Endpoints:

GET,POST,PUT,DELETE

para consumir el endpoint y traer todos los clientes

Ejemplo desde .net core:

Pasamos la dirección de la api a nustro cliente http y deserializamos el contenido JSON para pasarlo a nuesto viewmodel previamente hecho

try

{

// URL base de la API

var baseUrl = $"{apiurl}api/customer";

// Realiza la solicitud GET a la URL construida

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

response.EnsureSuccessStatusCode();

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

// Deserializa el JSON a tu modelo de vista

var customers = JsonSerializer.Deserialize<List<CustomerViewModel>>(

content,

new JsonSerializerOptions { PropertyNameCaseInsensitive = true }

);

return customers;

}

catch (Exception)

{

return new List<CustomerViewModel>();

}

Para la creación necesitamos los datos que convertiremos a formato json y enviarlo mediante el cliente especificando que sea un post:

try

{

// URL base de la API

var baseUrl = $"{apiurl}api/customer";

var json = JsonSerializer.Serialize(new { customer.Name, customer.Email }, \_jsonOptions);

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

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

return response.IsSuccessStatusCode;

}

catch (Exception)

{

return false;

}

Para la actualizar necesitamos los datos que convertiremos a formato json y enviarlo mediante el cliente especificando que sea un put muy similar al post:

try

{

// URL base de la API

var baseUrl = $"{apiurl}api/customer";

var json = JsonSerializer.Serialize(new { customer.Name, customer.Email }, \_jsonOptions);

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

var response = await \_httpClient.PutAsync($"{baseUrl}/{customer.Id}", content);

return response.IsSuccessStatusCode;

}

catch (Exception)

{

return false;

}

Para eliminar necesitamos enviarle el id del elemento a eliminar y especicicando que será un delete en el endpoint:

try

{

// URL base de la API

var baseUrl = $"{apiurl}api/customer";

var response = await \_httpClient.DeleteAsync($"{baseUrl}/{id}");

return response.IsSuccessStatusCode;

}

catch (Exception)

{

return false;

}