

Displaying Intricate Data in ListViews



Matthew Soucoup

PRINCIPAL

@codemillmatt codemilltech.com



Displaying Data Topics



Grouping related data

Laying out controls with the Grid

Using binding converters to translate data types

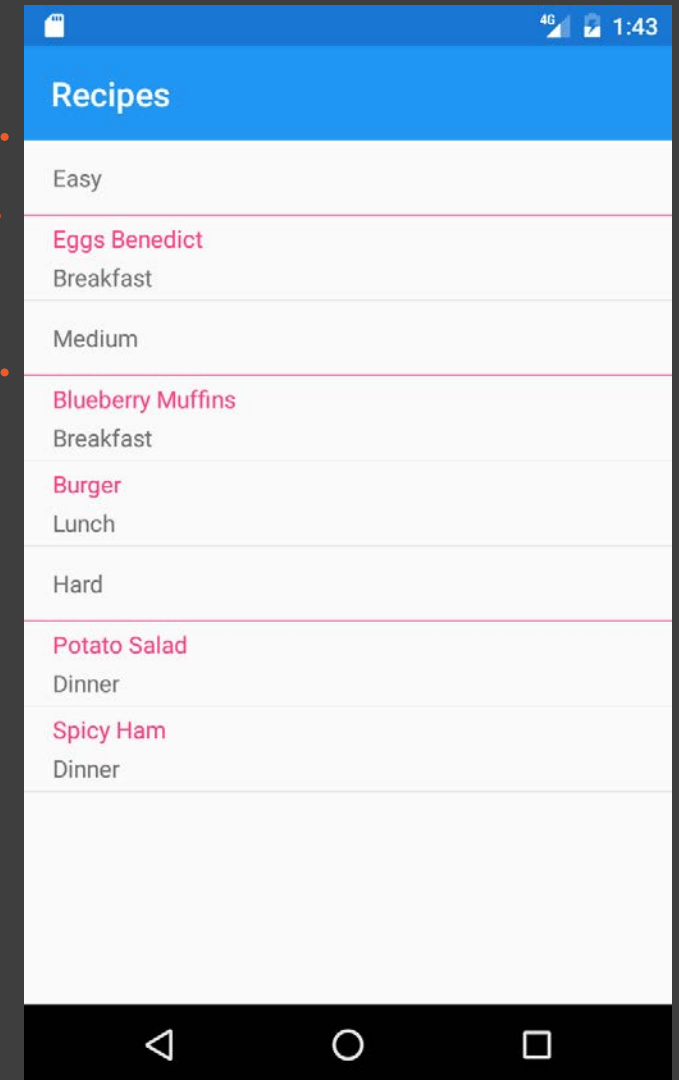
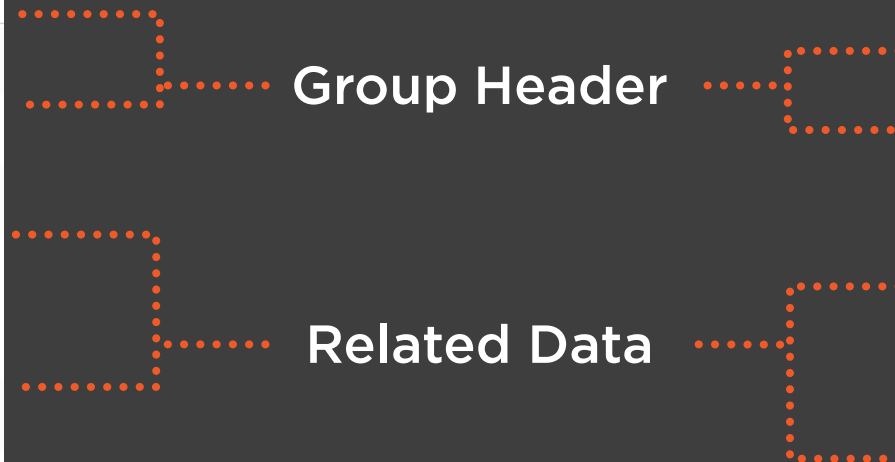
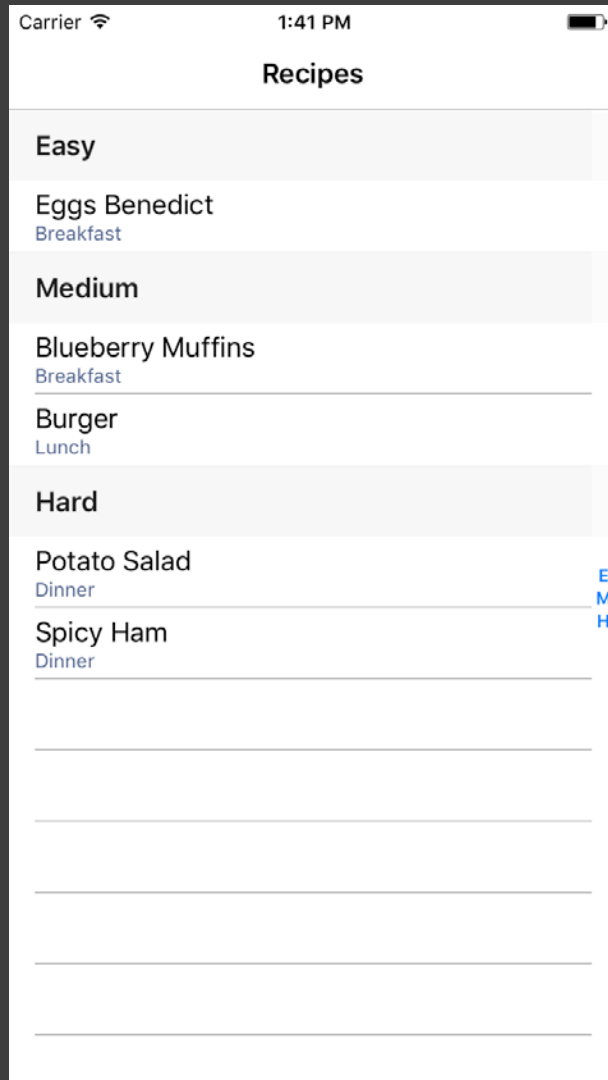
Employing DataTemplateSelectors



