SelectedItems / Multiselect RadGridView Telerik

Funkcni project je v [C:\Users\phlavenka\OneDrive\Projects\RadGridViewBindSelectedItems](file:///C:\Users\phlavenka\OneDrive\Projects\RadGridViewBindSelectedItems)

Je to projekt od admina z Teleriku. Jejich radGrid jako obvykle nema SelectedItems (nebo ma ale je readOnly takze nejde jednoduse setnout z xaml)

Je potreba vytvorit tridu MultiSelectBehavior ktera pouziva dependency property jako uloziste pro SelectedItemsProperty

# Konecne (Spravne reseni)

Nakonec to musel porazit Filip:

[‎4/‎13/‎2018 2:47 PM] Filip Čálek:

multiselect pushnut

upravuj dle libosti;)

[‎4/‎13/‎2018 2:47 PM] Peter Hlavenka:

to si delas srandu

[‎4/‎13/‎2018 2:47 PM] Filip Čálek:

ee. mrkni na to

radgridview tam ma totiz "featuru", ze setovani property nic nedela

ale maj metody Select a SelectAll

listBox.Select(selectedItems);

## Vytvorime CustomControl podedeny od RadGridView

using System.Collections;

using System.Windows;

using Telerik.Windows.Controls;

namespace MIR.Media.Changing2.Gui.Behaviors

{

public class ExtendedRadGridView : RadGridView

{

public static readonly DependencyProperty BindableSelectedItemsProperty = DependencyProperty.Register(

"BindableSelectedItems", typeof(IList), typeof(ExtendedRadGridView), new PropertyMetadata(default(IList), SelectedItemsChanged));

private static bool m\_selectionChanging;

public IList BindableSelectedItems

{

get => (IList) GetValue(BindableSelectedItemsProperty);

set => SetValue(BindableSelectedItemsProperty, value);

}

// Touto metodou muzeme urcit SelectedItems z ViewModelu

private static void SelectedItemsChanged(DependencyObject dependencyObject, DependencyPropertyChangedEventArgs eventArgs)

{

if (m\_selectionChanging)

{

return;

}

if (dependencyObject is ExtendedRadGridView radGridView)

{

try

{

m\_selectionChanging = true;

var selectedItems = (IList) eventArgs.NewValue;

// Tim ze nas dependency objekt ma predka , ma i kolekci SelectedItems . Vyprazdnime ji.

radGridView.SelectedItems.Clear();

// radgridview tam ma totiz "featuru", ze setovani property nic nedela ale maj metody Select a SelectAll

radGridView.Select(selectedItems);

}

finally

{

m\_selectionChanging = false;

}

}

}

}

}

## V Xamlu pouzijeme misto klasicke telerikovske komponenty rozsirenou komponentu:

xmlns:behaviors="clr-namespace:MIR.Media.Changing2.Gui.Behaviors"

<behaviors:ExtendedRadGridView // Z jmenneho prostoru behaviors

x:Name="RadGridView"

Grid.Row="1"

Grid.Column="0"

Margin="0"

AutoExpandGroups="True"

AutoGenerateColumns="False"

BindableSelectedItems="{Binding SelectedItems, Mode=TwoWay}" // Nova dependencyProperty

CanUserFreezeColumns="True"

CanUserResizeColumns="True"

Focusable="False"

GroupPanelBackground="LightBlue"

GroupRenderMode="Flat"

IsReadOnly="True"

IsSynchronizedWithCurrentItem="True"

ItemsSource="{Binding Motivlets}"

RowIndicatorVisibility="Collapsed"

SelectedItem="{Binding SelectedItem}" // SelectedItem potrebujeme tady v setteru budeme nastavovat to, co se ma dostat do kolekce SelectedItems

SelectionMode="Extended" // Nezapomenout Extended Mode

ShowColumnSortIndexes="True">

## ViewModel

* Mame kolekci ktera je pro RadGridView nastavena jako ItemsSource:

public BindableCollection<ComposedWrapper> Motivlets

{

get => m\_motivlets;

set

{

m\_motivlets = value;

NotifyOfPropertyChange();

}

}

* Ve ViewModelu mame kolekci SelectedItems , coz musi byt IList():

private IList m\_selectedItems;

public IList SelectedItems

{

get => m\_selectedItems;

set

{

m\_selectedItems = value;

NotifyOfPropertyChange();

}

}

* A mame SelectedItem ktery muze byt jakehokoli typu:

Tady se nastavuje, co se dostane do kolekce SelectedItems

public ComposedWrapper SelectedItem

{

get => m\_selectedItem;

set

{

var test = Motivlets.Skip(15).Take(3); // Musi to byt objekty, ktere se nachazi na kolekci ktera je nastavena jako ItemsSource

SelectedItems = new List<ComposedWrapper>(test); // Dulezite je nepridavat do kolekce itemy, ale vytvorit novy objekt ( jinak se do setteru SelectedItems nedostaneme)

