

# Bài: ListView Filtering trong Lập trình WPF

Xem bài học trên website để ủng hộ Kteam: [ListView Filtering trong Lập trình WPF](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

## Dẫn nhập

Sức mạnh của hệ điều hành Window là không thể chối cãi. Và để tạo nên sức mạnh đó không thể thiếu những ứng dụng mạnh mẽ. Vậy để tạo ra những ứng dụng đó, người lập trình viên cần học cái gì? [Lập trình Winform](#) thật sự rất mạnh. Nhưng vẫn còn những giới hạn. Vậy cùng nhau vượt qua các giới hạn đó với [Lập trình WPF](#).

## Nội dung

[Tổng quan lập trình WPF](#)

[Download project](#)

**MainWindow.xaml.cs**

**C#:**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace ListView_Filtering
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            List<User> items = new List<User>();
            items.Add(new User() { Name = "HowKteam", Age = 42 });
            items.Add(new User() { Name = "Kteam", Age = 39 });
            items.Add(new User() { Name = "Free Education", Age = 13 });
            items.Add(new User() { Name = "Share to be better", Age = 13 });
            lvUsers.ItemsSource = items;

            CollectionView view = (CollectionView)CollectionViewSource.GetDefaultView(lvUsers.ItemsSource);
            view.Filter = UserFilter;
        }

        private bool UserFilter(object item)
        {
            if (String.IsNullOrEmpty(txtFilter.Text))
                return true;
            else
                return ((item as User).Name.IndexOf(txtFilter.Text, StringComparison.OrdinalIgnoreCase) >= 0);
        }

        private void txtFilter_TextChanged(object sender, System.Windows.Controls.TextChangedEventArgs e)
        {
            CollectionViewSource.GetDefaultView(lvUsers.ItemsSource).Refresh();
        }
    }

    public enum SexType { Male, Female };

    public class User
    {
        public string Name { get; set; }

        public int Age { get; set; }

        public string Mail { get; set; }

        public SexType Sex { get; set; }
    }
}

```

MainWindow.xaml

xml:

```
<Window x:Class="ListView_Filtering.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525">
    <DockPanel Margin="10">
        <TextBox DockPanel.Dock="Top" Margin="0,0,0,10" Name="txtFilter" TextChanged="txtFilter_TextChanged" />
        <ListView Name="lvUsers">
            <ListView.Resources>
                <Style TargetType="{x:Type GridViewColumnHeader}">
                    <Setter Property="HorizontalContentAlignment" Value="Center"></Setter>
                </Style>
            </ListView.Resources>
            <ListView.View>
                <GridView>
                    <GridViewColumn Header="Name" Width="120" >
                        <GridViewColumn.CellTemplate>
                            <DataTemplate>
                                <Grid Width="110">
                                    <TextBlock Text="{Binding Name}" HorizontalAlignment="Center"></TextBlock>
                                </Grid>
                            </DataTemplate>
                        </GridViewColumn.CellTemplate>
                    </GridViewColumn>
                    <GridViewColumn Header="Age" Width="50" DisplayMemberBinding="{Binding Age}" />
                </GridView>
            </ListView.View>
        </ListView>
    </DockPanel>
</Window>
```

### Kết luận

Bài sau chúng ta sẽ cùng nhau tìm hiểu về [TreeView trong lập trình WPF](#) nhé.

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.