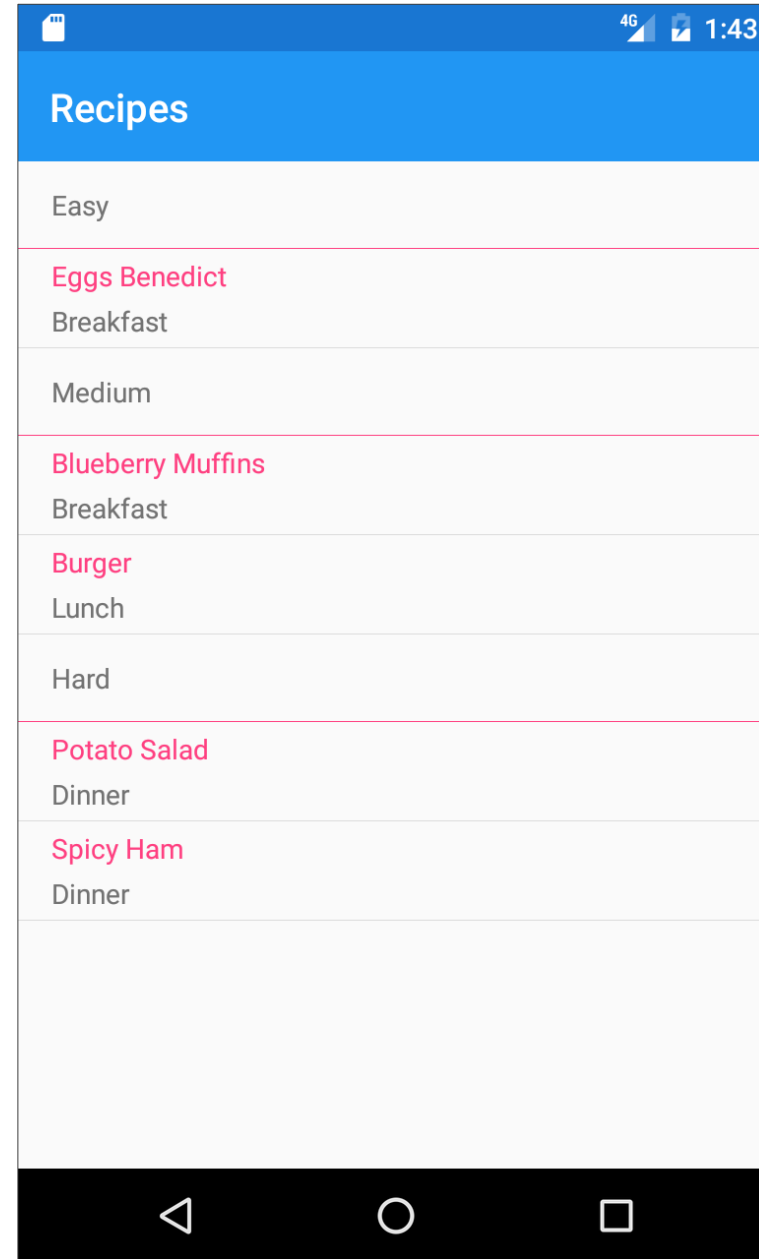
Grouping Related Data



Grouping in ListViews

