public enum HttpMethod

{

Post,

Get

}

public class Result

{

public bool Error { get; set; }

public string Message { get; set; }

public string Data { get; set; }

}

public Result AjaxCall(HttpMethod Method, string url, object data = null)

{

Result result = new Result();

try

{

using (HttpClient client = new HttpClient())

{

client.BaseAddress = new Uri(ApiHost);

//这里设置的是返回接收的数据格式，并不是发送的数据格式

client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

//client.DefaultRequestHeaders.TryAddWithoutValidation("Content-Type", "application/json; charset=utf-8");

//client.DefaultRequestHeaders.Add("Content-Type", "application/json");

client.DefaultRequestHeaders.AcceptCharset.Add(new StringWithQualityHeaderValue("utf-8"));

if (!string.IsNullOrWhiteSpace(AccessToken))

client.DefaultRequestHeaders.Add("Authorization", $"Bearer {AccessToken}");

var jsonStr = JsonConvert.SerializeObject(data);

//这里才是设置发送数据格式的地方，两种方法都可以

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

var content = new StringContent(jsonStr);

content.Headers.ContentType = new MediaTypeHeaderValue("application/json");

HttpResponseMessage response = new HttpResponseMessage();

switch (Method)

{

case HttpMethod.Post:

response = client.PostAsync(url, content).Result;

break;

case HttpMethod.Get:

response = client.GetAsync(url).Result;

break;

}

if (response.IsSuccessStatusCode)

{

result.Error = false;

result.Message = "OK";

result.Data = response.Content.ReadAsStringAsync().Result;

}

else

{

result.Error = true;

result.Message = $"Http Request Status: {response.StatusCode} –

{response.Content.ReadAsStringAsync().Result}";

result.Data = response.Content.ReadAsStringAsync().Result;

}

}

}

catch(Exception err)

{

result.Error = true;

result.Message = $"Error: {err.Message}\n\nMessage: {err.StackTrace}";

result.Data = "";

}

return result;

}

使用JsonConvert 转化为不同类型

public T toObject<T>(Result result) {

if(result.Error)

{

return default(T);

}

else

{

return JsonConvert.DeserializeObject<T>(result.Data);

}

}

public IList<T> toList<T>(Result result)

{

if (result.Error)

{

return new List<T>();

}

else

{

return JsonConvert.DeserializeObject<IList<T>>(result.Data);

}

}

HttpClient 跳过SSL 加密验证：

using System.Net;

ServicePointManager.ServerCertificateValidationCallback += (sender, cert, chain, sslPolicyErrors) => true;