ItemTemplateSelector

Kdyz potrebujeme menit dataTemplaty v zavislosti na nejake vlastnosti datacontextu, pouzijeme ItemTemplateSelector (u elementu ktere maji itemy). V zmenovadle jsem potreboval zmenit template pro ComboBox . Jedna templata byla pouzita kdyz nebyl otevreny popup a druha na itemy v otevrenem popup okne.

* V resourcech okna si definujeme templaty:

<DataTemplate x:Key="MySelectedItemTemplate">

<TextBlock FontWeight="DemiBold" Text="{Binding Name}" FontSize="13"/>

</DataTemplate>

<DataTemplate x:Key="MyDropDownTemplate">

<StackPanel Orientation="Horizontal">

<TextBlock Text="{Binding Name}" >

<TextBlock.FontWeight>

<MultiBinding Converter="{StaticResource NameToFontWeightConverter}">

<Binding Path="Name" />

<Binding RelativeSource="{RelativeSource AncestorType=UserControl}" />

</MultiBinding>

</TextBlock.FontWeight>

<TextBlock.Foreground>

<MultiBinding Converter="{StaticResource NameToForegroundConverter}">

<Binding Path="Name" />

<Binding RelativeSource="{RelativeSource AncestorType=UserControl}" />

</MultiBinding>

</TextBlock.Foreground>

</TextBlock>

</StackPanel>

</DataTemplate>

* Vytvorime tridu ktera bude rozhodovat ktera templata bude pouzita. Musi dedit od DataTemplateSelector:

public class ComboBoxTemplateSelector : DataTemplateSelector

{

public override DataTemplate SelectTemplate(object item, DependencyObject container)

{

FrameworkElement elemnt = container as FrameworkElement;

// var type = item as ObjectiveType;

ComboBoxItem comboBoxItem = container.GetVisualParent<ComboBoxItem>();

if (comboBoxItem == null)

{

return elemnt.FindResource("MySelectedItemTemplate") as DataTemplate;

}

else

{

return elemnt.FindResource("MyDropDownTemplate") as DataTemplate;

}

}

* Pridame jmenny prostor podle umisteni tridy:

xmlns:controls="clr-namespace:MIR.Media.Changing2.Gui.Controls"

a do resources okna:

<controls:ComboBoxTemplateSelector x:Key="ComboBoxTemplateSelector" />

* Pouziti : V nasem combu rekneme ze:

<ComboBox

Name="VariableComboBox"

Grid.Row="1"

Grid.Column="1"

MinWidth="300"

Margin="7"

ItemTemplateSelector="{StaticResource ComboBoxTemplateSelector}"

ItemsSource="{Binding Path=DataContext.VariableComboItems, RelativeSource={RelativeSource FindAncestor, AncestorType=UserControl}}"

SelectedItem="{Binding Path=SelectedVariable, Mode=OneWayToSource, UpdateSourceTrigger=PropertyChanged}"

Style="{StaticResource ComboStyle}"

Visibility="Visible" />

* Ve tride comboBoxTemplateSelector se v tomto pripade rozlisuje jestli jestli je otevrene popup okno na zaklade comboBoxItem == null. Stejne tak se ale da rozhodovat napr na zaklade nejake vlastnosti . Jde o to ze tady napr. Bindujeme selectedItem na SelectedVariable ktera je ale na objektu FilterElement. Takze kdyby jsme rekli ze

Var type = item as FilterPartView;

Var context = type.DataContext;

Var collection = context.Filters;

If(collection.neco > 3) {return neco}

else {return necoJineho}

 public override DataTemplate SelectTemplate(object item, DependencyObject container)

    {

        FrameworkElement elemnt = container as FrameworkElement;

        User user = item as User;

        if(user.IsPremiumUser)

        {

            return elemnt.FindResource("PremiumUserDataTemplate") as DataTemplate;

        }

        else

        {

            return elemnt.FindResource("NormalUserDataTemplate") as DataTemplate;

        }

    }