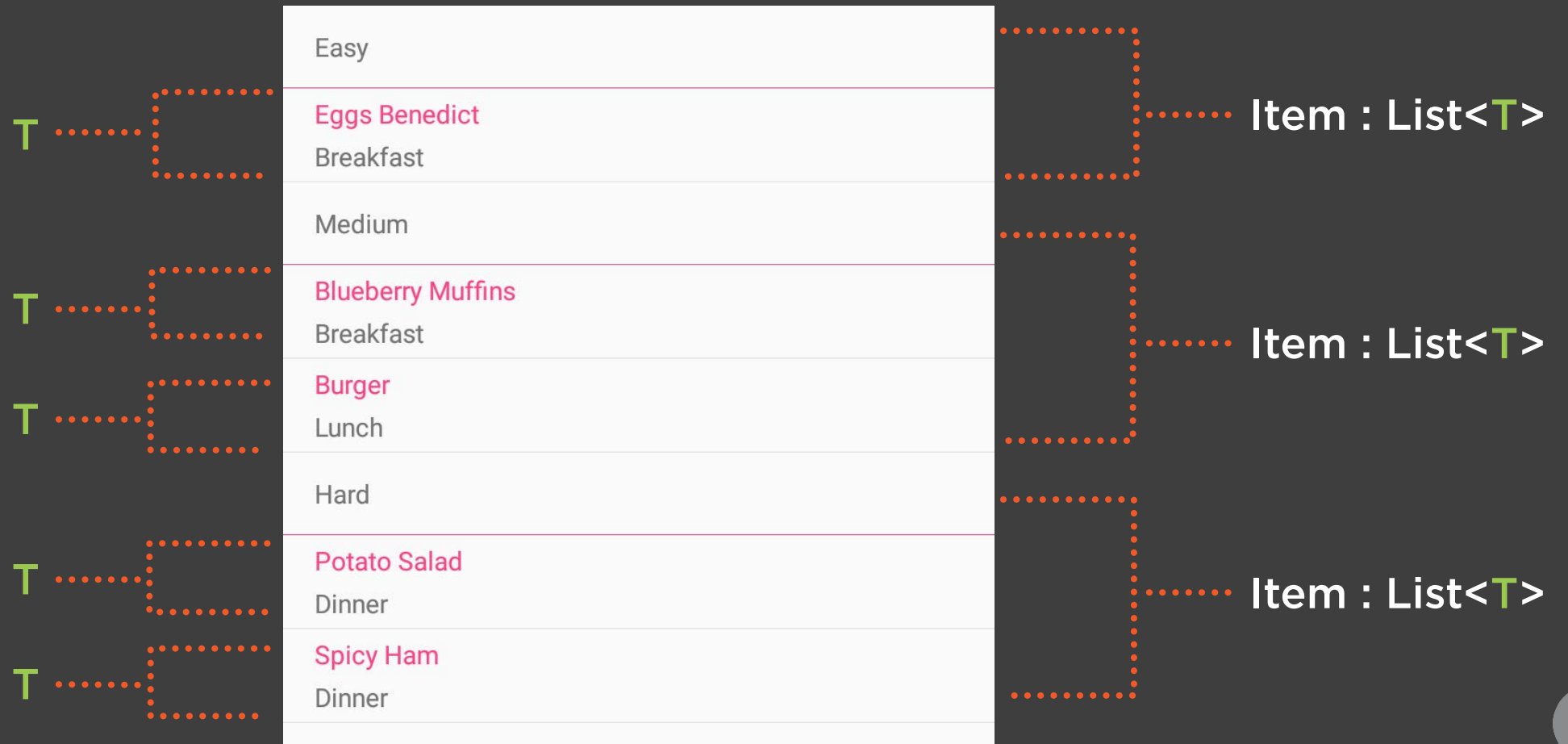


Single ItemsSource
Headers need to bind
Cells need to bind
Accommodate both?



