**Xamarin Forms 4.0 Pre**

Hello guys in this blog I will be explaining a new feature of xamarin forms 4.0 preview so one of the control is a collection view, it is a very beautiful control which I need to use,

The Collection View is a very successor of intended to the listview, improve the design and reducing the complexity and collection view is very flexible.

If you want to use the collection view in our project so you will make sure to initialize the in (xamarin android) MainActivity.cs and (xamarin ios) AppDelegate.cs project.

global::Xamarin.Forms.Forms.SetFlags("CollectionView\_Experimental");

if you want to update an existing project to the Xamarin.Forms 4.0- NuGet (available via your favorite NuGet package manager using the pre-release option).

A Basic Layout

In Xamarin Forms 4.0 review, One of the biggest changes between ListView and CollectionView is the removal of wrapping content in a ViewCell. This allows for significant gains to performance.

Below is an example layout. In this Example, we create a List<string> of people’s names in your binding context and then bind to it in XAML:

1 <StackLayout>

2 <CollectionView x:Name="CV"

3 ItemsSource="{Binding People}"

4 Margin="10,0,0,0">

5 <CollectionView.ItemTemplate>

6 <DataTemplate>

7 <StackLayout HeightRequest="50" WidthRequest="200" Orientation="Horizontal" Padding="0,5,0,5">

8 <Image Source="person"/>

9 <Label Text="{Binding}" VerticalOptions="Center"/>

10 </StackLayout>

11 </DataTemplate>

12 </CollectionView.ItemTemplate>

13 </CollectionView>

14 </StackLayout>

You have a simple list, this new CollectionView is familiar to the ListView you’ve used in the past. Let’s look now at what else it can do.

**A Grid Layout**

By default, CollectionView represent a linear layout. That means you can not only do vertical lists, but horizontal as well, This is accomplished by setting the ItemsLayout property to a new instance of ListItemsLayout which takes an ItemsLayoutOrientation parameter. You can use as horizontal listview.

In this CollectionView we have also another option available: the GridItemsLayout.

Going back to the previous example above, you can use this option to make the CollectionView look a little fancier. Using a GridItemsLayout, you still get a scrollable list, but now with two items in each row:

1 <StackLayout>

2 <CollectionView x:Name="CV" ItemsSource="{Binding People}" VerticalOptions="Center" HorizontalOptions="Center" Margin="10,0,10,0">

3 <CollectionView.ItemsLayout>

4 <GridItemsLayout Orientation="Horizontal" Span="2"/>

5 </CollectionView.ItemsLayout>

6 <CollectionView.ItemTemplate>

7 <DataTemplate>

8 <Frame BorderColor="LightGray" CornerRadius="3" HasShadow="False">

9 <Grid>

10 <Grid.ColumnDefinitions>

11 <ColumnDefinition Width="Auto"/>

12 <ColumnDefinition Width="100" />

13 </Grid.ColumnDefinitions>

14 <Grid.RowDefinitions>

15 <RowDefinition Height="Auto"/>

16 </Grid.RowDefinitions>

17 <Image Grid.Column="0" Source="person" Aspect="Fill"/>

18 <StackLayout Grid.Column="1">

19 <Label Text="{Binding}" HorizontalOptions="EndAndExpand" VerticalOptions="CenterAndExpand"/>

20 </StackLayout>

21 </Grid>

22 </Frame>

23 </DataTemplate>

24 </CollectionView.ItemTemplate>

25 </CollectionView>

26 </StackLayout>

In the above code, we use the ItemsLayout in the XAML. This is the Setting of Orientation to Vertical indicates the direction in which the CollectionView expands, and the Span property sets the number of items per row.

**Use Like CarouselView**

The previous example in which we show the simple scrollable grid of people, But If I want to show only one items at a time in a card format, So You can use CarouselView, a separate control which uses CollectionView as its basis, can be used to adjust your layout, and add in some business card-like information:

1 <CarouselView x:Name="CV"

2 ItemsSource="{Binding People}"

3 HeightRequest="200"

4 HorizontalOptions="Center"

5 VerticalOptions="CenterAndExpand"

6 Margin="10">

7 <CarouselView.ItemsLayout>

8 <GridItemsLayout Orientation="Horizontal"/>

9 </CarouselView.ItemsLayout>

10 <CarouselView.ItemTemplate>

11 <DataTemplate>

12 <Frame BorderColor="LightGray" CornerRadius="3" HasShadow="False">

13 <Grid>

14 <Grid.ColumnDefinitions>

15 <ColumnDefinition Width="75"/>

16 <ColumnDefinition Width="\*" />

17 </Grid.ColumnDefinitions>

18 <Grid.RowDefinitions>

19 <RowDefinition Height="\*"/>

20 <RowDefinition Height="\*"/>

21 </Grid.RowDefinitions>

22 <Image Grid.Column="0" Grid.Row="0" Source="person" VerticalOptions="Start"/>

23 <StackLayout Grid.Column="1" Grid.Row="1" HorizontalOptions="EndAndExpand" VerticalOptions="EndAndExpand">

24 <Label Text="{Binding}" FontSize="24" HorizontalOptions="EndAndExpand"/>

25 <Label Text="Company Address" HorizontalOptions="EndAndExpand"/>

26 <Label Text="City, State" HorizontalOptions="EndAndExpand"/>

27 </StackLayout>

28 </Grid>

29 </Frame>

30 </DataTemplate>

31 </CarouselView.ItemTemplate>

32 </CarouselView>

**It’s a Snap to Use**

In CollectionView One of the great features is SPAN, span have the ability to add snap points. For the page you just designed, it allows to stop the middle part of the card. What if you want to show only one card at a time instead? With ItemLayout with SnapLointsAlignment and SnapPointsType properties, you can configure this behavior in a snap (the purpose of the sentence)! To enable this behavior, add the following to your XAML:

<CarouselView.ItemsLayout>

<GridItemsLayout

Orientation="Horizontal"

SnapPointsAlignment="Center"

SnapPointsType="Mandatory"/>

</CarouselView.ItemsLayout>

**It’s so Empty in Here**

Lastly, a common scenario many developers may face is what to display when the ItemsSource is empty. With the CollectionView you can now set any content to the EmptyView of your list for just this purpose. With the XAML below, the string “No items currently exist!” will be shown inside of the CollectionView when the list of people is empty:

<CollectionView.EmptyView>

<Label Text="No items currently exist!"/>

</CollectionView.EmptyView>

You can customize this view you want to like, including interactive content and supporting bindings.