Caso get:

using System;

using System.Collections.Generic;

using System.Net.Http;

using System.Threading.Tasks;

using Newtonsoft.Json;

namespace TuNamespace

{

public class DataService

{

private readonly HttpClient \_httpClient;

public DataService()

{

\_httpClient = new HttpClient();

}

public async Task<List<User>> GetUsersAsync(string url)

{

try

{

// Envía la solicitud GET

var response = await \_httpClient.GetAsync(url);

if (response.IsSuccessStatusCode)

{

// Lee y deserializa la respuesta JSON en una lista de objetos User

var jsonResponse = await response.Content.ReadAsStringAsync();

var users = JsonConvert.DeserializeObject<List<User>>(jsonResponse);

return users;

}

else

{

Console.WriteLine($"Error: {response.StatusCode}");

return null;

}

}

catch (Exception ex)

{

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

return null;

}

}

}

// Clase que representa a un usuario

public class User

{

public int Id { get; set; }

public string Name { get; set; }

public string Email { get; set; }

}

}

Uso del get:  
public class MainPageViewModel

{

private readonly DataService \_dataService;

public MainPageViewModel()

{

\_dataService = new DataService();

}

public async void LoadUsers()

{

string apiUrl = "https://api.example.com/users";

var users = await \_dataService.GetUsersAsync(apiUrl);

if (users != null)

{

foreach (var user in users)

{

Console.WriteLine($"User: {user.Name}, Email: {user.Email}");

}

}

else

{

Console.WriteLine("Error al cargar los usuarios.");

}

}

}

Caso post:

using System;

using System.Net.Http;

using System.Text;

using System.Threading.Tasks;

using Newtonsoft.Json;

namespace TuNamespace

{

public class DataService

{

private readonly HttpClient \_httpClient;

public DataService()

{

\_httpClient = new HttpClient();

}

public async Task<UserResponse> CreateUserAsync(string url, User user)

{

try

{

// Serializa el objeto User a JSON

var jsonData = JsonConvert.SerializeObject(user);

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

// Envía la solicitud POST con los datos JSON

var response = await \_httpClient.PostAsync(url, content);

if (response.IsSuccessStatusCode)

{

// Lee y deserializa la respuesta JSON en un objeto UserResponse

var jsonResponse = await response.Content.ReadAsStringAsync();

var result = JsonConvert.DeserializeObject<UserResponse>(jsonResponse);

return result;

}

else

{

Console.WriteLine($"Error: {response.StatusCode}");

return null;

}

}

catch (Exception ex)

{

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

return null;

}

}

}

// Clase que representa un usuario

public class User

{

public string Name { get; set; }

public string Email { get; set; }

public string Password { get; set; }

}

// Clase que representa la respuesta al crear un usuario

public class UserResponse

{

public int Id { get; set; }

public string Name { get; set; }

public string Email { get; set; }

}

}

Uso del post:  
public class MainPageViewModel

{

private readonly DataService \_dataService;

public MainPageViewModel()

{

\_dataService = new DataService();

}

public async void CreateNewUser()

{

string apiUrl = "https://api.example.com/users";

var newUser = new User

{

Name = "John Doe",

Email = "john.doe@example.com",

Password = "securepassword123"

};

var response = await \_dataService.CreateUserAsync(apiUrl, newUser);

if (response != null)

{

Console.WriteLine($"User created: {response.Name}, Email: {response.Email}");

}

else

{

Console.WriteLine("Error al crear el usuario.");

}

}

}