

C#

- [C-Sharp and API-NG](#)
- [How To Run](#)
- [Code Snippet](#)
 - [Json-Rpc](#)
 - [Rescript](#)
 - [Example usage of the code above](#)

C-Sharp and API-NG

This page provides a guide on how to communicate with API-NG using C#, and some code snippets showing its purpose. The C-sharp code is written against .Net 4, and is a simple Console application.

In the sample code we use the Json-rpc and rescript protocol. All requests are sent and received using json format. To post a request we prepare the json string using C# object and then we serialize the object using the [Json .Net library](#)

This documentation refers to the code available at <https://github.com/betfair/API-NG-sample-code/tree/master/cSharp>



Expect- 100 Continue Header

Please note: When using .NET you set the "Expect -100Continue" property to false. This property sits in the [ServicePointManager](#) class

How To Run

Provided that you have .net 4 installed, and the .net related dll's sit in the location : C:\Program Files\Reference Assemblies\Microsoft\Framework\NETFramework\v4.0\Profile\Client\

The sample takes three arguments the app key and session token are mandatory. the third argument is to do with which client you want to use rescript or jsonrpc, if nothing is passed then its defaulted to jsonrpc.

Then it is a simple case of opening up command line and executing:

```
<Path To Cloned Repository>\Api-ng-sample-code\Api-ng-sample-code\bin\Release\Api-ng-sample-code.exe <appkey>  
<sessontoken> <(optional)rescript/jsonrpc>
```

Code Snippet

Json-Rpc

This is the main, brain code snippet of the json-rpc calls all methods go through here, initially we create a request object, which contains necessary headers. the appkey and session token are in the custom headers which gets instantiated when the client is instantiated., the invoke methods actually serializes the request objects makes the request, and upon receiving the response, de-serialize it into the response object specified as the T using generics.

```

protected WebRequest CreateWebRequest(Uri uri)
{
    WebRequest request = WebRequest.Create(new Uri(EndPoint));
    request.Method = "POST";
    request.ContentType = "application/json-rpc";
    request.Headers.Add(HttpRequestHeader.AcceptCharset, "ISO-8859-1,utf-8");
    request.Headers.Add(CustomHeaders);
    return request;
}

public T Invoke<T>(string method, IDictionary<string, object> args = null)
{
    if (method == null)
        throw new ArgumentNullException("method");
    if (method.Length == 0)
        throw new ArgumentException(null, "method");

    var request = CreateWebRequest(new Uri(EndPoint));

    using (Stream stream = request.GetRequestStream())
    using (StreamWriter writer = new StreamWriter(stream, Encoding.UTF8))
    {
        var call = new JsonRequest { Method = method, Id = 1, Params = args };
        JsonConvert.Export(call, writer);
    }
    Console.WriteLine("Calling: " + method + " With args: " + JsonConvert.
Serialize<IDictionary<string, object>>(args));

    using (WebResponse response = GetWebResponse(request))
    using (Stream stream = response.GetResponseStream())
    using (StreamReader reader = new StreamReader(stream, Encoding.UTF8))
    {
        var jsonResponse = JsonConvert.Import<T>(reader);
        Console.WriteLine("Got Response: " + JsonConvert.Serialize<JsonResponse<T>>(jsonResponse));
        if (jsonResponse.HasError)
        {
            throw ReconstituteException(jsonResponse.Error);
        }
        else
        {
            return jsonResponse.Result;
        }
    }
}

private static System.Exception ReconstituteException(Api_ng_sample_code.TO.Exception ex)
{
    var data = ex.Data;

    // API-NG exception -- it must have "data" element to tell us which exception
    var exceptionName = data.Property("exceptionname").Value.ToString();
    var exceptionData = data.Property(exceptionName).Value.ToString();
    return JsonConvert.Deserialize<APINGException>(exceptionData);
}
}
}

```

Example usage of the code above is:

```
public IList<EventTypeResult> listEventTypes(MarketFilter marketFilter, string locale = null)
{
    var args = new Dictionary<string, object>();
    args[FILTER] = marketFilter;
    args[LOCALE] = locale;
    return Invoke<List<EventTypeResult>>(LIST_EVENT_TYPES_METHOD, args);
}
```

Rescript

Below is the rescript implementation of the functionality mentioned above

```

protected HttpRequest CreateWebRequest(String restEndPoint)
{
    HttpRequest request = (HttpRequest)WebRequest.Create(restEndPoint);
    request.Method = "POST";
    request.ContentType = "application/json";
    request.ContentLength = 0;
    request.Headers.Add(HttpRequestHeader.AcceptCharset, "UTF-8");
    request.Accept = "application/json";
    request.Headers.Add(CustomHeaders);
    return request;
}

public T Invoke<T>(string method, IDictionary<string, object> args = null)
{
    if (method == null)
        throw new ArgumentNullException("method");
    if (method.Length == 0)
        throw new ArgumentException(null, "method");

    var restEndpoint = EndPoint + method + "/";
    var request = CreateWebRequest(restEndpoint);

    var postData = JsonConvert.Serialize<IDictionary<string, object>>(args) + "}";
    Console.WriteLine("Calling: " + method + " With args: " + postData);

    var bytes = Encoding.GetEncoding("UTF-8").GetBytes(postData);
    request.ContentLength = bytes.Length;

    using (Stream stream = request.GetRequestStream())
    {
        stream.Write(bytes, 0, bytes.Length);
    }
    using (WebResponse response = GetWebResponse(request))

    using (Stream stream = response.GetResponseStream())
    using (StreamReader reader = new StreamReader(stream, Encoding.UTF8))
    {
        var jsonResponse = reader.ReadToEnd();
        Console.WriteLine("Got response: " + jsonResponse);
        if (jsonResponse.Contains("exception")) {
            throw ReconstituteException(JsonConvert.Deserialize<Api_ng_sample_code.TO.Exception>
(jsonResponse));
        }
        return JsonConvert.Deserialize<T>(jsonResponse);
    }
}

private static System.Exception ReconstituteException(Api_ng_sample_code.TO.Exception ex)
{
    var data = ex.Detail;

    // API-NG exception -- it must have "data" element to tell us which exception
    var exceptionName = data.Property("exceptionname").Value.ToString();
    var exceptionData = data.Property(exceptionName).Value.ToString();
    return JsonConvert.Deserialize<APINGException>(exceptionData);
}

```

and the example usage

```
public IList<EventTypeResult> listEventTypes(MarketFilter marketFilter, string locale = null)
{
    var args = new Dictionary<string, object>();
    args[FILTER] = marketFilter;
    args[LOCALE] = locale;
    return Invoke<List<EventTypeResult>>(LIST_EVENT_TYPES_METHOD, args);
}
```

Example usage of the code above

```
IClient client = null;
string clientType = null;
if (args.Length == 3)
{
    clientType = args[2];
}
// if rescript has been passed as the third argument use it otherwise default to json client
if (!string.IsNullOrEmpty(clientType) && clientType.Equals("rescript"))
{
    Console.WriteLine("Using RescriptClient");
    client = new RescriptClient(Url, appkey, sessionToken);
}
else
{
    Console.WriteLine("Using JsonRpcClient");
    client = new JsonRpcClient(Url, appkey, sessionToken);
}
Console.WriteLine("\nBeginning sample run!\n");
var marketFilter = new MarketFilter();
marketFilter.TextQuery = "Horse Racing";

var eventTypes = client.listEventTypes(marketFilter);
```

- [C-Sharp and API-NG](#)
- [How To Run](#)
- [Code Snippet](#)
 - [Json-Rpc](#)
 - [Rescript](#)
 - [Example usage of the code above](#)