BAT4 – réseaux **2017-2018**

Using Web Services

In this lab, you will learn how to use Web Services that deliver data as *json* texts. You will use a weather web service and a geographical Web Service. The goal is to be able to collect information that can be useful to manage a building (for example, to regulate the temperature of the building).

First part : using json data

Many Web Services send json data in response to a request from a client. JSON stands for JavaScript Object Notation. It is a syntax for storing and exchanging data. JSON is text, written with JavaScript object notation (see short introduction https://www.w3schools.com/js/js_json_intro.asp).

This first part illustrates how to access information stored into a json description.

1. Download the "json example" C# solution and open it with VS.
2. If the code doesn't compile, add the package NewtonSoft.Json : projet → gérer les packages nugets → rechercher NewtonSoft.Json → installer.
   This package provides predefined tools to transform json data into C# objects. You must install it in all the C# solutions that use json.
3. Read, run and understand the main.
4. Add a line to print the countryName token of the data `timeZoneJson`
5. Change the value of sunrise in the data `timeZoneJson` and check if the method "DayDuration" is still working correctly.
6. Complete the method "PrintJetLag". It must display a message like "coming from Austria, the jetlag is 2 hours". Hint : the GMT offset of France is 2, you just have to subtract France GMT offset and Austria GMT offset (given in `timeZoneJson)`
   Modify `timeZoneJson` changing the country name to Mongolia and the GMT offset to 8. Check if your program still works properly.

Second part : using a webservice

Here you will use 2 webservices that deliver data as json texts.

1. Try one example on https://openweathermap.org/current to observe the json which has been sent by the server.
2. Try the astergdem service on http://www.geonames.org/export/ws-overview.html (json version). If the limit number of demos has been reached, you can use your proper user name or userName=heleneCollavizza instead of demo. Replace the latitude and longitude values in the url with the latitude and longitude values of Nice (or another city of your choice).
3. Download the "weather web service example" C# solution.
4. Read, run and understand the main.
5. Complete the main to connect to the astergdem service on geonames server. The url must be :
   `http://api.geonames.org/astergdemJSON?formatted=true&lat=43.70&lng=7.26&username=heleneCollavizza`
6. Write the elevation sent by the server.
7. Modify the part of the code that connects to openweather by asking to the user the city of its choice (using Console.ReadLine)

8. Go back to your json example solution. Modify the main in order to :
    - connect to the “timezone” service on geonames : it takes a geographical coordinate and returns the time zone.
    - ask the user for the latitude and the longitude.
    - Print the DayDuration and the JetLag. Hint : you must complete the class WeatherInfo in order to store the new information you get via the Web Service (for example, timezoneId is missing in the class WeatherInfo).