IsGroupingEnabled = true

ItemsSource expects...



Other ListView Grouping Properties

GroupDisplayBinding

GroupShortNameBinding

| | |
|-----------------------------|-----------|
| Easy | |
| Eggs Benedict | Breakfast |
| Breakfast Potatoes | Breakfast |
| French Toast | Breakfast |
| Medium | |
| Blueberry Muffins | Breakfast |
| Burger | Lunch |
| Hard | |
| Potato Salad | Dinner |
| Spicy Ham | Dinner |
| Difficult | |
| Sausage and Mushroom Strata | Breakfast |
| Braised Lamb | Dinner |
| Sweet Potato and Tofu | |



Demo



Create a grouping class

Implement grouping in a ListView

Create a generic/reusable grouping class

Forge custom group headers



ListView Grouping Summary



Group related data together

Properties

- IsGroupingEnabled
- GroupDisplayBinding
- GroupShortNameBinding


Bind to a list of lists

Custom headers

Grid Layout



Grid Layout

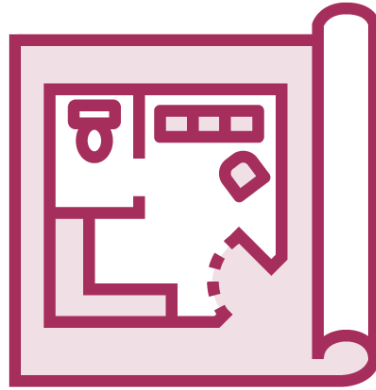
| Breakfast | | |
|---|-------------|----------------------------------|
|  <h2>Blueberry Muffins</h2> | | |
| 10 min prep | 25 min cook | serves 12 |
| | | View Recipe > |



Grid



Easy to build with



Explicitly for 2D



No visible manifestation



```
<Grid.RowDefinitions>  
  <RowDefinition Height="30" />  
  <RowDefinition Height="Auto" />  
  <RowDefinition Height="*" />  
  <RowDefinition Height="2*" />  
</Grid.RowDefinitions>
```

Declaring Rows

RowDefinitions collection

One RowDefinition object per row

Height property



```
<Grid.ColumnDefinitions>  
  <ColumnDefinition Width="Auto" />  
  <ColumnDefinition Width="45" />  
  <ColumnDefinition Width="*" />  
</Grid.ColumnDefinitions>
```

