**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по лабораторным и практическим работам**

**МДК 11.01**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Студент | ИСПП-35 | |  |  | |  | | --- | | *Пожидаев Г.С.*  *Пожидаев Г.С* | |
|  | (Группа) | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Преподаватель | | |  |  | *Маломан Ю. С.* |
|  | |  | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2024

# **Практическая работа №2**

**Вызов REST API**

1. **Цель работы** 
   1. Научиться проверять работоспособность RESTful API в клиентском приложении.
   2. Научиться выполнять тестирование RESTful API методом черного ящика.
2. **Контрольные вопросы**

Вопрос: Как указать у объекта HttpClient базовый адрес?

Для указания базового адреса можно использовать свойство `BaseAddress`:

HttpClient client = new HttpClient();

client.BaseAddress = new Uri("https://api.example.com");

Альтернативно, можно передать адрес при создании экземпляра:

HttpClient client = new HttpClient("https://api.example.com");

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

Вот основные коды ответа HTTP и их значения [1]:

1. 200 OK - запрос успешно обработан

2. 201 Created - ресурс создан

3. 202 Accepted - запрос принят для обработки

4. 204 No Content - запрос успешно обработан, но без содержимого

5. 301 Moved Permanently - ресурс перемещен permanently

6. 302 Found - временное перенаправление

7. 304 Not Modified - ресурс не изменился

8. 400 Bad Request - неправильный запрос

9. 401 Unauthorized - аутентификация требуется

10. 403 Forbidden - доступ запрещен

11. 404 Not Found - ресурс не найден

12. 405 Method Not Allowed - метод недопустим для этого ресурса

13. 500 Internal Server Error - внутренняя ошибка сервера

14. 501 Not Implemented - сервер не реализует функцию

15. 502 Bad Gateway - неверная внешняя связь

16. 503 Service Unavailable - сервис недоступен

17. 504 Gateway Timeout - ожидание ответа от интермедиарного сервера

Вопрос: Какой метод класса HttpClient используется для получения данных?

Метод `GetAsync()` используется для получения данных [1]:

var response = await client.GetAsync("api/resource");

Вопрос: Какой метод класса HttpClient используется для вставки данных?

Метод `PostAsync()` используется для отправки POST-запросов с данными [1]:

var content = new StringContent(JsonConvert.SerializeObject(data), Encoding.UTF8, "application/json");

var response = await client.PostAsync("api/resource", content);

Вопрос: Какой метод класса HttpClient используется для изменения данных?

Метод `PutAsync()` используется для обновления существующего ресурса [1]:

var content = new StringContent(JsonConvert.SerializeObject(data), Encoding.UTF8, "application/json");

var response = await client.PutAsync("api/resource", content);

Вопрос: Какой метод класса HttpClient используется для удаления данных?

Метод `DeleteAsync()` используется для удаления ресурса [1]:

var response = await client.DeleteAsync("api/resource");

При работе с асинхронными операциями важно использовать ключевое слово `async` и `await`:

try

{

var response = await client.GetAsync("api/resource");

response.EnsureSuccessStatusCode();

string responseBody = await response.Content.ReadAsStringAsync();

}

catch (HttpRequestException e)

{

Console.WriteLine($"Error: {e.Message}");

}

1. **Вывод**
   1. В ходе лабораторной работы мы научились проверять работоспособность RESTful API в клиентском приложении.
   2. Научились выполнять тестирование RESTful API методом черного ящика.