

WPF中轻松生成动态图表：实例详解(MVVM模式)

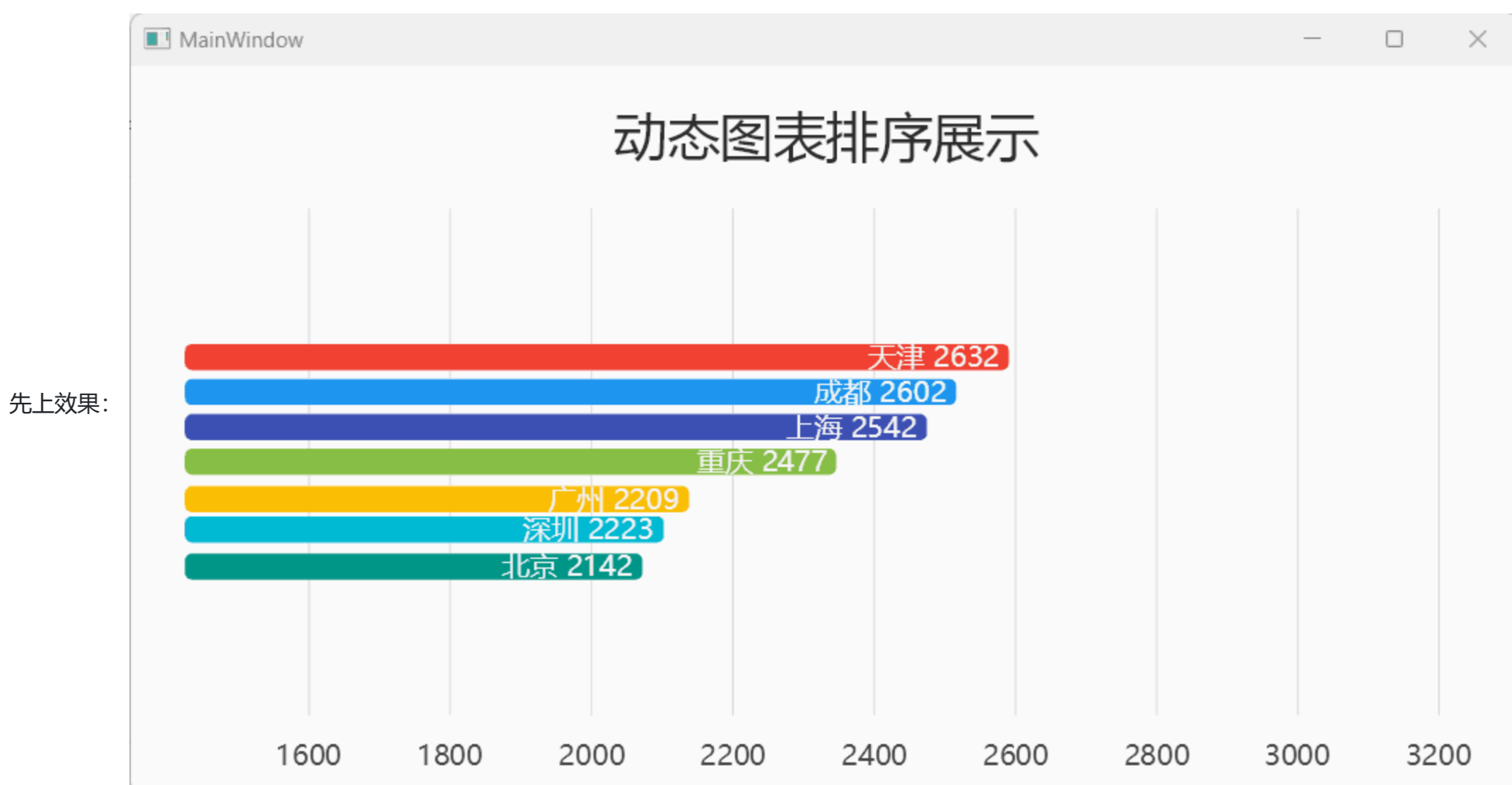
作者：微信公众号：【架构师老卢】

12-15 8:17

~ 142



概述：本文代码示例演示了如何在WPF中使用LiveCharts库创建动态条形图。通过创建数据模型、ViewModel和在XAML中使用'CartesianChart'控件，你可以轻松实现图表的数据绑定和动态更新。我将通过清晰的步骤指南包括详细的中文注释，帮助你快速理解并应用这一功能。



在WPF中使用LiveCharts生成动态的条形图需要以下步骤。以下是详细的实例源代码：

步骤 1: 引用LiveCharts库

首先，在项目中引用LiveCharts库。你可以通过NuGet包管理器来安装LiveCharts.Wpf：

```
1 | Install-Package LiveCharts.Wpf
```

步骤 2: 创建WPF项目

创建一个新的WPF项目，确保已经在XAML文件中引用了LiveCharts的命名空间：

```
1 | xmlns:lvc="clr-namespace:LiveCharts.Wpf;assembly=LiveCharts.Wpf"
```