Declaring Columns

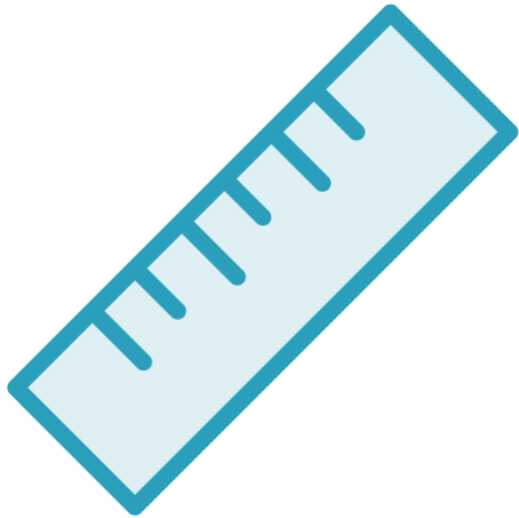
ColumnDefinitions collection

One ColumnDefinition object per column

Width property



Row and Column Sizing



GridUnitType Enum

Absolute

- Specify device-specific units

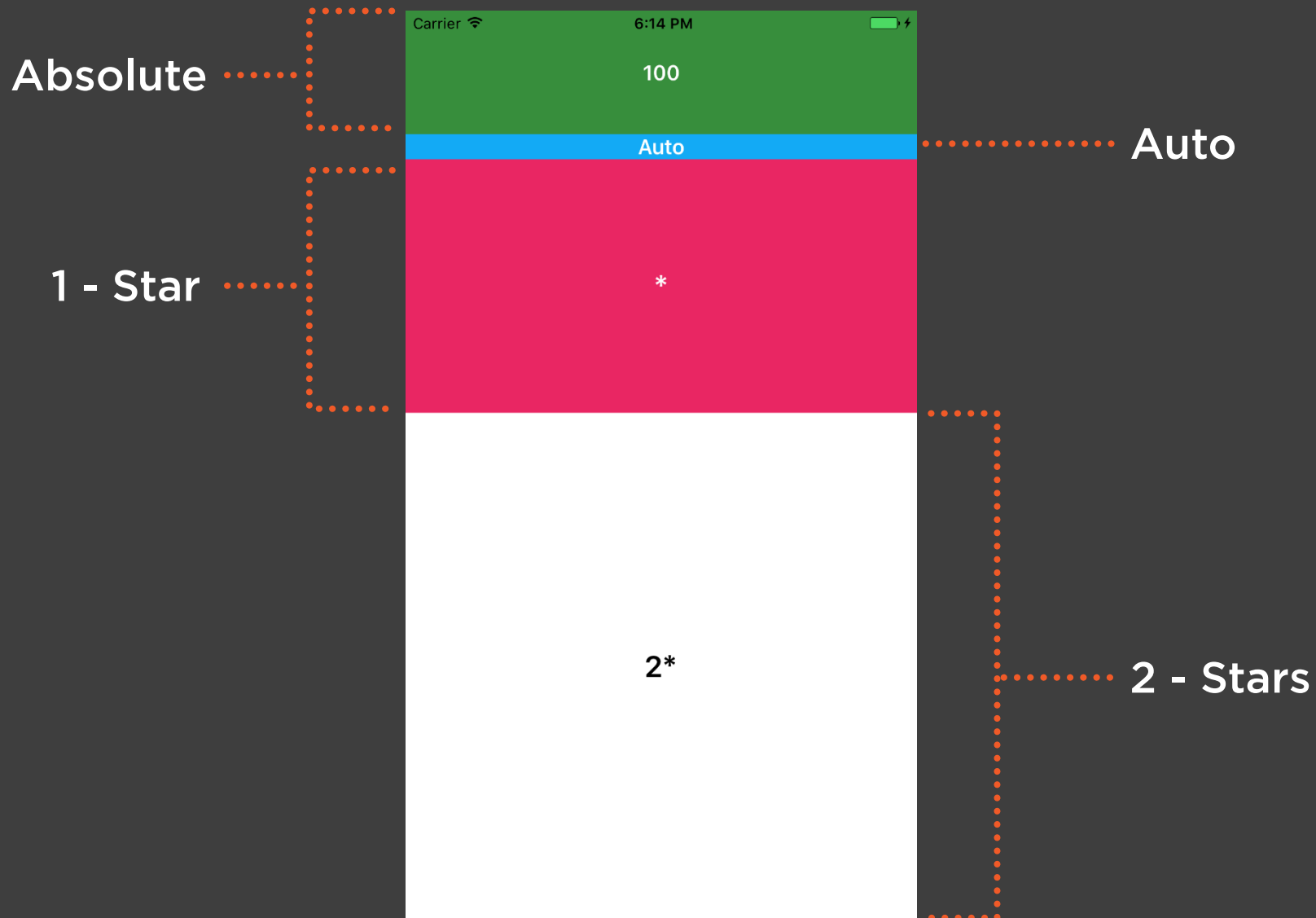
Auto

- Grid chooses size to fit children

Star

- Proportionally allocates area

GridUnitType



Populating a Grid

```
<Grid RowSpacing="3" ColumnSpacing="3">
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto" />
    <RowDefinition Height="30" />
    <RowDefinition Height="*" />
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="*" />
  </Grid.ColumnDefinitions>
  <Label Text="Hello" BackgroundColor="#3F51B5" TextColor="White"
    Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="3" />
  <BoxView Color="#757575"
    Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="2" />
  <BoxView Color="#4CAF50"
    Grid.Row="1" Grid.Column="2" Grid.RowSpan="2" />
</Grid>
```



