

C# Mobile

Les 4

- CollectionView



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be



Doelstellingen

- De junior-collega kent de basiswerking van het `CollectionView`
- De junior-collega kan zelf een template definiëren in een `CollectionView`
- De junior-collega kan interacties op een lijst implementeren
- De junior-collega kan een eenvoudige *TableView* instantiëren
- De junior-collega kent de basiswerking van *Shared Resources*



.NET MAUI

CollectionView vs ListView

Views: ListView

- Lijsten tonen in de UI
- Wordt heel vaak gebruikt in applicaties
 - Producten
 - Liedjes
 - Foto's
 - ...



Baboon

Africa & Asia



Capuchin Monkey

Central & South America



Blue Monkey

Central and East Africa



Squirrel Monkey

Central & South America



Golden Lion Tamarin

Brazil

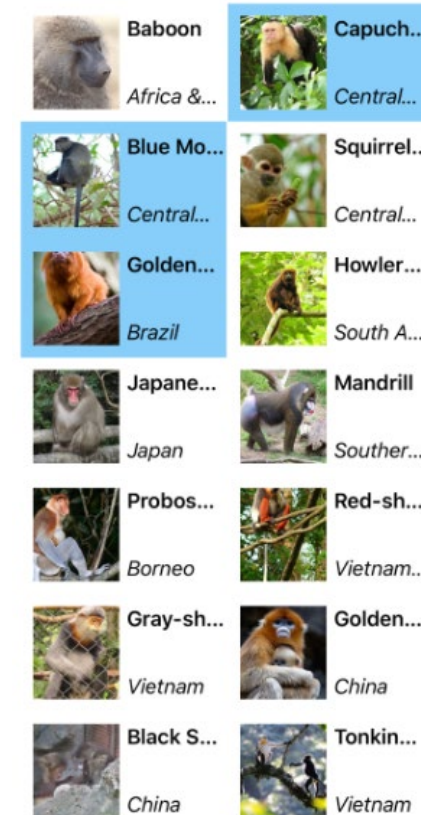
Views: CollectionView

- Update van de oudere ListView (Xamarin)
- Flexibeler en normaal performanter

CollectionView and ListView differences

While the `CollectionView` and `ListView` APIs are similar, there are some notable differences:

- `CollectionView` has a flexible layout model, which allows data to be presented vertically or horizontally, in a list or a grid.
- `CollectionView` supports single and multiple selection.
- `CollectionView` has no concept of cells. Instead, a data template is used to define the appearance of each item of data in the list.
- `CollectionView` automatically utilizes the virtualization provided by the underlying native controls.
- `CollectionView` reduces the API surface of `ListView`. Many properties and events from `ListView` are not present in `CollectionView`.
- `CollectionView` does not include built-in separators.
- `CollectionView` will throw an exception if its `ItemsSource` is updated off the UI thread.



CollectionView: Basis

- XAML element: <CollectionView></CollectionView>
- De items voegen we toe via de ItemsSource property
- Meestal doen we dit in de code-behind file (.xaml.cs)



Placing a [CollectionView](#) inside a [VerticalStackLayout](#) can stop the [CollectionView](#) scrolling, and can limit the number of items that are displayed. In this situation, replace the [VerticalStackLayout](#) with a [Grid](#).

CollectionView: Labo

- Open *Labo – CollectionView* in de Blackboard cursus
- Accepteer de assignment
- Open je repository
- Maak een clone van je repository
- Open het README.md bestand
- Voltooi Deel 1

CollectionView: Basis

- Meestal gaan we onze collectie niet vullen met een lijst van strings maar met een lijst van objecten
- Standaard worden enkel strings weergegeven in de CollectionView

CollectionView: Labo

- Volttooi deel 2 + 3

CollectionView: Template

```
<CollectionView>  
  <CollectionView.ItemTemplate>  
    <DataTemplate>  
      <HorizontalStackLayout>  
      </HorizontalStackLayout>  
    </DataTemplate>  
  </CollectionView.ItemTemplate>  
</CollectionView>
```

