Weather Online app for Windows Phone

This code example creates a weather app for Windows Phone 7 using XML data from the World Weather Online service.

Introduction

The example uses the free World Weather Online service to get and display the weather in the user's specified location. We can search for weather based on the get the based on city name, and also specify state (US only) and country (other search options supported by the API are given here). The API also allows us to specify how many days the weather information should return.

The service requires that we register for a key, which is used in our web request. The format of the web request is as shown below (where **xxx** is the key, **yyy** is the city name and **zzz** is the country name):

```
http://free.worldweatheronline.com/feed/weather.ashx?
key=xxx&q=yyy,zzz&num_of_days=3&format=xml
```

Once we get the XML data we parse the content and display on the screen.

Current Weather

Implementation

First lets create a project with *Windows Phone Application* Template. Once the project is being created, lets add a `TextBox` (search box), `Button` and two `ListBox` in the *MainPage.xaml* file.

MainPage.xaml

```
<TextBox Height="72" HorizontalAlignment="Left" Margin="6,76,0,0" Name="locationTextBox"
Text="" VerticalAlignment="Top" Width="345" />
            <Button Content="Go" Height="72" HorizontalAlignment="Right"
Margin="0,76,6,0" Name="goButton" VerticalAlignment="Top" Width="93"
Click="goButton_Click" />
            <ListBox Height="496" HorizontalAlignment="Left" Margin="-4,155,0,0"
Name="listBox1" VerticalAlignment="Top" Width="460" >
                <ListBox.ItemTemplate>
                    <DataTemplate>
                        <StackPanel Orientation="Horizontal" Height="132" >
```


(C) Copyright Nokia 2014. All rights reserved.

```
                        <Grid Height="100">
                            <Grid.ColumnDefinitions>
                                <ColumnDefinition Width="100"></ColumnDefinition>
                                <ColumnDefinition Width="*"></ColumnDefinition>
                            </Grid.ColumnDefinitions>

                            <Image Grid.Column="0" Height="100" Width="100"
Source="{Binding Path=icon}"></Image>
                            <TextBlock Text="{Binding Path=day_of_week}"
Foreground="Orange"  Grid.Column="1" Margin="10,10,10,60"></TextBlock>
                            <TextBlock Grid.Column="1" Margin="10,40,300,30" Text="MinF:
"></TextBlock>
                            <TextBlock Text="{Binding Path=low}" Grid.Column="1"
Margin="70,40,250,30" FontWeight="Bold"></TextBlock>
                            <TextBlock Grid.Column="1" Margin="150,40,160,30"
Text="MaxF: "></TextBlock>
                            <TextBlock Text="{Binding Path=high}" Grid.Column="1"
Margin="215,40,90,30" FontWeight="Bold"></TextBlock>
                            <TextBlock Text="{Binding Path=condition}" Grid.Column="1"
Margin="10,75,10,0"></TextBlock>

                        </Grid>
        </StackPanel>
        </DataTemplate>

                </ListBox.ItemTemplate>
            </ListBox>
```

We parse the XMLs in two sections. One is the **Current Condition** of weather, and another is the information of present day along with the next two days . We use XElement to parse the element and store in the list box.

```
XElement XmlWeather = XElement.Parse(e.Result);
//current
listBoxc.ItemsSource = from weather in XmlWeather.Descendants("current_condition")

select new WeatherInfoCurrent
    {
    observation_time = weather.Element("observation_time").Value,
    temp_C = weather.Element("temp_C").Value,
    temp_F = weather.Element("temp_F").Value,
    icon = weather.Element("weatherIconUrl").Value,
    condition = weather.Element("weatherDesc").Value
    };

//all three days
listBox1.ItemsSource = from weather in XmlWeather.Descendants("weather")

select new WeatherInfo
    {
    day_of_week = weather.Element("date").Value,
    high = weather.Element("tempMaxF").Value,
    low = weather.Element("tempMinF").Value,
    icon = weather.Element("weatherIconUrl").Value,
    condition = weather.Element("weatherDesc").Value
    };
```


(C) Copyright Nokia 2014. All rights reserved.

We create two different classes for storing data in two different list box

WeatherInfo.cs

```
public class WeatherInfo
    {
        public string day_of_week { get; set; }
        public string low { get; set; }
        public string high { get; set; }
        public string icon { get; set; }
        public string condition { get; set; }
    }

public class WeatherInfoCurrent
    {
        public string observation_time { get; set; }
        public string temp_C { get; set; }
        public string temp_F { get; set; }
        public string icon { get; set; }
        public string condition { get; set; }
    }
```

Source Code

The full source code of the example is available here: File:WeatherOnline.zip

(C) Copyright Nokia 2014. All rights reserved.