

Практическая работа №2

Вызов методов REST API

1 Цель работы

- 1.1 Научиться проверять работоспособность RESTful API в клиентском приложении.
- 1.2 Научиться выполнять тестирование RESTful API методом черного ящика.

2 Литература

- 2.1 <https://metanit.com/sharp/net/2.7.php>
- 2.2 <https://learn.microsoft.com/ru-ru/dotnet/csharp/tutorials/console-webapiclient>
- 2.3 <https://learn.microsoft.com/ru-ru/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-client>

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см.п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

Для выполнения тестирования должен быть запущен проект Web-API (ПР №1).

5.1 Создание сервисного слоя и клиентского приложения

5.1.1 Создать библиотеку классов, в которую добавить:

- класс модели данных с открытыми автореализуемыми свойствами на чтение и запись для работы с объектами, отображающими данные таблицы фильмов,
- сервис для работы с API, в котором настроить поля HttpClient и базовый адрес.

5.1.2 Создать оконное приложение .NET с интерфейсом для отображения:

- информации о фильмах в виде таблицы,
- информации о фильме в полях ввода.

5.2 Получение данных

5.2.1 Создать в сервисе асинхронный метод, возвращающий список объектов, используя API (метод GET).

Вызвать созданный асинхронный метод в клиентском приложении и вывести информацию о полученных объектах на экран.

5.2.2 Создать в сервисе асинхронный метод, возвращающий объект по идентификатору, используя API (метод GET с параметром).

Вызвать созданный асинхронный метод в клиентском приложении и вывести информацию о полученном объекте на экран.

Если объект с указанным идентификатором не найден, сообщить об этом.

5.3 Вставка данных

Создать в сервисе асинхронный метод, создающий объект в таблице и возвращающий его адрес, используя API (метод POST).

Создать в клиентском приложении объект класса, вызвать созданный асинхронный метод, передав в параметрах созданный объект.

Вывести на экран адрес страницы созданного объекта (если запись добавлена) и

сообщение, удалось ли добавить запись.

Проверить, что в БД добавлена новая строка с данными, полученными из клиентского приложения.

5.4 Удаление данных

Создать асинхронный метод, удаляющий объект из таблицы по идентификатору, используя API (метод DELETE).

Вызвать созданный асинхронный метод в клиентском приложении. Вывести на экран код http-ответа и сообщение, удалось ли удалить запись.

Проверить, что в БД удалена строка с указанным в клиентском приложении идентификатором.

5.5 Изменение данных

Создать асинхронный метод, изменяющий переданный в параметрах объект в таблице по его идентификатору, используя API (метод PUT).

Для тестирования:

- получить объект по id,
- изменить в клиентском приложении значение одного из свойств полученного объекта класса,
- вызвать созданный асинхронный метод, передав в параметрах созданный объект.

Вывести на экран данные обновленного объекта и сообщение, удалось ли изменить запись.

Проверить, что в БД данные строки изменены на данные, полученные из клиентского приложения.

6 Порядок выполнения работы

6.1 Выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Как указать у объекта HttpClient базовый адрес?

8.2 Какие коды ответа могут быть получены при Http-запросе (указать коды и значения)?

8.3 Какой метод класса HttpClient используется для получения данных?

8.4 Какой метод класса HttpClient используется для вставки данных?

8.5 Какой метод класса HttpClient используется для изменения данных?

8.6 Какой метод класса HttpClient используется для удаления данных?