

# Тестовое задание на позицию Backend-разработчик

## Краткое описание.

Требуется создать веб-сервер на базе ExpressJS, который будет работать с данными по занятиям. Данные хранятся в СУБД PostgreSQL, дамп тестовых данных нужно скачать и развернуть: <https://disk.yandex.ru/d/dtr3sxsuceMfCw>

Исполнителю предлагается сделать задачу, выбирая адекватные инструменты и общепринятые способы организации кода и API-интерфейсов, учитывая указанные в задании требования. При разработке учитывать, что данных может быть очень много (миллионы занятий). Задача должна быть сделана в точном соответствии с заданием.

Реализация веб-сервера в данном задании - вторична. Важна именно работа с данными и соответствие стандартам.

## Описание таблиц

### Таблица lessons

Содержит занятия. Поля:

id,

date (дата занятия),

title (описание, тема занятия),

status (статус занятия, тип - число. 1-проведено, 0 - не проведено)

### Таблица teachers

Содержит учителей. Поля:

id,

name (имя)

### Таблица students

Содержит учеников. Поля:

id,

name (имя)

### Таблица lesson\_teachers

Связка учителей и занятий - кто какое занятие ведет. Может быть так, что одно занятие ведет несколько учителей.

## Таблица lesson\_students

Связка учеников и занятий - кто на какое занятие записан. Есть дополнительное поле visit - посетил ученик занятие, или нет.

## Общие требования

Язык: JavaScript или TypeScript

Веб-сервер: ExpressJS

Версия NodeJS: 16 или выше

Работа с СУБД через knex или sequelizeJS

Используемый Content-Type при работе - application/json

Код должен быть выложен на GitHub или GitLab, система контроля версий - git.

## Задача. Запрос данных.

Сделать метод api "/lessons", который осуществляет поиск по данным и возвращает массив объектов - занятий.

Метод принимает параметры фильтра. Все параметры необязательные. Все параметры должны учитываться одновременно. С помощью параметров должна работать пагинация.

Параметры фильтра:

- **date.** Либо одна дата в формате YYYY-MM-DD, либо две в таком же формате через запятую (например, «2019-01-01,2019-09-01»). Если указана одна дата, выбираются занятия на эту дату. Если указаны 2 даты, то выбираются занятия за период, включая указанные даты.
- **status.** Статус занятия. принимается либо 0 (не проведено), либо 1 (проведено)
- **teacherIds.** id учителей через запятую. Выбираются все занятия, которые ведет хотя бы один из указанных учителей.
- **studentsCount.** количество записанных на занятия учеников. либо одно число (тогда выбирается занятие с точным числом записанных), либо 2 числа через запятую, тогда они рассматриваются как диапазон и выбираются занятия с количеством записанных, попадающих в диапазон включительно.
- **page.** Номер возвращаемой страницы. первая страница - 1.
- **lessonsPerPage.** Количество занятий на странице. По-умолчанию - 5 занятий.

В случае некорректных данных метод должен возвращать ошибку 400 с описанием ошибки (формат на выбор исполнителя).

В нормальном случае возвращается массив объектов-занятий. Каждый объект должен иметь вид:

```
{
  id : 9 // id занятия
  date: '2019-09-01' // Дата занятия
  title: 'Orange', // Тема занятия
  status: 1 // Статус занятия
  visitCount: 3, // Количество учеников, посетивших занятие (по полю visit)
  students: [ // Массив учеников, записанных на занятие
    { id: 1, // id ученика
```

```
    name: 'Ivan' // имя
    visit: true,
  }
],
teachers: [ // Массив учителей, ведущих занятие
  { id: 1, // id учителя
    name: 'Tanya' // имя
  }
]
}
```