TP .Net Maui n° 3 : Utilisation de Web Service

AppShell.xaml

```
<?xml version="1.0" encoding="UTF-8" ?>
<Shell
    x:Class="TPMaui3.AppShell"
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:TPMaui3.Views"
    Shell.FlyoutBehavior="Disabled">

    <TabBar>
        <ShellContent
            Title="Température"
            ContentTemplate="{DataTemplate local:vTemperature}"
            Icon="temperature0.png" />
        <ShellContent
            Title="Météo"
            ContentTemplate="{DataTemplate local:vMeteo}"
            Icon="weather.svg" />
        <ShellContent
            Title="Graphique"
            ContentTemplate="{DataTemplate local:vGraphe}"
            Icon="graphic0.png" />
    </TabBar>
</Shell>
```

vTemperature.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="TPMaui3.Views.vTemperature"
             Title="Température">
    <VerticalStackLayout Spacing="20" Margin="10" >
        <Button
                x:Name="btTemp"
                Text="Afficher"
                Clicked="AfficheTemperature"
                HorizontalOptions="Center" />
        <Label
            Text="Température!"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Large" TextColor="{StaticResource Primary}"/>
        <Label
            Text="?"
            x:Name="lbTemp"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Large" TextColor="{StaticResource Secondary}"/>
    </VerticalStackLayout>
</ContentPage>
```

vTemperature.xaml.cs

```
using System.Diagnostics.Metrics;
using System.Text.Json;
using System.Xml.Serialization;

namespace TPMaui3.Views;

public partial class vTemperature : ContentPage
{
	public vTemperature()
	{
		InitializeComponent();


    }
    private void AfficheTemperature(object sender, EventArgs e)
    {
        LireTemperature();
    }
    private async void LireTemperature()
	{
        Uri uri = new("http://meteorestsrvmobile.lab3il.fr/RestServiceMeteo.svc/xml/1");
            try
            {
            HttpClient client = new();
            HttpResponseMessage response = await client.GetAsync(uri);
            if (response.IsSuccessStatusCode)
            {
                String content = await response.Content.ReadAsStringAsync();
                content = content.Substring(content.IndexOf("<XMLDataResult>") + 15);
                content = content.Substring(0, content.IndexOf("</XMLDataResult>"));
                lbTemp.Text = content + " °C";
            }
        }
            catch (Exception ex)
        {
            await this.DisplayAlert("Error", ex.Message, "OK");
        }
    }
}
```

vMeteo.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="TPMaui3.Views.vMeteo"
             Title="Météo">
    <VerticalStackLayout Spacing="20" Margin="10" >
        <Label
            Text="Météo"
            x:Name="lbMeteo"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Medium" TextColor="{StaticResource Primary}"/>
        <Label
            Text="?"
            x:Name="lbTemperature"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Medium" TextColor="{StaticResource Secondary}"/>
        <Label
            Text="?"
            x:Name="lbPression"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Medium" TextColor="{StaticResource Secondary}"/>
```

```
        <Label
            Text="?"
            x:Name="lbHumidite"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Medium" TextColor="{StaticResource Secondary}"/>
        <Label
            Text="?"
            x:Name="lbPrecipitationJour"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Medium" TextColor="{StaticResource Secondary}"/>
        <Label
            Text="?"
            x:Name="lbTemperatureRessentie"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Medium" TextColor="{StaticResource Secondary}"/>
        <Label
            Text="?"
            x:Name="lbVentDirection"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Medium" TextColor="{StaticResource Secondary}"/>
        <Label
            Text="?"
            x:Name="lbVentVitesse"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Medium" TextColor="{StaticResource Secondary}"/>
    </VerticalStackLayout>
</ContentPage>
```

vMeteo.xaml.cs

```
using System.Diagnostics.Metrics;
using System.Text.Json;
using TPMaui3.Models;

namespace TPMaui3.Views;

public partial class vMeteo : ContentPage
{
    public vMeteo()
    {
        InitializeComponent();
        Meteo();
    }
    private async void Meteo()
    {
        Uri uri = new("http://meteorestsrvmobile.lab3il.fr/RestServiceMeteo.svc/jsontmps");
        try
        {
            HttpClient client = new();
            HttpResponseMessage response = await client.GetAsync(uri);
            JsonSerializerOptions jsonSerializerOptions = new JsonSerializerOptions();
            if (response.IsSuccessStatusCode)
            {
                var result = await response.Content.ReadAsStringAsync();
                result = result.Replace("{\"JSONDataTmpsResult\":", "");
                   result = result.Replace("}}", "}");
                var vMeteo = Newtonsoft.Json.JsonConvert.DeserializeObject<CMeteo>(result);
                lbMeteo.Text = "Relevé à " + vMeteo.DateReleve.ToString();
                lbTemperature.Text = "Temp: " + vMeteo.Temperature.ToString() + " °C";
                lbHumidite.Text = "Hum: " + vMeteo.Humidite.ToString() + " %";
                lbPression.Text = "Pres: " + vMeteo.Pression.ToString() + " hPa";
                lbTemperatureRessentie.Text = "TR: " + vMeteo.TemperatureRessentie.ToString()
                                + " °C";
                lbPrecipitationJour.Text = "PrecJ: " + vMeteo.PrecipitationJour.ToString()
                                + " mm";
                lbVentVitesse.Text = "Vit: " + vMeteo.VentVitesse.ToString() + " m/s";
```

```
                lbVentDirection.Text = "Vent: " + vMeteo.VentDirection.ToString();
            }
        }
        catch (Exception ex)
        {
            await this.DisplayAlert("Error", ex.Message, "OK");
        }
    }
}
```

