

WPF : Tri simple d'une LISTVIEW

Pour trier une ListView il faut intercepter le message de click sur une entête de la liste (Header.Click) et déclencher le tri de la liste via la binding collection.

Il suffit pour cela d'ajouter dans le XAML l'orientation du click

```
<ListView x:Name="LvData" ItemsSource="{Binding PartData.PartSet}"  
GridViewColumnHeader.Click="HeaderClick">
```

Le HeaderClick dans le code behind devrait tout simplement ajouter un appel au SortDescription de la CollectionView qui fait référence aux ItemsSource courant, Vu qu'en WPF le binding est nécessaire dans une ListView, le ItemsSource n'est jamais null (théoriquement). Mais rien n'empêche d'ajouter un try catch pour traiter les exceptions.

La variable _lastDirection permet d'inverser l'ordre de tri à chaque click.

Code behind

```
ListSortDirection _lastDirection = ListSortDirection.Ascending;  
private void HeaderClick(object sender, RoutedEventArgs e)  
{  
    GridViewColumnHeader headerClicked = e.OriginalSource as  
GridViewColumnHeader;  
    if (headerClicked == null) return;  
  
    CollectionView currentView =  
(CollectionView)CollectionViewSource.GetDefaultView(LvData.ItemsSource);  
    currentView.SortDescriptions.Clear();  
    currentView.SortDescriptions.Add(new  
SortDescription(headerClicked.Content.ToString(), _lastDirection));  
  
    if (_lastDirection == ListSortDirection.Ascending) _lastDirection =  
ListSortDirection.Descending;  
    else _lastDirection = ListSortDirection.Ascending;  
}  
  
//Si le nom de la colonne et le nom du binding sont différents il faudrait utiliser  
//le nom du Binding dans le méthode SortDescription de la sorte  
  
Binding bnd = (Binding)headerClicked.Column.DisplayMemberBinding;  
string s = bnd.Path.Path;  
currentview.SortDescriptions.Add(new SortDescription(s, _lastDirection));  
  
//à la place de  
currentView.SortDescriptions.Add(new  
SortDescription(headerClicked.Content.ToString(), _lastDirection));
```

Avec ce petit code dans le XAML et le code behind, on a un tri automatique des colonnes.

On peut ajouter deux flèches dans l'entête, pour cela il faut ajouter deux templates dans le XAML qui dessine les flèches Up et Down :

XAML Code

```
<Window.Resources>  
    <DataTemplate x:Key="HeaderTemplateArrowUp">  
        <DockPanel>
```

```

        <TextBlock Text="{Binding }" />
        <Path x:Name="arrowUp" StrokeThickness="1" Fill="Gray" Data="M 4,8 L 12,8
L 8,4 L 4,8" DockPanel.Dock="Left" Width="12" HorizontalAlignment="Right"
Margin="4,0,4,0" SnapsToDevicePixels="True"/>
    </DockPanel>
</DataTemplate>
<DataTemplate x:Key="HeaderTemplateArrowDown">
    <DockPanel>
        <TextBlock Text="{Binding }" />
        <Path x:Name="arrowDown" StrokeThickness="1" Fill="Gray" Data="M 4,4 L
8,8 L 12,4 L 4,4" DockPanel.Dock="Left" Width="12" HorizontalAlignment="Right"
Margin="4,0,4,0" SnapsToDevicePixels="True"/>
    </DockPanel>
</DataTemplate>
</Window.Resources>

```

L'exploitation de ces flèches sera faite directement dans le code behind et dans la méthode HeaderClick selon la valeur de `_lastDirection`.

```

if (_lastDirection == ListSortDirection.Ascending) headerClicked.Column.HeaderTemplate =
    Resources["HeaderTemplateArrowUp"] as DataTemplate;
else
    headerClicked.Column.HeaderTemplate = Resources["HeaderTemplateArrowDown"] as
DataTemplate;

```

Ainsi le tri par colonne est réalisable à moindre coût.