Grid Spacing



Demo



Build a Grid

Arrange visual elements

Incorporate into app



Grid Summary



Two dimensional layout

Row and column collections

Height and width

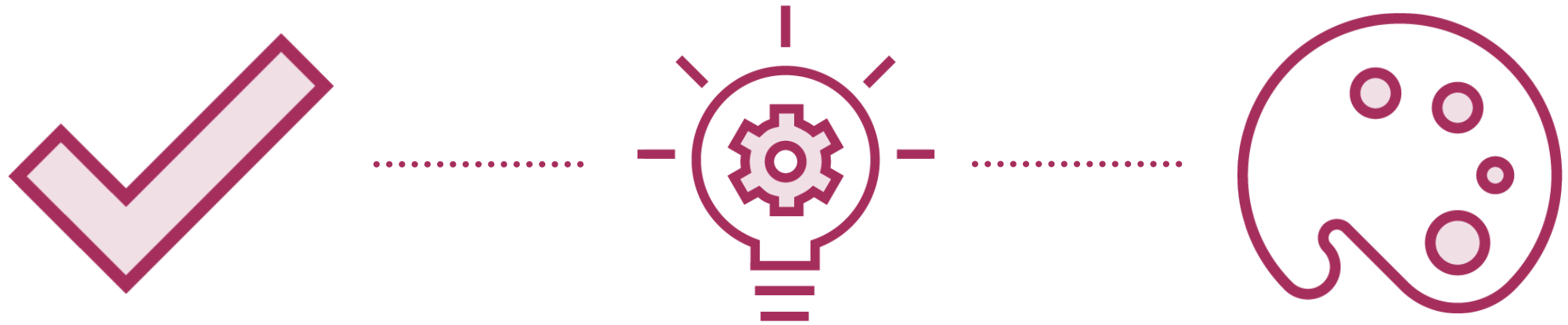
- Absolute
- Auto
- Star

Attached properties to place

Binding Converters



Binding Converters



Original data type

Binding converter

New data type -
meant for display



```
public class RecipeToImageValueConverter : IValueConverter
{
    public object Convert(object value, Type targetType,
                        object parameter, CultureInfo culture) { }

    public object ConvertBack(object value, Type targetType,
                        object param, CultureInfo culture) { }
}
```

IValueConverter

Two Functions

- Convert, ConvertBack

Parameters

- value, targetType, parameter, culture



Recipe Image Name to Image - Convert

```
// Helper property set elsewhere
public string Assembly { get; set; }

public object Convert(object value, Type targetType,
                      object parameter, CultureInfo culture)
{
    var source = value as string;
    if (string.IsNullOrEmpty(source))
        return null;

    var path = $"{Assembly}.{source}";

    return ImageSource.FromResource(path);
}
```



Recipe Image Name to Image - ConvertBack

```
public object ConvertBack(object value, Type targetType,  
                          object param, CultureInfo culture)  
{  
    throw new NotImplementedException();  
}
```



