

از `ItemsControl` برای ارائه مجموعه ای از کنترل ها استفاده می شود، در اینجا قرار است از آن استفاده کنیم و یک کنترل پویا ایجاد کنیم. برای مثال در نظر بگیرید، قرار است یک `DropDown Panel` ایجاد کنیم و در جاهای مختلف برنامه کنترل های مختلفی را درون آن قرار بدهیم. برای ایجاد آن به صورت زیر عمل میکنیم:

```
<UserControl x:Class="MySystem.Common.Controls.DropDownPanel"
    x:Name="This">
    <Grid>
        <ToggleButton x:Name="ShowPopupButton"/>
        <Popup
            PlacementTarget="{Binding ElementName=ShowPopupButton}"
            Placement="{Binding PopupPlacement, ElementName=this}"
            PopupAnimation="Slide"
            AllowsTransparency="True"
            Focusable="True"
            StaysOpen="False"
            IsOpen="{Binding IsChecked, ElementName=ShowPopupButton }">
            <Border Background="#FFE3EAF3" BorderThickness="1" Padding="2">
                <Grid>
                    <ItemsControl ItemsControl.ItemsSource="{Binding Path=PanelItems,ElementName=This"
                >
                    <ItemsControl.ItemsPanel>
                        <ItemsPanelTemplate>
                            <Grid>
                                </Grid>
                            </ItemsPanelTemplate>
                        </ItemsControl.ItemsPanel>
                    </ItemsControl>
                </Grid>
            </Border>
        </Popup>
    </Grid>
</UserControl>
```

همانگونه که در کد بالا می بینید، برای ایجاد `DropDown Panel` از یک `ToggleButton` و یک `Popup` که خصوصیت `IsOpen` آن به `IsChecked` مربوط به `ToggleButton` وصل شده است، استفاده کردیم و در قسمت بدنه کنترل به جای قراردادن کنترل هایی که قرار است در آن نمایش داده شوند، از یک `ItemsControl` استفاده کردیم که خصوصیت `ItemsSource` آن به یک خصوصیت پیوست شده از نوع `ObservableCollection<UIElement>` در `Code Behind`، مقید شده است. تعریف این خصوصیت پیوست شده به صورت زیر است:

```
public static readonly DependencyProperty PanelItemsProperty = DependencyProperty.Register("PanelItems"
    , typeof(ObservableCollection<UIElement>)
    , typeof(DropDownPanel)
    , new PropertyMetadata(new ObservableCollection<UIElement>()));
public ObservableCollection<UIElement> PanelItems
{
    get
    {
        return (ObservableCollection<UIElement>)GetValue(PanelItemsProperty);
    }
    set
    {
        SetValue(PanelItemsProperty, value);
    }
}
```

برای استفاده از کنترل یک وهله از این کنترل را ایجاد می کنیم و کنترل هایی که قرار است در DropDown Panel نمایش داده شوند را به PanelItems اضافه می کنیم:

```
<Window x:Class="MySystem.UI.View.Window1"
    xmlns:controls="clr-
namespace:MySystem.Common.Controls;assembly=GoldAccountingSystem.Common.Controls">
    <Grid>
        <controls:DropDownPanel>
            <controls:DropDownPanel.PanelItems>
                !--Put Controls Here--!
            </controls:DropDownPanel.PanelItems>
        </controls:DropDownPanel>
    </Grid>
</Window>
```

تا این مرحله کنترل مورد نظر را ایجاد و استفاده کردیم. اما یک مشکل وجود دارد، چنانچه از این کنترل چند بار در یک فرم استفاده شود، به درستی عمل نمی کند به اینصورت که فرزندان PanelItems تمام شی های ساخته شده از کنترل در یک فرم برابر هم و برابر مقداری می شود که برای آخرین کنترل قراردادده ایم. دلیل این امر این است که ما یکبار در هنگام تعریف خصوصیت PanelItems یک وهله از آن را به عنوان مقدار پیش فرض ایجاد کردیم و برای همه ی نمونه هایی از کنترل که در فرم قرار می گیرند از همان وهله استفاده می شود.

برای حل مشکل فوق یک کلاس از نوع ObservableCollection<UIElement> ایجاد کرده و هنگام ساختن کنترل در فرم از این کلاس برای وهله سازی مجدد از PanelItems استفاده می کنیم:

```
namespace MySystem.Common.Controls
{
    public class UIElementCollection : ObservableCollection<UIElement> { }
}
```

همانطور که گفته شد از کلاس ایجاد شده برای وهله سازی به صورت زیر استفاده می شود:

```
<Window x:Class="MySystem.UI.View.Window1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:controls="clr-
namespace:MySystem.Common.Controls;assembly=GoldAccountingSystem.Common.Controls">
    <Grid>
        <commonControls:DropDownPanel>
            <commonControls:DropDownPanel.PanelItems>
                <commonControls:UIElementCollection>
                    !--Put Controls Here--!
                </commonControls:UIElementCollection>
            </commonControls:DropDownPanel.PanelItems>
        </commonControls:DropDownPanel>
    </Grid>
</Window>
```

یک مثال ساده از توضیحات بالا را از آدرس روبرو می توانید دریافت نمایید: <a349625a156d4aeaaca288f000ae6d7a.rar>

### نظرات خوانندگان

نویسنده:

محسن خان

تاریخ:

۱۸:۳۲ ۱۳۹۲/۰۴/۰۶

اگر امکانش باشه این مطلب رو به صورت یک مثال یا پروژه ساده قابل دریافت قرار بدید، امکان پیگیری اون بیشتر خواهد شد. ممنون.

نویسنده:

محبوبه محمدی

تاریخ:

۹:۷ ۱۳۹۲/۰۴/۰۹

سلام، ممنون از پیشنهادتون، یک مثال خیلی ساده به توضیحات بالا اضافه کردم.