



JETBRAINS



Reactor



# Avalonia Value Converter

Juster zhu (朱震)

Microsoft MVP | Huawei HCDE | GeneralUpdate Owner

# 本期内容

## 1. 值转换器

# 数据转换

为什么需要数据转换？

数据源（如ViewModel）与UI控件的数据类型/格式可能不匹配（如bool转FontWeight）

需将原始数据转换为可视化形式（如枚举值转文字描述、数字转颜色）

多数据源聚合计算（如RGB三通道合成颜色）

转换器（Converter）的作用：

实现IValueConverter接口：处理单向或双向绑定中的类型转换

实现IMultiValueConverter接口：处理多绑定（MultiBinding）的聚合逻辑

# 数据转换

Avalonia提供常用内置转换器，位于命名空间Avalonia.Data.Converters：

## 1.布尔转换器

BoolConverters.And/Or：多条件逻辑判断

StringConverters.IsNotNullOrEmpty：控制控件显隐

```
<TextBlock IsVisible="{Binding Text, Converter={x:Static StringConverters.IsNotNullOrEmpty}}"/>
```

## 2.空值检查转换器

ObjectConverters.IsNull：数据为空时返回true

常用于显示占位内容：

```
<Image Source="{Binding Avatar}" IsVisible="{Binding Avatar, Converter={x:Static ObjectConverters.IsNotNull}}"/>
```

```
<TextBlock Text="无头像" IsVisible="{Binding Avatar, Converter={x:Static ObjectConverters.IsNull}}"/>
```

# 单值转换器 (IValueConverter)

场景示例：根据布尔值切换字体粗细

步骤1：创建转换器类

```
public class FontWeightConverter : IValueConverter
{
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
    {
        return (bool)value ? FontWeight.Bold : FontWeight.Normal;
    }
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
    {
        return (FontWeight)value == FontWeight.Bold; // 双向绑定需实现
    }
}
```

# 单值转换器 (IValueConverter)

场景示例：根据布尔值切换字体粗细

步骤2：在XAML中声明资源

```
<Window.Resources>  
    <local:FontWeightConverter x:Key="BoldConverter"/>  
</Window.Resources>
```

步骤3：绑定使用

```
<TextBlock Text="重要标题" FontWeight="{Binding IsImportant, Converter={StaticResource BoldConverter}}"/>
```

# 多值转换器 (IMultiValueConverter)

场景示例：RGB三通道合成画刷颜色

转换器实现：

```
public class RgbToBrushConverter : IMultiValueConverter
{
    public object Convert(IList<object> values, Type targetType, object parameter, CultureInfo culture)
    {
        if (values.Count != 3 || values.Any(v => v is not decimal))
            return Brushes.Black; // 安全回退
        var r = (byte)(decimal)values[0];
        var g = (byte)(decimal)values[1];
        var b = (byte)(decimal)values[2];
        return new SolidColorBrush(Color.FromRgb(r, g, b));
    }
}
```

# 多值转换器 (IMultiValueConverter)

XAML多绑定配置:

```
<TextBlock>  
  <TextBlock.Foreground>  
    <MultiBinding Converter="{StaticResource RgbConverter}">  
      <Binding Path="RedChannel" />  
      <Binding Path="GreenChannel" />  
      <Binding Path="BlueChannel" />  
    </MultiBinding>  
  </TextBlock.Foreground>  
</TextBlock>
```



# 小结

- 1.转换器的概念
- 2.单值转换器
- 3.多值转换器