- De layout van elk object van een collection kan worden bepaald met *templates*
- We doen dit door de *ItemTemplate* property van de *CollectionView* in te stellen als een *DataTemplate*
- Content van een *DataTemplate* bestaat uit layouts en views!
- Door gebruik te maken van bindings kunnen we de eigenschappen van elk object “linken” aan een view

CollectionView: Labo

- Voltooi deel 4

CollectionView: Binding

- Een CollectionView heeft achterliggend een lijst van objecten
- De opmaak van elk object wordt bepaald door een template
- Een template bestaat uit layouts en views
- Door gebruik te maken van bindings kunnen we de eigenschappen van elk object “linken” aan een view van de template

CollectionView: Binding

```
5 references
public class MiniFig
{
    4 references
    public string Name { get; set; }
    3 references
    public string ImageUrl { get; set; }
    4 references
    public string Race { get; set; }
}
```

```
<CollectionView x:Name="ExampleCollection">
    <CollectionView.ItemTemplate>
        <DataTemplate>
            <Grid Padding="10">
                <Grid.RowDefinitions>
                    <RowDefinition Height="Auto" />
                    <RowDefinition Height="Auto" />
                </Grid.RowDefinitions>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="Auto" />
                    <ColumnDefinition Width="Auto" />
                </Grid.ColumnDefinitions>
                <Image Grid.RowSpan="2"
                    Source="{Binding ImageUrl}"
                    Aspect="AspectFill"
                    HeightRequest="60"
                    WidthRequest="60" />
                <Label Grid.Column="1"
                    Text="{Binding Name}"
                    FontAttributes="Bold" />
                <Label Grid.Row="1"
                    Grid.Column="1"
                    Text="{Binding Race}"
                    FontAttributes="Italic"
                    VerticalOptions="End" />
            </Grid>
        </DataTemplate>
    </CollectionView.ItemTemplate>
</CollectionView>
```

CollectionView: Selectie

- Op een CollectionView zijn verschillende selectiemogelijkheden
 - Single
 - Multiple
 - None
- Indien de user iets selecteert wordt het SelectionChanged event afgevuurd

CollectionView: EmptyView

- EmptyView wordt getoond indien de lijst leeg is
- Kan via Attribute of Property Element syntax

```
<CollectionView x:Name="TodoCollection"  
    EmptyView="No Items found">
```

```
<CollectionView.EmptyView>  
    <ContentView>  
    </ContentView>  
</CollectionView.EmptyView>
```

CollectionView: Labo

- Voltooi deel 5 + 6

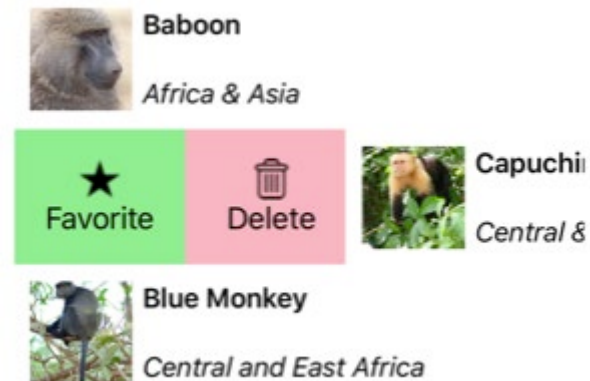


PXL

.NET MAUI CollectionView – Extra's

CollectionView: SwipeView

- Wordt vaak gebruikt binnen een template van de CollectionView
- Alternatief voor ContextMenu



CollectionView: RefreshView

- CollectionView kan genest worden binnen een RefreshView
- “Pull to refresh” functionaliteit
- Biedt de mogelijkheid om data opnieuw te laden (“refreshen”)

