Writing C# Http Rest Client

# Intorduction

We can easily create rest application in C# that communicate with node js server, MVC web server , WCF http server.

To do so, let’s use the rest client I took from code project

(added credentials only)

Credits to: <https://www.codeproject.com/Tips/497123/How-to-make-REST-requests-with-Csharp>

The restClient.cs is located at : DriversProjects\RestClient.cs

To use the rest client we can create function like the following:

public List<Algorithems> getAlgorithems(int testid)

{

lock (m\_lock)

{

try

{

string parameters = "?type=" + testid;

m\_restClient.EndPoint = m\_url + "/getAlgorithems ";

m\_restClient.Method = HttpVerb.GET;

m\_restClient.PostData = null;

var json = m\_restClient.MakeRequest(parameters);

dynamic d = JObject.Parse(json);

string r = d.Result;

List< Algorithems > l = JsonConvert.DeserializeObject<List<Algorithems>>(r);

return l;

}

catch (Exception err)

{

throw (new SystemException(err.Message));

}

}

}

Algorithems is just a struct that we can read and write using the jsonSerialzer.

We can do GET, POST

public void UpdatePictureFileName(string Name, string filename, int picnum)

{

lock (m\_lock)

{

try

{

string parameters = string.Format(@"?dishName={0}&filename={1}&picnum={2}", dishName, filename, picnum);

m\_restClient.EndPoint = m\_url + "/UpdateDishPictureFileName";

m\_restClient.Method = HttpVerb.POST;

m\_restClient.PostData = null;

var json = m\_restClient.MakeRequest(parameters);

dynamic d = JObject.Parse(json);

string r = d.Result;

}

catch (Exception err)

{

throw (new SystemException(err.Message));

}

}

}

You can communicate also with Mongoose light C Web Server

<https://www.cesanta.com/products/binary>

<https://github.com/cesanta/mongoose>

you can see a HTTP WCF Server I build that can be call using the rest client class