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

Fullstack-разработка

Mongo
POST-запросы
Изменение данных

МШП

Цель урока:

Разработать форму

редактирование заказа

МШП

Разбор ДЗ

Что получилось?

МШП

Теория

Изменение документа

МШП

На данный момент мы умеем пользоваться следующими методами:

1. **find, sort, limit**
2. **findOne**
3. **deleteOne**
4. **save**

МШП

Model.save

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

    // изменение свойства
    book.isRead = true;

    // сохранение документа
    await book.save();

    res.send(book);
})
```

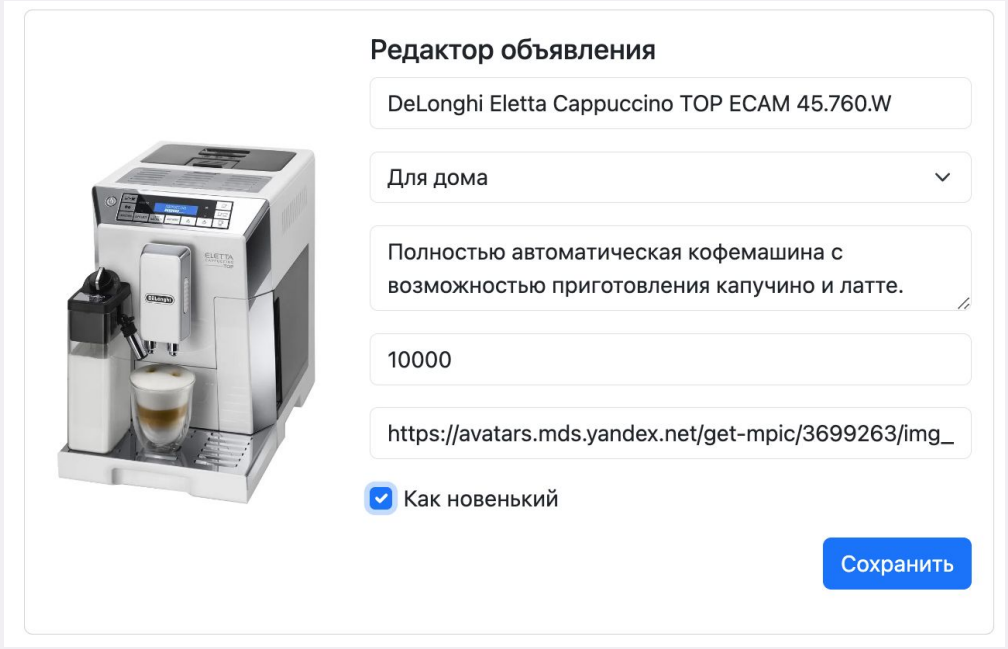
МШП

Синтаксис изменения модели очень прост!

Однако важно правильно показать и обработать каждое поле формы для редактирования.

МШП

Давайте разбираться вместе...



МШП

Практика

Редактор объявления

МШП

Практика

Редактор объявления

Валидация схемы

МШП

Теория

Полезная теория

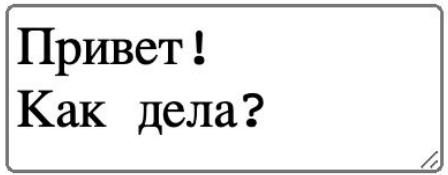
МШП

Поля ввода

В текстовые поля ввода можно подставить значение по умолчанию.

```
<input type="text" name="username"
       value="Тестер">
```


```
<textarea name="message">
Привет!
Как дела?
</textarea>
```



МШП

Поля ввода

С помощью атрибута **checked** можно сделать чекбокс выбранным по умолчанию.

```
<input type="checkbox" name="agree">
```


```
<input checked type="checkbox" name="agree">
```

МШП

Поля ввода

По умолчанию в тег **select** подставляется первый **option**.

Это можно изменить, добавив к нужному **option** атрибут **selected.**

```
<select name="direction">
   <option value="right">Направо</option>
   <option value="left" selected>Налево</option>
</select>
```

МШП

Условие if

Условный оператор очень полезен, когда нужно вставить в итоговый шаблон фрагмент кода по условию.

```
{{#if isRead}}Прочитано{{/if}}
```

```
{{#if isRead}}
    Прочитано
{{else}}
    Не прочитано
{{/if}}
```

МШП

Вставка текста

Двойные скобки {{ }} умеют вставлять строки только как **текст.**

```
res.render(`index`, {
  text: '<h1>!!!</h1>'
});
```

```
{{text}}
```

<h1>!!!</h1>

МШП

Вставка HTML

Тройные скобки {{{ }}} умеют вставлять строки как **HTML теги.**

```
res.render(`index`, {
  text: '<h1>!!!</h1>'
});
```

```
{{{text}}}
```

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

Валидация схемы

Валидация – автоматическая проверка полей документа по заданным в схеме условиям.

В mongoose валидация происходит, когда вызывается метод **save**().

```
let schema = new mongoose.Schema({
    username: {
        type: String,
        required: true, // обязательно
        unique: true    // уникально
    },
    age: {
        type: Number,
        min: 0,         // от
        max: 120        // до
    },
    isAdmin: Boolean
});
```

МШП

Исключения

Если документ
не соответствует правилам валидации, то в момент выполнения метода **save** произойдёт **exception – исключение.**

Исключение – это ошибка, которая останавливает выполнение кода.

```
app.get(`/`, async function (req, res) {
    let user = await User.findOne({
        username: 'tester'
    });

    // изменение свойства
    user.age = 10000;

    // EXCEPTION, возраст неверный
    await user.save();

    res.send(user);
})
```

МШП

Обработка исключения

Исключения "отлавливают" с помощью конструкции **try / catch,** которая немного похожа на **if / else.**

Если внутри блока **try** произойдёт исключение, то выполнится код внутри **catch**.

```
app.get(`/`, async function (req, res) {
    ...

    try {
        // План А (идеальный)
        await user.save();
        res.send(user);
    } catch (err) {
        // План Б (что-то пошло не так)
        res.send('error!');
    }
})
```

МШП

Цель урока:

✓ Разработали форму редактирование заказа

МШП

Что умеем?

1. **Отправлять** данные на сервер через GET и POST **запросы**
2. Выводить данные через **шаблонизатор**
3. Взаимодействовать с базой данных через библиотеку **mongoose**

Кстати, это наше последнее теоретическое занятие перед аттестацией :)

МШП

Домашнее задание