

Extensões de marcação XAML

10 minutos

Grande parte de sua definição de XAML será definida no tempo de compilação. Geralmente, você sabe onde os elementos devem ficar posicionados, quais cores e fontes serão usadas e quais valores literais devem ser atribuídos às propriedades.

No entanto, às vezes você precisa definir um valor da propriedade para um valor que não pode ser determinado em tempo de compilação. Esses valores são conhecidos apenas quando o programa está em execução. Nessas situações, você pode criar um objeto que fornece um valor para XAML em runtime. O XAML dá suporte a **Extensões de Marcação** para essa finalidade.

Nesta unidade, você aprenderá a criar e usar extensões de marcação.

O que é uma extensão de marcação?

Uma extensão de marcação é uma classe que você usa em XAML para acessar valores de runtime. Suponha que você tenha vários rótulos definidos em sua interface do usuário XAML e queira definir a propriedade `FontSize` com o mesmo valor em todo o aplicativo para garantir que o estilo de todos os rótulos seja consistente. Você pode definir a propriedade `FontSize` usando XAML, conforme mostrado no exemplo a seguir:

XML

```
<Label Text="Hello, World!"
      Grid.Row="0"
      SemanticProperties.HeadingLevel="Level1"
      FontSize="28"
      HorizontalOptions="CenterAndExpand"/>
```

Você pode repetir essa mesma configuração para cada rótulo, mas e se posteriormente quiser alterar esse valor? Você precisa encontrar todas as instâncias dessa propriedade e fazer a alteração. Além disso, suponha que você não saiba qual valor usar; ele pode ser calculado em runtime com base em fatores como orientação do dispositivo, resolução de tela ou outras considerações. Nesses casos, você precisa de algo mais sofisticado do que um literal embutido em código. Aqui, uma extensão de marcação é útil. Extensões de marcação oferecem flexibilidade no modo de obter um valor usado em XAML.

Criando uma extensão de marcação

Uma extensão de marcação é uma classe que implementa a interface **Microsoft.Maui.Controls.Xaml.IMarkupExtension**. Essa interface define um método, chamado **ProvideValue**, com a seguinte assinatura:

C#

```
public object ProvideValue(IServiceProvider serviceProvider)
{
    ...
}
```

A finalidade desse método é fornecer um valor para sua marcação XAML. Observe que o tipo de retorno é `object`, portanto, o valor pode ser de qualquer tipo desde que seja apropriado para o local em que está sendo usado. Por exemplo, em uma extensão de marcação que calcula e retorna um tamanho de fonte, o tipo de retorno deve ser um `double`.

O parâmetro **serviceProvider** contém informações contextuais sobre onde a extensão de marcação está sendo usada no código XAML. Entre outras informações, ele identifica o controle ao qual a extensão está sendo aplicada.

Você pode manter a extensão de marcação para a propriedade simples `FontSize`. No exemplo a seguir, a classe **MainPage** expõe um campo `double` chamado **MyFontSize**. A classe **GlobalFontSizeExtension** implementa a interface **IMarkupExtension**, e o método **ProvideValue** retorna o valor da variável **MyFontSize**:

C#

```
namespace MyMauiApp;

public partial class MainPage : ContentPage
{
    public const double MyFontSize = 28;

    public MainPage()
    {
        InitializeComponent();
        ...
    }
    ...
}

public class GlobalFontSizeExtension : IMarkupExtension
{
    public object ProvideValue(IServiceProvider serviceProvider)
    {
```

```
        return MainPage.MyFontSize;  
    }  
}
```

⚠ Observação

O campo **MyFontSize** deve ser um membro `static` da classe **MainPage** para permitir que ele seja referenciado no método **ProvideValue** de sua maneira. Uma boa prática é que, nesse caso, a variável também deve ser uma constante. Um valor de `const` é `static`.

O método **ProvideValue** também pode fazer ajustes no valor retornado dependendo da orientação e do fator forma do dispositivo.

Aplicando a extensão de marcação a um controle em XAML

Para usar a extensão de marcação no código XAML, adicione o namespace que contém a classe **GlobalFontSizeExtension** à lista de namespaces na marca `ContentPage`. No exemplo a seguir, esse namespace recebe o alias **mycode**:

XML

```
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"  
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"  
             xmlns:mycode="clr-namespace:MyMauiApp"  
             x:Class="MyMauiApp.MainPage">
```

Você pode usar a extensão de marcação para definir a propriedade `FontSize` dessa forma. Observe que a convenção é uma extensão de marcação tenha o sufixo **Extension** no nome. O XAML reconhece esse sufixo e você não precisa incluí-lo ao chamar a extensão do código XAML. No exemplo abaixo, a classe **GlobalFontSizeExtension** é referenciada simplesmente como **GlobalFontSize**:

XML

```
<Label Text="Hello, World!"  
       Grid.Row="0"  
       SemanticProperties.HeadingLevel="Level1"  
       FontSize="{mycode:GlobalFontSize}"  
       HorizontalOptions="CenterAndExpand"/>
```

Você pode aplicar a mesma extensão de marcação em todo o código XAML para qualquer controle que precise especificar o tamanho da fonte. Posteriormente, se você decidir alterar o tamanho da fonte, só precisará modificar a definição da variável **MyFontSize** na classe **MainPage**.

A classe StaticExtension

Por mais que a extensão de marcação **GlobalFontSize** seja útil, é improvável que você crie essa extensão. O motivo para isso é simples; o .NET MAUI já fornece uma extensão mais generalizada que permite que você faça referência a qualquer valor estático em seu código. Essa extensão é denominada **StaticExtension**, ou **Static** para abreviar. O código a seguir mostra a estrutura de tópicos básica desta classe de extensão:

C#

```
[ContentProperty ("Member")]
public class StaticExtension : IMarkupExtension
{
    public string Member {get; set;}
    public object ProvideValue (IServiceProvider serviceProvider)
    {
        ...
    }
}
```

📌 Observação

A finalidade das extensões de marcação personalizadas é permitir que você lide com situações mais complexas em vez do caso estático simples. Por exemplo, talvez você precise alterar dinamicamente o tamanho da fonte com base no fator forma do dispositivo.

Para usar essa classe no código XAML, forneça o nome da variável estática que deseja referenciar na propriedade **Member**, e o método **ProvideValue** retornará o valor nessa variável. O seguinte exemplo ilustra como usá-lo:

XML

```
<Label Text="Hello, World!"
        Grid.Row="0"
        SemanticProperties.HeadingLevel="Level1"
        FontSize="{x:Static Member=mycode:MainPage.MyFontSize}"
        HorizontalOptions="CenterAndExpand"/>
```

O .NET MAUI fornece um conjunto de outras classes de extensão de marcação que você pode usar para cenários como vinculação de dados, referência a estilos e recursos dinâmicos e manipulação de matrizes de dados.

Verificação de conhecimento

1. Qual extensão de marcação permite que você defina uma propriedade XAML em um valor estático definido em uma classe code-behind? *

☐ StaticResource

☐ Associação

☒ Estático

✓ Correto. Faça referência a ele como `x:Static` no código XAML.

2. Qual interface que você usa para criar uma extensão de marcação personalizada? *

☐ IMarkup

☒ IMarkupExtension

✓ Correto.

☐ IValueConverter

Unidade seguinte: Valores específicos de plataforma em XAML

Continuar >