Demo



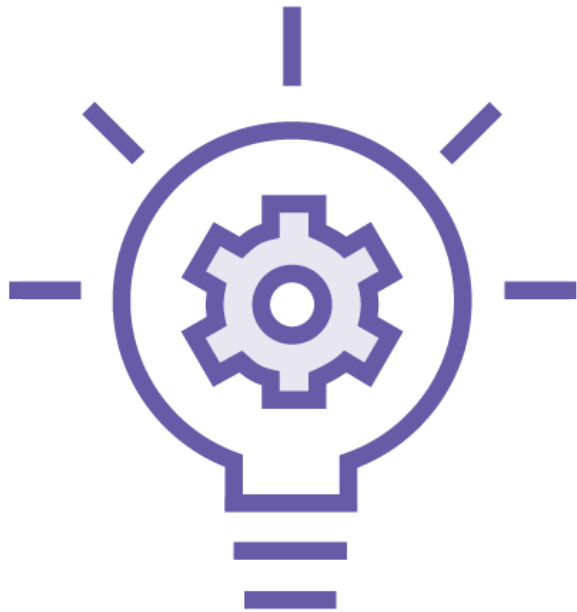
Create a binding converter

Implement a binding converter

Create reusable binding converter



Binding Converters Summary



Convert data types for display

IValueConverter

- Convert
- ConvertBack

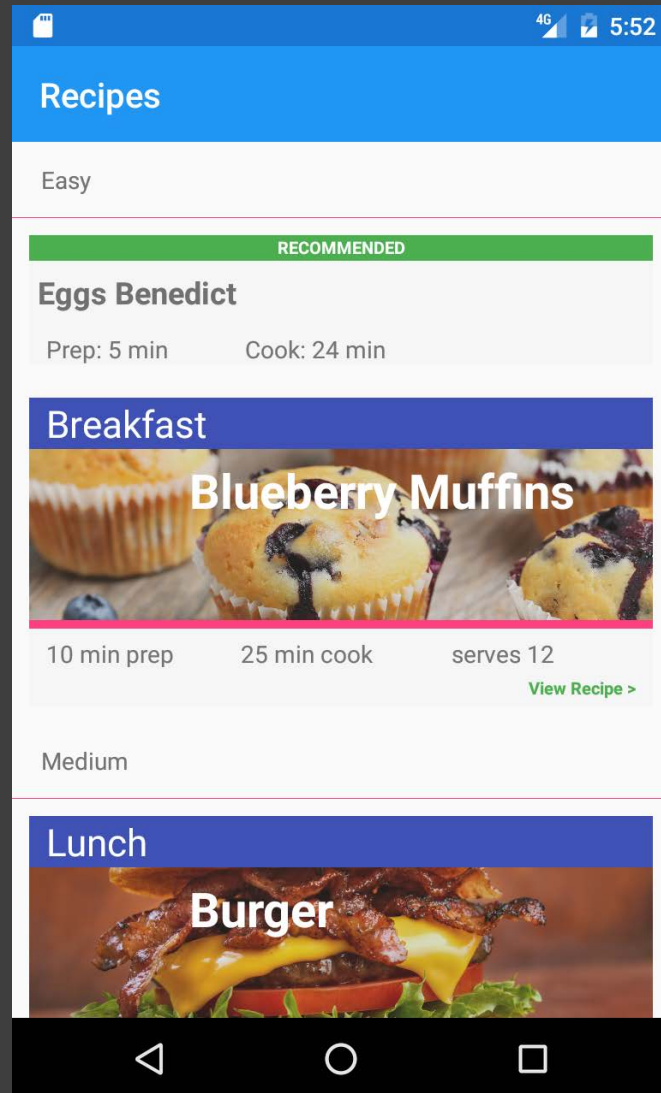
Specific or generic

DataTemplateSelector



DataTemplateSelector

Recommended Recipe



Recipe



```
public class RecipeDataTemplateSelector : DataTemplateSelector
{
    protected override DataTemplate OnSelectTemplate(
        object item, BindableObject container)
    {
    }
}
```

DataTemplateSelector

OnSelectTemplate

- DataTemplate return contains ViewCell
- item parameter is data being bound



```
<ResourceDictionary>
    <local:RecipeDataTemplateSelector x:Key="recipeTempSel" />
</ResourceDictionary>

...

<ListView
    ItemsSource="{x:Static local:RecipeData.AllRecipesGrouped}"
    ItemTemplate="{StaticResource recipeTempSel}" />
```

Consuming DataTemplateSelector

Define in ResourceDictionary

Set in ItemTemplate



Demo



Refactor ViewCells into classes

Create DataTemplateSelector

Implement



DataTemplateSelector Rules

Return same template for
same data

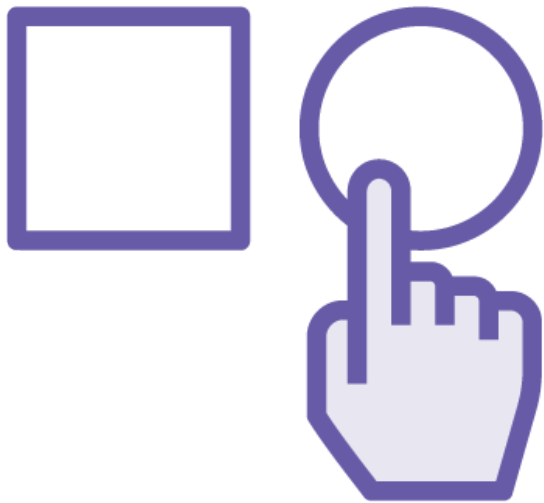
Cannot return
DataTemplateSelector classes

Same instance of template
should be returned

20 templates max on Android



DataTemplateSelector Summary



Select more than one template at runtime

Subclass DataTemplateSelector

- OnSelectTemplate

Several rules to follow



Simple, meaningful, beautiful

Grouping for related data

Grid for elegant and easy layout

Binding converters for data type translation

DataTemplateSelector for multiple
templates at runtime

