

```
1 using LiveCharts;
2 using LiveCharts.Wpf;
3 using System;
4 using System.Collections.Generic;
5 using System.Linq;
6 using System.Net.Http;
7 using System.Text.Json;
8 using System.Threading.Tasks;
9 using System.Windows;
10 using System.Windows.Controls;
11
12 namespace WpfApp1
13 {
14     /// <summary>
15     /// Interaction logic for MainWindow.xaml
16     /// </summary>
17     public partial class MainWindow : Window
18     {
19         string url = "https://data.moenv.gov.tw/api/v2/
20             aqx_p_432?api_key=e8dd42e6-9b8b-43f8-991e-
21             b3dee723a52d&limit=1000&sort=ImportDate%
22             20desc&format=JSON";
23         AQIData aqidata = new AQIData();
24         List<Field> fields = new List<Field>();
25         List<Record> records = new List<Record>();
26         List<Record> selectedRecords = new List<Record>();
27         SeriesCollection seriesCollection = new
28             SeriesCollection();
29         public MainWindow()
30         {
31             InitializeComponent();
32             UrlTextBox.Text = url;
33             selectedRecords.Clear();
34         }
35
36         private async void GetWebDataButton_Click(object
37             sender, RoutedEventArgs e)
38         {
39             ContentTextBox.Text = "正在抓取網路資料...";
40
41             string jsonText = await FetchContentAsync(url);
42             ContentTextBox.Text = jsonText;
43             aqidata = JsonSerializer.Deserialize<AQIData>
44                 (jsonText);
45             fields = aqidata.fields.ToList();
46             records = aqidata.records.ToList();
47             selectedRecords = records;
```

```
42      StatusTextBlock.Text = $"共有 {records.Count} 筆資料";
43      DisplayAQIData();
44  }
45
46  private void DisplayAQIData()
47  {
48      RecordDataGrid.ItemsSource = records;
49      DataWrapPanel.Children.Clear();
50
51      foreach (var field in fields)
52      {
53          var propertyInfo = typeof(Record).GetProperty
54              (field.id);
55          if (propertyInfo != null)
56          {
57              string value = propertyInfo.GetValue
58                  (records[0]) as string;
59              if (double.TryParse(value, out double v))
60              {
61                  CheckBox cb = new CheckBox
62                  {
63                      Content = field.info.label,
64                      Tag = field.id,
65                      Margin = new Thickness(3),
66                      Width = 120,
67                      FontSize = 14,
68                      FontWeight = FontWeights.Bold,
69                  };
70                  cb.Checked += UpdateChart;
71                  cb.Unchecked += UpdateChart;
72                  DataWrapPanel.Children.Add(cb);
73              }
74          }
75      }
76
77      private void UpdateChart(object sender,
78          RoutedEventArgs e)
79      {
80          seriesCollection.Clear();
81
82          foreach (CheckBox cb in DataWrapPanel.Children)
83          {
84              if (cb.IsChecked == true)
```

```

85         var tag = cb.Tag as String;
86         ColumnSeries columnSeries = new ColumnSeries();
87         ChartValues<double> values = new ChartValues<double>();
88         List<String> labels = new List<String>();
89
90         foreach (var record in selectedRecords)
91         {
92             var propertyInfo = typeof (Record).GetProperty(tag);
93             if (propertyInfo != null)
94             {
95                 string value = propertyInfo.GetValue(record) as string;
96                 if (double.TryParse(value, out double v))
97                 {
98                     values.Add(v);
99                     labels.Add(record.sitename);
100                 }
101             }
102         }
103         columnSeries.Values = values;
104         columnSeries.Title = tag;
105         columnSeries.LabelPoint = point => $"{labels[(int)point.X]}: {point.Y.ToString()}";
106         seriesCollection.Add(columnSeries);
107     }
108 }
109 AQIChart.Series = seriesCollection;
110 }
111
112 private async Task<string> FetchContentAsync(string url)
113 {
114     try
115     {
116         using (HttpClient client = new HttpClient())
117         {
118             return await client.GetStringAsync(url);
119         }
120     }
121     catch (Exception ex)
122     {
123         return $"發生錯誤: {ex.Message}";

```

```
124     }
125 }
126
127 private void RecordDataGrid_SelectionChanged(object sender, SelectionChangedEventArgs e)
128 {
129     selectedRecords =
130         RecordDataGrid.SelectedItems.Cast<Record>().ToList();
131     StatusTextBlock.Text = $"共選取 {selectedRecords.Count} 筆資料";
132 }
133 private void RecordDataGrid_LoadingRow(object sender, DataGridRowEventArgs e)
134 {
135     e.Row.Header = e.Row.GetIndex() + 1;
136 }
137 }
138 }
139
```