CMeteo.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Linq.Expressions;
using System.Text;
using System.Text.Json;
using System.Threading.Tasks;
using TPMaui3.Views;

namespace TPMaui3.Models
{
    internal class CMeteo
    {
        public CMeteo()
        {
        }
        public string DateReleve { get; set; }
        public string Temperature { get; set; }
        public string Pression { get; set; }
        public string Humidite { get; set; }
        public string PrecipitationJour { get; set; }
        public string TemperatureRessentie { get; set; }
        public string VentDirection { get; set; }
        public string VentVitesse { get; set; }
    }
}
```

vGraphe.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="TPMaui3.Views.vGraphe"
             xmlns:drawable="clr-namespace:TPMaui3"
             Title="Historique">
    <ContentPage.Resources>
        <drawable:GraphicsDrawable x:Key="MyDrawable" />
    </ContentPage.Resources>
    <VerticalStackLayout Spacing="20" Margin="10">
        <Label
            Text="Historique"
            x:Name="lbTemp"
            VerticalOptions="Center"
            HorizontalOptions="Center" FontSize="Medium" TextColor="{StaticResource Primary}"/>
        <Button
                x:Name="btTemp"
                Text="Afficher"
                Clicked="AfficherGraph"
                HorizontalOptions="Center" />
        <HorizontalStackLayout x:Name="HSL" Spacing="10" Margin="10" HorizontalOptions="Center"
                    HeightRequest="200">
         <Image Source="axis.PNG" />
```

```
         <Label
            Text=""
            x:Name="lb1"
            VerticalOptions="End"
            BackgroundColor="{StaticResource Secondary}"
            WidthRequest="20" HeightRequest="10"/>
        <Label
            Text=""
            x:Name="lb2"
            VerticalOptions="End"
            BackgroundColor="{StaticResource Secondary}"
            WidthRequest="20" HeightRequest="10"/>
        <Label
            Text=""
            x:Name="lb3"
            VerticalOptions="End"
            BackgroundColor="{StaticResource Secondary}"
            WidthRequest="20" HeightRequest="10"/>
        <Label
            Text=""
            x:Name="lb4"
            VerticalOptions="End"
            BackgroundColor="{StaticResource Secondary}"
            WidthRequest="20" HeightRequest="10"/>
        <Label
            Text=""
            x:Name="lb5"
            VerticalOptions="End"
            BackgroundColor="{StaticResource Secondary}"
            WidthRequest="20" HeightRequest="10"/>
        </HorizontalStackLayout>
    </VerticalStackLayout>
</ContentPage>
```

vGraphe.xaml.cs

```
using System.Diagnostics.Metrics;
using System.Drawing;
using System.Linq.Expressions;
using System.Text.Json;
using TPMaui3.Models;

namespace TPMaui3.Views;

public partial class vGraphe : ContentPage
{
    public vGraphe()
      {
            InitializeComponent();

    }

    private async void AfficherGraph(object sender, EventArgs e)
    {
        Uri uri = new Uri("http://meteorestsrvmobile.lab3il.fr/RestServiceMeteo.svc/xmlhisto");

        try
        {
            HttpClient client = new();
            HttpResponseMessage response = await client.GetAsync(uri);
            if (response.IsSuccessStatusCode)
            {
                var result = await response.Content.ReadAsStringAsync();
                double dTaux = HSL.HeightRequest / 20;
                double dTest = ReadValue(result, "TMn0");
                int iTest = (int)dTest;
```

```
                lb1.HeightRequest = (int)(ReadValue(result, "TMn0") * dTaux);
                lb2.HeightRequest = (int)(ReadValue(result, "TMn1") * dTaux);
                lb3.HeightRequest = (int)(ReadValue(result, "TMn2") * dTaux);
                lb4.HeightRequest = (int)(ReadValue(result, "TMn3") * dTaux);
                lb5.HeightRequest = (int)(ReadValue(result, "TMn4") * dTaux);

            }
        }
        catch (Exception ex)
        {
            await this.DisplayAlert("Error", ex.Message, "OK");
        }
    }
    private double ReadValue(String sXml, String sFlag)
    {
        try
        {
            String sFlag1 = "<a:" + sFlag + ">";
            String sFlag2 = "</a:" + sFlag + ">";
            sXml = sXml.Substring(sXml.IndexOf(sFlag1) + sFlag1.Length);
            sXml = sXml.Substring(0, sXml.IndexOf(sFlag2));
            sXml = sXml.Replace(",", ".");
            double dVal = Convert.ToDouble(sXml);
            return dVal;
        }
        catch { }
        return 0;
    }

}
```