

Тестовое задание для собеседования

Предыстория задания:

К вам пришел заказчик – владелец завода по производству свеч от геморроя “Здоровая норка”. У него есть подозрения что работники постоянно опаздывают на работу или уходят раньше. Он просит вас создать систему, которая могла бы отслеживать приход и уход на работу каждого сотрудника.

Каждый сотрудник на проходной будет прикладывать карточку к считывателю, который будет отправлять информацию о сотруднике в данную систему. От вас потребуется получить номер пропуска сотрудника, найти его в базе и записать его время прохождения КПП на вход или на выход.

Детали реализации:

1. Проект должен быть в формате **ASP.NET Web API**, без авторизации, визуальное оформление (razor pages) не нужно, просто контроллеры api, принимающие данные и (если нужно) возвращающие результат в формате JSON.
2. Необходимо создать базу сотрудников предприятия с возможностью добавления, удаления и изменения данных сотрудников, а так же вывода списка всех имеющихся работников. (т.е. простейший CRUD - интерфейс). Для простоты, формат БД – **Sqlite**. Для создания базы понадобится **EntityFramework Core 5.0** и провайдер для **Sqlite**, доступный в NuGet под названием `Microsoft.EntityFrameworkCore.Sqlite`.
3. **Сотрудники** должны иметь следующие данные:
 1. номер пропуска (ID),
 2. Фамилия,
 3. Имя,
 4. Отчество,
 5. Должность,
 6. Список всех входов/выходов с завода (назовем их смены).
4. **Должности** у сотрудников будут трех видов:
 1. Менеджер,
 2. Инженер,
 3. Тестировщик свечей
5. **Смены** содержат:
 1. id смены (поле для БД)
 2. время начала смены (когда человек пришел на завод “Здоровая норка”),
 3. конца смены (когда ушел)
 4. количество отработанных часов (заполняется в конце смены)
 5. ссылку на работника.
6. По итогу, должен получиться проект WebApi с двумя контроллерами:

1. Контроллер “отдела кадров”, с возможностью создать, удалить или изменить данные работников.
2. контроллер “КПП”, в который можно отправить номер пропуска сотрудника и дату/время входа/выхода.
3. (бонус, если все вышеперечисленное было слишком просто) в контроллере “отдела кадров” сделать вывод статистики нарушений работников за месяц (данные необязательно заполнять в базу самим, можно использовать генератор). Если сотрудник приходит на работу позже 9:00 и уходит раньше 18:00, проставлять ему замечание и выводить кол-во этих замечаний за месяц. Тестировщики свечей от геморроя работают по 12 часов, их время ухода со смены должно быть не раньше 21:00, иначе проставляется замечание.

Детали работы контроллеров:

Все контроллеры должны возвращать либо Ok (200) и, если требуется, результат запроса в JSON, либо BadRequest (400) и сообщение об ошибке.

Кпп:

1. Содержит 2 метода (экшена) StartShift и EndShift, оба принимают Id сотрудника (он же номер пропуска) и время начала (StartShift) / конца (EndShift) смены.
2. Если сотрудника с заданным номером пропуска нет в базе – возвращается ошибка.
3. Если сотрудник зашел на завод и вышел не отметившись, он не может пройти кпп на вход пока не закроет предыдущую смену – возвращается ошибка.
4. Если сотрудник каким-то образом попал на завод не пробив пропуск, он не может выйти пока не отметит начало своей смены – возвращается ошибка.

Отдел кадров:

1. Методы контроллера:
 1. Добавить сотрудника: обязательные поля – Фамилия, Имя, Должность, если отсутствует одно из них – возвращается ошибка. После добавления сотрудника в базу, вернуть объект нового сотрудника в формате JSON.
 2. Изменить сотрудника: обязательные поля – Id сотрудника, которого мы меняем, если Id сотрудника не указан или сотрудник не найден в базе – возвращается ошибка. При успешном изменении, вернуть сотрудника в формате JSON.
 3. Удалить сотрудника: обязательные поля – Id сотрудника, которого мы меняем, если Id сотрудника не указан или сотрудник не найден в базе – возвращается ошибка.
 4. Получить список всех сотрудников: возвращает список всех сотрудников в формате JSON. Опциональный аргумент – должность, если он передан, возвращаются **только** сотрудники с указанной должностью. Если передается несуществующая должность – вернуть ошибку
 5. Получить список всех должностей.

2. (Бонус) если все-таки решили делать статистику нарушений, метод из пункта d должен показывать её в списке сотрудников.

Еще раз список используемых технологий:

1. NET 5 <https://dotnet.microsoft.com/download/dotnet/5.0>
2. ASP.NET Core 5.0 Web Api, <https://docs.microsoft.com/ru-ru/aspnet/core/tutorials/first-web-api?view=aspnetcore-5.0&tabs=visual-studio>, <https://docs.microsoft.com/en-us/learn/modules/build-web-api-aspnet-core/>
3. EntityFramework Core 5.0 <https://docs.microsoft.com/en-us/ef/core/> , <https://docs.microsoft.com/en-us/ef/core/get-started/overview/first-app?tabs=netcore-cli>
4. EF Core Sqlite <https://entityframeworkcore.com/providers-sqlite>