

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 2. “RESTful API средствами express
+ typescript”

Выполнил:

Тимофеев Николай

Группа

К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

Задача

Реализовать сайт администратора интернет-магазина

- Вход
- Регистрация
- Учёт товара на складе
- Графики по продажам тех или иных товаров, по общей выручке предприятия
- Управление сотрудниками

Ход работы

1. Создание моделей

Модель Employee - сотрудник предприятия

```
@Table
class Employee extends Model {
    @AllowNull(false)
    @Unique
    @Column
    name: string

    @AllowNull(false)
    @Column
    position: string

    @Column
    active: boolean = true

    @ForeignKey(() => Warehouse)
    @Column
    warehouseId: number

    @BelongsTo(() => Warehouse)
    warehouse: Warehouse;
}
```

Модель Deals - совершенные сделки

```
@Table
class Deal extends Model {
  @AllowNull(false)
  @Column
  amount: number;

  @AllowNull(false)
  @ForeignKey(() => Unit)
  @Column
  unitId: number;

  @AllowNull(false)
  @ForeignKey(() => Employee)
  @Column
  employeeId: number;

  @BelongsTo(() => Unit)
  unit: Unit;

  @BelongsTo(() => Employee)
  employee: Employee;
}
```

Модель Unit - товар

```
@Table
class Unit extends Model {
  @AllowNull(false)
  @Unique
  @Column
  title: string;

  @AllowNull(false)
  @Column
  price: number;

  @Column
  stockAmount: number;

  @ForeignKey(() => Warehouse)
  @Column
  warehouseId: number;

  @BelongsTo(() => Warehouse)
  warehouse: Warehouse;
}
```

Модель Warehouse - склад

```
@Table
class Warehouse extends Model {
  @Column
  title: string

  @Column
  address: string

  @HasMany(() => Unit)
  units: Unit[]

  @HasMany(() => Employee)
  employees: Employee[]
}
```

2. Сервисы

Deals:

- get - получение списка всех сделок
- getById - получение конкретной сделки по ID
- create - создание и валидация новой сделки
- revenue - расчет прибыли предприятия по дням от всех сделок

Employees:

- get - получение списка всех сотрудников нашего предприятия
- getById - получение конкретного сотрудника по ID
- create - создание и валидация нового сотрудника
- changeStatus - активация/деактивация аккаунта сотрудника

Unit:

- get - получение списка всех товаров
- getById - получение конкретного товара по ID
- create - создание и валидация нового товара
- search - поиск товара (regex) по названию
- analytics - сколько товаров такого типа находится на складах в принципе

Warehouses:

- get - получение списка всех складов
- getById - получение конкретного склада по ID
- create - создание и валидация нового склада

3. Роуты

Сделки:

```
const controller: DealController = new DealController();

router.route("/").post(controller.post);
router.route("/").get(controller.getAll);
router.route('/revenue').get(controller.revenue);
router.route("/:id").get(controller.get);
```

Сотрудники:

```
const controller: EmployeeController = new EmployeeController();

router.route("/").post(controller.post);
router.route("/").get(controller.getAll);
router.route("/:id").get(controller.get);
router.route("/:id").delete(controller.changeStatus);
```

Товары:

```
const controller: UnitController = new UnitController();

router.route("/").post(controller.post);
router.route("/").get(controller.getAll);
router.route("/analytics").get(controller.stockGraph);
router.route("/:id").get(controller.get);
router.route('/search/:query').get(controller.search);
```

Склады:

```
const controller: WarehouseController = new WarehouseController();

router.route("/").post(controller.post);
router.route("/").get(controller.getAll);
router.route("/:id").get(controller.get);
```

routes/v1/index.ts

```
const router: express.Router = express.Router()

router.use('/users', userRoutes)
router.use('/employees', employeeRoutes)
router.use('/warehouses', warehouseRoutes)
router.use('/units', unitRoutes)
router.use('/deals', dealRouter) You, 10 минут назад • fix: lr2 multiple serv
```

Вывод

В ходе выполнения работы я ознакомился с продвинутыми возможностями ORM sequelize (literal, op, raw sql query и тд) и написал полноценное REST API с использованием express и typescript.