}

}

# Predchozi ( Nespravna reseni )

## MultiSelectBehavior

using System.Collections;

using System.Collections.Specialized;

using System.Windows;

using System.Windows.Interactivity;

using Telerik.Windows.Controls;

namespace WpfApplication1

{

public class MultiSelectBehavior : Behavior<RadGridView>

{

// Using a DependencyProperty as the backing store for SelectedItemsProperty. This enables animation, styling, binding, etc...

public static readonly DependencyProperty SelectedItemsProperty =

DependencyProperty.Register("SelectedItems", typeof(INotifyCollectionChanged), typeof(MultiSelectBehavior), new PropertyMetadata(OnSelectedItemsPropertyChanged));

private RadGridView Grid => AssociatedObject;

public INotifyCollectionChanged SelectedItems

{

get => (INotifyCollectionChanged) GetValue(SelectedItemsProperty);

set => SetValue(SelectedItemsProperty, value);

}

private static void OnSelectedItemsPropertyChanged(DependencyObject target, DependencyPropertyChangedEventArgs args)

{

var collection = args.NewValue as INotifyCollectionChanged;

if (collection != null)

{

collection.CollectionChanged += ((MultiSelectBehavior) target).ContextSelectedItems\_CollectionChanged;

}

}

protected override void OnAttached()

{

base.OnAttached();

Grid.SelectedItems.CollectionChanged += GridSelectedItems\_CollectionChanged;

}

private void ContextSelectedItems\_CollectionChanged(object sender, NotifyCollectionChangedEventArgs e)

{

UnsubscribeFromEvents();

Transfer(SelectedItems as IList, Grid.SelectedItems);

SubscribeToEvents();

}

private void GridSelectedItems\_CollectionChanged(object sender, NotifyCollectionChangedEventArgs e)

{

UnsubscribeFromEvents();

Transfer(Grid.SelectedItems, SelectedItems as IList);

SubscribeToEvents();

}

private void SubscribeToEvents()

{

Grid.SelectedItems.CollectionChanged += GridSelectedItems\_CollectionChanged;

if (SelectedItems != null)

{

SelectedItems.CollectionChanged += ContextSelectedItems\_CollectionChanged;

}

}

private void UnsubscribeFromEvents()

{

Grid.SelectedItems.CollectionChanged -= GridSelectedItems\_CollectionChanged;

if (SelectedItems != null)

{

SelectedItems.CollectionChanged -= ContextSelectedItems\_CollectionChanged;

}

}

public static void Transfer(IList source, IList target)

{

if (source == null || target == null)

{

return;

}

target.Clear();

foreach (var o in source)

{

target.Add(o);

}

}

}

}

## MainWindow.xaml

Zadefinujeme si jmenny prostor

xmlns:i="clr-namespace:System.Windows.Interactivity;assembly=System.Windows.Interactivity"

Pomoci ktereho v gridu ovladame kolekci SelectedItems

<i:Interaction.Behaviors>

<local:MultiSelectBehavior SelectedItems="{Binding SelectedItems, Source={StaticResource context}}" />

</i:Interaction.Behaviors>

Zadefinujeme datacontext

<Window.Resources>

<local:MyDataContext x:Key="context" />

</Window.Resources>

<Window

x:Class="WpfApplication1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:i="clr-namespace:System.Windows.Interactivity;assembly=System.Windows.Interactivity" // Jmenny prostor

xmlns:local="clr-namespace:WpfApplication1"

xmlns:telerik="http://schemas.telerik.com/2008/xaml/presentation"

Title="MainWindow">

<Window.Resources>

<local:MyDataContext x:Key="context" /> // Datacontext

</Window.Resources>

<Grid

x:Name="LayoutRoot"

Background="White"

DataContext="{StaticResource context}">

<Grid.ColumnDefinitions>

<ColumnDefinition />

<ColumnDefinition Width="200" />

</Grid.ColumnDefinitions>

<telerik:RadGridView

ItemsSource="{Binding Data}"

SelectedItem="{Binding SelectedItem, Mode=TwoWay}"

SelectionMode="Extended"> // Mode

<i:Interaction.Behaviors>

<local:MultiSelectBehavior SelectedItems="{Binding SelectedItems, Source={StaticResource context}}" /> // Ukazeme na kolekci SelectedItems na datacontextu

</i:Interaction.Behaviors>

</telerik:RadGridView>

<StackPanel Grid.Column="1" Orientation="Vertical">

<telerik:RadButton Click="Button\_Click" Content="Clear selected items" />

<TextBlock Text="Selected items:" />

<ListBox DisplayMemberPath="ID" ItemsSource="{Binding SelectedItems}" />

</StackPanel>

</Grid>

</Window>

## Na co dat pozor

1. Original projekt ma starsi .net Framework . Nainstaloval jsem proto nuget Telerik.Windows.Diagrams.3.0.0
2. Je treba zadefinovat datacontext na kterem je kolekce SelectedItems ( nema setter) Kolekce se normalne plni, kdekoli v kodu se k ni dostaneme.

