

Bài: TreeView Binding trong Lập trình WPF

Xem bài học trên website để ủng hộ Kteam: [TreeView Binding 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.Collections.ObjectModel;
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 TreeView_Binding
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            //MenuItem root = new MenuItem() { Title = "Menu" };
            //MenuItem childItem1 = new MenuItem() { Title = "Child item #1" };
            //childItem1.Items.Add(new MenuItem() { Title = "Child item #1.1" });
            //childItem1.Items.Add(new MenuItem() { Title = "Child item #1.2" });
            //root.Items.Add(childItem1);
            //root.Items.Add(new MenuItem() { Title = "Child item #2" });
            //trvMenu.Items.Add(root);

            List<Family> families = new List<Family>();

            Family family1 = new Family() { Name = "HowKteam.com" };
            family1.Members.Add(new FamilyMember() { Name = "HowKteam", Age = 42 });
            family1.Members.Add(new FamilyMember() { Name = "Kteam", Age = 39 });
            family1.Members.Add(new FamilyMember() { Name = "Free Education", Age = 13 });
            families.Add(family1);

            Family family2 = new Family() { Name = "Kteam" };
            family2.Members.Add(new FamilyMember() { Name = "K9", Age = 31 });
            family2.Members.Add(new FamilyMember() { Name = "Mushi", Age = 28 });
            families.Add(family2);

            trvFamilies.ItemsSource = families;
        }
    }

    public class MenuItem
    {
        public MenuItem()
        {
            this.Items = new ObservableCollection<MenuItem>();
        }

        public string Title { get; set; }

        public ObservableCollection<MenuItem> Items { get; set; }
    }

    public class Family
    {

```

```

    public Family()
    {
        this.Members = new ObservableCollection<FamilyMember>();
    }

    public string Name { get; set; }

    public ObservableCollection<FamilyMember> Members { get; set; }
}

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

    public int Age { get; set; }
}
}

```

MainWindow.xaml

xml:

```

<Window x:Class="TreeView_Binding.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:self="clr-namespace:TreeView_Binding"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
        <!--<TreeView Name="trvMenu">
            <TreeView.ItemTemplate>
                <HierarchicalDataTemplate DataType="{x:Type self:MenuItem}"
                    ItemsSource="{Binding Items}">
                    <TextBlock Text="{Binding Title}" />
                </HierarchicalDataTemplate>
            </TreeView.ItemTemplate>
        </TreeView>-->

        <TreeView Name="trvFamilies">
            <TreeView.Resources>
                <HierarchicalDataTemplate DataType="{x:Type self:Family}" ItemsSource="{Binding Members}">
                    <StackPanel Orientation="Horizontal">
                        <Image Width="20" Stretch="UniformToFill" Source="/Images/logo1.png" Margin="0,0,5,0" />
                        <TextBlock Text="{Binding Name}" />
                        <TextBlock Text=" [" Foreground="Blue" />
                        <TextBlock Text="{Binding Members.Count}" Foreground="Blue" />
                        <TextBlock Text="]" Foreground="Blue" />
                    </StackPanel>
                </HierarchicalDataTemplate>
                <DataTemplate DataType="{x:Type self:FamilyMember}">
                    <StackPanel Orientation="Horizontal">
                        <Image Width="20" Stretch="UniformToFill" Source="/Images/logo2.png" Margin="0,0,5,0" />
                        <TextBlock Text="{Binding Name}" />
                        <TextBlock Text=" (" Foreground="Green" />
                        <TextBlock Text="{Binding Age}" Foreground="Green" />
                        <TextBlock Text=" years)" Foreground="Green" />
                    </StackPanel>
                </DataTemplate>
            </TreeView.Resources>
        </TreeView>
    </Grid>
</Window>

```

Kết luận

Bài sau chúng ta sẽ cùng nhau tìm hiểu về [TreeView Lazyloading 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"**.

