Microsoft.WindowsAzure.MobileServices, MobileServiceClient

reading/writing into table of MobileService, also demonstrates async, await

nuget: <package id="WindowsAzure.MobileServices" version="1.3.2" targetFramework="net46" />

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Microsoft.WindowsAzure.MobileServices;
using Newtonsoft.Json.Linq;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
class ReadingWritingToAzureMobileService
{
// main cannot be marked as async
static void Main(string[] args)
{
MobileServiceClient msc = new MobileServiceClient("https://baracuda.azure-mobile.net/", "CjvMxVJcepqVrhnrFcMUzBkFZYcuhK76");
IMobileServiceTable table1 = msc.GetTable("table1");
```

```
// http://www.newtonsoft.com/json/help/html/CreatingLINQtoJSON.htm
// http://www.dotnetperls.com/async
// this doesn't work, because "Main" is not a async method: await X
// instead we need to wait on returned task-object
// Task t = Insert50Records(table1);
// t.Wait();
Task t = UpdateRecord(table1, "8B98E279-1652-42F4-B4B5-09A7675AAD8B", "neuer text");
t.Wait();
Task t2 = FindRecords(table1);
t2.Wait();
}
private static async Task<int> FindRecords(IMobileServiceTable table1)
{
// returns all records !!!
JArray token1 = (JArray)await table1.ReadAsync("select ssss id,text,completed,when from table1 where text='text changed 234'");
return 20;
}
private static async Task UpdateRecord(IMobileServiceTable table1, string id, string text)
{
JObject token1 = (JObject)await table1.LookupAsync(id);
string name = token1.GetType().Name;
token1["text"] = "text changed 234";
JToken token2 = await table1.UpdateAsync(token1);
```

```
}
private static async Task Insert50Records(IMobileServiceTable table1)
{
for(int k=0;k<50;k++)
{
JObject jo = new JObject();
jo.Add("text", new JValue("text " + k));
jo.Add("completed", new JValue(k % 2));
jo.Add("when", new JValue(DateTime.Now.AddDays(k)));
// here we can wait for async method, because we are in a async method
await table1.InsertAsync(jo);
}
}
}
}
```