

Практическая работа №4

Разработка веб-клиента

1 Цель работы

- 1.1 Научиться разрабатывать клиентское веб-приложение для доступа к БД.
- 1.2 Научиться проверять работоспособность RESTful API в клиентском приложении.

2 Литература

- 2.1 <https://metanit.com/sharp/aspnet6/12.2.php> – клиент JS
- 2.2 <https://learn.microsoft.com/ru-ru/aspnet/core/data/ef-rp/intro?view=aspnetcore-9.0&tabs=visual-studio> — клиент ASP.Net (Razor Pages)

3 Задание

3.1 Разработать html-страницу, отображающую данные, полученные по API, в виде таблицы или списка при загрузке страницы.

В качестве источника можно использовать свой API или одного из публичных API, например:

- 1) <https://jsonplaceholder.typicode.com/> (например: <https://jsonplaceholder.typicode.com/todos>)
документация: <https://jsonplaceholder.typicode.com/guide/>
- 2) <https://dummyjson.com/> (например: <https://dummyjson.com/products?limit=10&skip=0&select=title,price>)
документация по использованию: <https://dummyjson.com/docs/products>

3.2 Для каждого элемента добавить кнопку «Удалить» и реализовать ее функциональность. После удаления записи данные на странице должны обновляться.

3.3 Добавить на страницу элементы для ввода данных и кнопку «Добавить», реализовать ее функциональность. После добавления записи данные на странице должны обновляться.

3.4 Для каждого элемента добавить кнопку «Отредактировать» и реализовать ее функциональность. После редактирования записи данные на странице должны обновляться.

3.5 Разработать веб-приложение ASP.Net, используя шаблон Razor Pages.

4 Контрольные вопросы

- 4.1 Как выполнить HTTP-запрос к API с помощью Fetch API?
- 4.2 Как обрабатывать ошибки при вызове API в JavaScript?
- 4.3 Что такое асинхронные функции и как они связаны с вызовом API?
- 4.4 Что такое Razor Pages?
- 4.5 Как создать и настроить новую страницу Razor в ASP.NET Core?