public ObservableCollection<ComposedWrapper> SelectedItems => m\_selectedItems ?? (m\_selectedItems = new ObservableCollection<ComposedWrapper>());

1. Pokud datacontext nema defaultni konstruktor je potreba ho najit ancestorem

<i:Interaction.Behaviors>

<behaviors:MultiSelectBehavior SelectedItems="{Binding DataContext.SelectedItems, RelativeSource={RelativeSource FindAncestor, AncestorType=UserControl}}" />

</i:Interaction.Behaviors>

1. Aby bylo mozne notifikovat UI o tom, ze se zmenila kolekce selectedItems, musim mit I propertu Selected Item

SelectedItem="{Binding SelectedItem, Mode=TwoWay}"

public MyObject SelectedItem

{

get

{

return \_SelectedItem;

}

set

{

if (\_SelectedItem != value)

{

\_SelectedItem = value;

OnPropertyChanged("SelectedItem");

}

}

}

Nebo ve zmenovadle to chtelo PropertyChangedEventArgs:

public ComposedWrapper SelectedItem

{

get { return m\_selectedItem; }

set

{

if (m\_selectedItem != value)

{

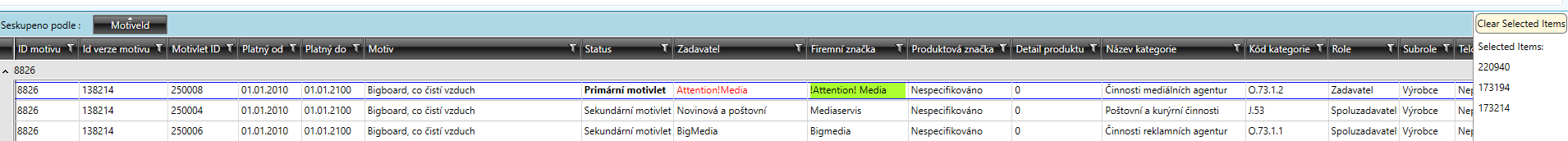
m\_selectedItem = value;

OnPropertyChanged(new PropertyChangedEventArgs("SelectedItem"));

}

}

}



Oprava od Filipa

V zadani bylo, aby se po kliknuti na motivlet, pridaly o ostatni motivlety ktere jsou na stejnem motivu, na jinych verzich. Ale jen pokud se jedna o ty ktere uzivatel vybral. (Stejne motivletid) Neslo zvyraznit dalsi radky. Filip nechal vyhodit vsechno co jsem pridal navic do tridy MultiSelectBehavior (Dependency property jako selectedDate a Motivlets). Z kolekce SelectedItems na viewModelu udelal BindableCollection, protoze po pridani Add nebo AddRange se ve tride BindableCollection vola Notify a rovnou na UIThreadu.

public BindableCollection<ComposedWrapper> SelectedItems { get; } = new BindableCollection<ComposedWrapper>();

V setru SelectedItem zavolame metodu ktera poresi to, jestli se ma do kolekce SelectedItems pridat jeste jina polozka. (To co jsem predtim resil v metode MultiselectBehavior.Transfer)

Nakonec pouzijeme tridu z Frameworku MultiSelectBehavior

Ta se ale zapisuje jako AttachedProperty a navic je staticka takze je ulozena nekde ve Frameworku a muzeme se k ni dostat pomoci namespacu.

**Puvodni zapis:**

<i:Interaction.Behaviors>

<behaviors:MultiSelectBehavior SelectedItems="{Binding DataContext.SelectedItems, RelativeSource={RelativeSource FindAncestor, AncestorType=UserControl}}" />

</i:Interaction.Behaviors>

**Novy zapis:**

<telerik:RadGridView behaviors:MultiSelectBehavior.SynchronizedSelectedItems="{Binding Path=SelectedItems}"

SelectedItems musi ted byt **IList**

Kdyz kliknu na jednu polozku, dostanu se do setteru SelectedItem.

Tady muzu vytvorit novy objekty typu

public ComposedWrapper SelectedItem

{

get => m\_selectedItem;

set

{

m\_selectedItem = value;

var collection = new BindableCollection<ComposedWrapper>();

collection.Add(m\_selectedItem);

SelectedItems = new BindableCollection<ComposedWrapper>(collection);

Dostanu se tim do setteru SelectedItems

A tim do tridy MultiSelectBehavior. GridSelectedItems\_CollectionChanged(object sender, NotifyCollectionChangedEventArgs e) tady se setne v kolekci SelectedItems zase jen to co je nakliknute.

Kdyz mam Mode MultiSelect , nebo Extended a drzim control, zacinam taky v metode GridSelectedItems\_CollectionChanged(object sender, NotifyCollectionChangedEventArgs e)