

Dependencies

PM> Install-Package Newtonsoft.Json

```
using Newtonsoft.Json.Linq;  
using System.Net.Http;
```

Main Asynchronous Task Example

```
await Console.Out.WriteLineAsync("GET request: ");  
  
Answer todos = await Get.Request(  
    "http://localhost:3000/todo",  
    new Dictionary<string, string>() {  
        { "User-Agent", "MyTestApplication/1.0.0" }  
    });  
  
await Console.Out.WriteLineAsync(todos.InternalError ? todos.InternalErrorMessage  
    : "No internal errors!");  
  
if (!todos.InternalError)  
{  
    await Console.Out.WriteLineAsync("\nServer sent headers:\n" +  
        todos.Response?.Headers.ToString());  
  
    await Console.Out.WriteLineAsync("\nStatusCode: " +  
        todos.Response?.StatusCode + " (" + (int)todos.Response?.StatusCode + ")");  
  
    await Console.Out.WriteLineAsync("\nResult:\n" + todos.Result);  
}
```

Answer Class

```
public class Answer  
{  
    public string InternalErrorMessage { get; set; } = string.Empty;  
    public bool InternalError {  
        get {  
            return !InternalErrorMessage.Equals(string.Empty);  
        }  
    }  
    public HttpResponseMessage? Response { get; set; }  
    public JObject? Result { get; set; }  
}
```

GET Method (Base)

```
public class Get
{
    public static async Task<Answer> Request(string apiUrl, Dictionary<string,
string> headers)
    {
        Answer answer = new Answer();
        using (HttpClient client = new HttpClient())
        {
            try
            {
                foreach (var item in headers)
                {
                    client.DefaultRequestHeaders.Add(item.Key, item.Value);
                }
                answer.Response = await client.GetAsync(apiUrl);
                answer.Result = JObject.Parse(
                    await answer.Response.Content.ReadAsStringAsync());
            }
            catch (HttpRequestException rex)
            {
                answer.InternalErrorMessage = rex.Message;
            }
        }
        return answer;
    }
}
```

POST Method (Base)

```
public class Post
{
    public static async Task<Answer> Request(string apiUrl, Dictionary<string,
string> headers, object parameters)
    {
        Answer answer = new Answer();
        using (HttpClient client = new HttpClient())
        {
            try
            {
                foreach (var item in headers)
                {
                    client.DefaultRequestHeaders.Add(item.Key, item.Value);
                }
                StringContent content = new
StringContent(JObject.FromObject(parameters).ToString(), Encoding.UTF8,
"application/json");
                answer.Response = await client.PostAsync(apiUrl, content);
                answer.Result = JObject.Parse(await
answer.Response.Content.ReadAsStringAsync());
            }
            catch (HttpRequestException rex)
            {
                answer.InternalErrorMessage = rex.Message;
            }
        }
        return answer;
    }
}
```

GET method with using DeserializeObject

```
using System;
using System.Net.Http;
using System.Threading.Tasks;
using Newtonsoft.Json;

class Program
{
    static async Task Main(string[] args)
    {
        await GetApiData();
    }

    static async Task GetApiData()
    {
        string apiUrl = "http://localhost:3000/todo";

        using (HttpClient client = new HttpClient())
        {
            try
            {
                HttpResponseMessage response = await client.GetAsync(apiUrl);

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

                var parsedData =
                    JsonConvert.DeserializeObject<dynamic>(responseBody);

                Console.WriteLine(parsedData);
            }
            catch (HttpRequestException e)
            {
                Console.WriteLine($"Request execution failed!More: {e.Message}");
            }
        }
    }
}
```