# Задача:

Реализовать REST API на фреймворке Laravel последней версии позволяющее управлять отделами и сотрудниками предприятия.

**API должно отдавать JSON и иметь по три метода для сущностей “Отделы” и “Сотрудники”**:

1. Получить все записи с пагинацией;
2. Создание/редактирование записи;
3. Удалить запись;

**Сущность “Отдел” должна содержать следующую информацию:**

1. ID;
2. Название отдела;

**Сущность “Сотрудник” должна содержать следующую информацию:**

1. ID;
2. Имя;
3. Фамилия;
4. Отчество;
5. Пол;
6. Заработная плата (целое число);

**Связь между сущностями “Сотрудник” и “Отдел” многие-ко-многим.**

**При создании/редактировании сущности “Сотрудник” на входе обрабатывать следующие поля:**

1. Имя (обязательно для заполнения);
2. Фамилия (обязательно для заполнения);
3. Отчество (обязательно для заполнения);
4. Пол;
5. Заработная плата (целое число, обязательно для заполнения);
6. Массив ID записей сущности “Отдел”. Т.е. указываем к каким отделам относится пользователь. Минимум одно значение в массиве.

**При создании/редактировании сущности “Отдел” на входе обрабатывать следующие поля:**

1. Название отдела (обязательно для заполнения);

**Метод получения списка отделов содержит следующие данные:**

1. ID;
2. Название отдела;
3. Количество сотрудников отдела;
4. Максимальная заработная плата среди сотрудников отдела;

**Метод получения списка сотрудников содержит следующие данные:**

1. ID;
2. ФИО (разделены пробелом);
3. Пол;
4. Заработная плата (целое число, обязательно для заполнения);

**Принцип работы удаления сущности “Отдел”:**

1. Нельзя удалить отдел в котором есть сотрудники.

Возвращать в формате JSON **ошибки, возникающие при редактировании/создании/удалении**.

При выполнении задания, желательно **минимизировать количество SQL-запросов** к СУБД. Для получения данных стараться использовать **возможности языка SQL** в реализации MySQL.

Составить документацию по спецификации OpenAPI. Например здесь <https://editor.swagger.io>. Можно максимально сократить, например только роут и входные параметры если есть. Желательно конвертировать в JSON и отдавать по роуту */api* или в крайнем случае сложить где-нибудь в проекте yaml или json файл.

**Результатом работы** должен являться репозиторий (GitHub, BitBucket etc.) с README файлом, в котором описано, что вы сделали, что не сделали, но хотели бы (и почему), как это развернуть (будет плюсом, если развернете сами и предоставите ссылку), затраченное время на реализацию.

**С чего начать?**

Форкаем себе репозиторий <https://github.com/websecret/donut-hole>. Клонируем к себе на рабочую машину, пишем код, пушим в свой репозиторий. Обязательно прочитать README файл.