步骤 3: 创建数据模型

创建一个数据模型，用于存储条形图的数据：

```
1 | // BarChartData.cs
2 | public class BarChartData
3 | {
4 |     public string Category { get; set; }
5 |     public double Value { get; set; }
6 | }
```

步骤 4: 创建ViewModel

创建一个ViewModel类，用于处理图表的数据和逻辑。在这个类中，你将生成数据并将其绑定到图表控件：

```
1 | public partial class ViewModel : ViewModelBase
2 | {
3 |     private readonly Random _r = new();
4 |     private static readonly (string, double)[] s_initialData =
5 |     {
6 |         ("上海", 500),
7 |         ("北京", 450),
8 |         ("深圳", 520),
9 |         ("广州", 550),
10 |        ("重庆", 660),
11 |        ("天津", 920),
12 |        ("成都", 1000)
13 |    };
14 |
15 |     [ObservableProperty]
16 |     private ISeries[] _series =
17 |         s_initialData
18 |         .Select(x => new RowSeries<ObservableValue>
19 |         {
20 |             Values = new[] { new ObservableValue(x.Item2) },
21 |             Name = x.Item1,
22 |             Stroke = null,
23 |             MaxBarWidth = 25,
24 |             DataLabelsPaint = new SolidColorBrush(new SKColor(245, 245, 245)),
25 |             DataLabelsPosition = DataLabelsPosition.End,
26 |             DataLabelsTranslate = new LvcPoint(-1, 0),
27 |             DataLabelsFormatter = point => $"{point.Context.Series.Name} {point.PrimaryValue}"
28 |         })
29 |         .OrderByDescending(x => ((ObservableValue[])x.Values!)[0].Value)
30 |         .ToArray();
31 |
32 |     [ObservableProperty]
33 |     private Axis[] _xAxes = { new Axis { SeparatorsPaint = new SolidColorBrush(new SKColor(220, 220, 220)) } };
34 |
35 |     [ObservableProperty]
36 |     private Axis[] _yAxes = { new Axis { IsVisible = false } };
37 |
38 |     /// <summary>
39 |     /// 动态修改数据,实际项目中读取真实数据
40 |     /// </summary>
41 |     public void RandomIncrement()
42 |     {
43 |         foreach (var item in Series)
44 |         {
45 |             if (item.Values is null) continue;
46 |
47 |             var i = ((ObservableValue[])item.Values)[0];
48 |             i.Value += _r.Next(0, 100);
49 |         }
50 |         //对新数据重新排序
51 |         Series = Series.OrderByDescending(x => ((ObservableValue[])x.Values!)[0].Value).ToArray();
52 |     }
53 | }
54 |
```

步骤 5: 在XAML中使用图表控件

在XAML中使用LiveCharts的'CartesianChart'控件来显示条形图：

```
1 | <!-- MainWindow.xaml -->
2 | <Window x:Class="Sample_Charts_Bars.MainWindow"
3 |     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
4 |     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5 |     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6 |     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
7 |     xmlns:lvc="clr-namespace:LiveChartsCore.SkiaSharpView.WPF;assembly=LiveChartsCore.SkiaSharpView.WPF"
8 |     xmlns:local="clr-namespace:Sample_Charts_Bars"
9 |     mc:Ignorable="d"
10 |     Title="MainWindow" Height="450" Width="800">
11 |     <Window.DataContext>
12 |         <local:ViewModel/>
13 |     </Window.DataContext>
14 |     <Grid>
15 |         <Grid.RowDefinitions>
16 |             <RowDefinition Height="*" />
17 |         </Grid.RowDefinitions>
18 |         <lvc:CartesianChart Title="{Binding Title}"
19 |             Series="{Binding Series}"
20 |             XAxes="{Binding XAxes}"
21 |             YAxes="{Binding YAxes}"
22 |             TooltipPosition="Hidden">
23 |         </lvc:CartesianChart>
24 |     </Grid>
25 | </Window>
26 |
```

步骤 6: 在MainWindow中设置DataContext

在MainWindow.xaml.cs中设置ViewModel的DataContext，以便数据绑定：

```
1 | // MainWindow.xaml.cs
2 | using System.Windows;
3 |
4 | namespace YourNamespace
5 | {
6 |     public partial class MainWindow : Window
7 |     {
8 |         public MainWindow()
9 |         {
10 |             InitializeComponent();
11 |             Update();
12 |         }
13 |
14 |         public async void Update()
15 |         {
16 |             var vm = (ViewModel)DataContext;
17 |             while (true)
18 |             {
19 |                 //更新数据
20 |                 Application.Current.Dispatcher.Invoke(vm.RandomIncrement());
21 |                 await Task.Delay(100);
22 |             }
23 |         }
24 |     }
25 | }
```




需要查看更多实例请关注我，后面将逐个发布。

相关代码下载地址



重要提示!：取消关注公众号后将无法再启用回复功能，不支持解封!

第一步: 微信扫码关注公众号“架构师老卢”

第二步: 在公众号聊天框发送 **code: 43861**，如：  **code: 43861**   获取下载地址

第三步: 恭喜你，快去下载你想要的资源吧