

Домашнее задание 10

Реализовать бэкэнд (серверную часть веб-приложения, web-api) для восьмого домашнего задания с использованием фреймворка ASP .NET Core. При реализации использовать минимальные api, а не реализацию с классами-контроллерами. Бэкэнд должен содержать следующие конечные точки:

- /comments
- /comments/{id}

Обработка http методов должна происходить по следующим правилам:

- запрос GET к /comments должен возвращать массив комментариев;
- запрос GET к /comments/{id} (например /comments/1) должен возвращать комментарий с указанным идентификатором (id);
- запрос POST к /comments должен добавлять новый комментарий в базу данных сервера (объект комментарий содержится в теле запроса);
- запрос PATCH к /comments/{id} должен изменять комментарий с указанным идентификатором (id); т.е., если в базу данных есть комментарий с указанным идентификатором, то его поля меняются на значения полей, которые указаны в теле запроса;
- запрос DELETE к /comments/{id} должен удалять комментарий с указанным id.

При реализации бэкэнда использовать слоистую архитектуру. Для этого реализуйте следующие слои:

- Слой данных куда входят модель и репозиторий. Модель будет состоять из одного класса Comment, репозиторий должен содержать интерфейс ICommentRepository, который описывает все необходимые для работы с данными, операции, и класса CommentRepository, который реализует интерфейс ICommentRepository. База данных к которой обращается репозиторий представляет собой объект Dictionary<int, Comment> внутри CommentRepository.
- Сервисный слой в котором находится класс CommentService. Сервис нужно реализовать даже в том случае, если он просто делегирует всю работу репозиторию и не выполняет самостоятельных вычислений.
- Слой контроллеров — слой с маршрутизацией и указанием обработчиков маршрутов.

Для проверки работоспособности используйте React приложение, которое реализовано в рамках восьмой домашней работы.