# Web-API-in-.NET in C#

On GitHub: <https://github.com/tlilov>

How to consume Web API using HttpClient in the console application.

### Run

Step 1. Create service class that points to the Web API resource: EmailService.cs

private readonly JsonServiceClient \_client;

public EmailService(string siteCode)

{

\_client = new JsonServiceClient(WebConnectionUriBuilder.GetFor(siteCode), WebConnectionUriBuilder.GetApiKey());

}

public AddContactResponse AddContact(AddContactRequest request)

{

return \_client.PostResult<AddContactRequest, AddContactResponse>("api/email/addcontact", request);

}

Step 2. Create a generic HttpClient helper to execute the HttpClient POST or GET methods: JasonServiceClient.cs

public TResult PostResult<TRequest, TResult>(string url, TRequest request)

{

TResult result;

using (var client = new HttpClient())

{

client.BaseAddress = new Uri(\_baseUrl);

client.DefaultRequestHeaders.Add("Api-Key", \_apiKey);

client.DefaultRequestHeaders.Accept.Clear();

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

var response = client.PostAsJsonAsync(FixUrl(url), request).Result;

result = response.IsSuccessStatusCode ? response.Content.ReadAsAsync<TResult>().Result : default(TResult);

}

return result;

}

Step 3. Set up a client that consumes the Web API resource in EmailService from step 1: Client.cs

try

{

var service = new Web.Api.Client.Services.EmailService(primarySiteCode, company);

var response = service.GetContact(new GetContactRequest(){Email = email});

if (response.Contact != null)

{

service.AddContact(new AddContactRequest() { Contact = new ContactView() {Id = response.Contact.Id, Email = email, ContactStatus = contactStatus }});

}

else

{

service.AddContact(new AddContactRequest() { Contact = new ContactView() { SiteCode = primarySiteCode, Email = email, ContactStatus = contactStatus, GUID = guid } });

}

}