if (data) {
        res.send('Ошибка');
        return;
    }
    ...
})
```

МШП

Странные данные

Например, в БД нужно число, а нам передают строку.

Или просим имя человека, а оно из одной буквы.

Подозрительно!

Продумай, как пользователь может пытаться **сломать систему,** и добавь соответствующую проверку.

МШП

Показать ошибку

Для отладки

```
res.send('Ошибка')
```

Отдельный шаблон

```
res.redirect('/error')
```

В шаблон рядом с формой

```
res.redirect('/?error=1)
```

```
res.redirect('/?success=1)
```

МШП

Показать ошибку

Если выбрать последний способ, то затем можно передать ошибку в шаблон и через {{#if}} показать нужное сообщение.

```
app.post(`/`, async function (req, res) {
    let error = req.query.error;
    let success = req.query.success;

    res.render('index', {
        error: error,
        success: success
    });
})
```

МШП

Практика

Проверка данных

МШП

Цель урока:

✓ Научились добавлять и удалять данные из БД с помощью **mongoose**

МШП

Домашнее задание