

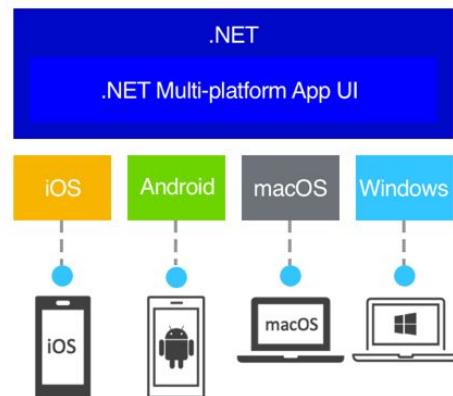
.NET MAUI

FmgLib.MauiMarkup ile sıfırdan .NET MAUI

.NET MAUI Nedir?

.NET Multi-platform App UI (.NET MAUI), C# ve XAML ile yerel mobil ve masaüstü uygulamaları oluşturmaya yönelik platformlar arası bir çerçevedir.

.NET MAUI kullanarak, tek bir kod tabanından Android, iOS, macOS ve Windows üzerinde çalışabilen uygulamalar geliştirebilirsiniz.



.NET MAUI Desteklenen Platformlar

.NET MAUI uygulamaları aşağıdaki platformlar için yazılabilir:

- Android 5.0 (API 21) veya üzeri,
- iOS 11 veya üzeri,
- Mac Catalyst kullanarak macOS 11 veya üzeri,
- Windows 11 ve Windows 10 sürüm 1809 veya üzeri gerektirir

.NET MAUI Blazor uygulamaları aşağıdaki ek platform gereksinimlerine sahiptir:

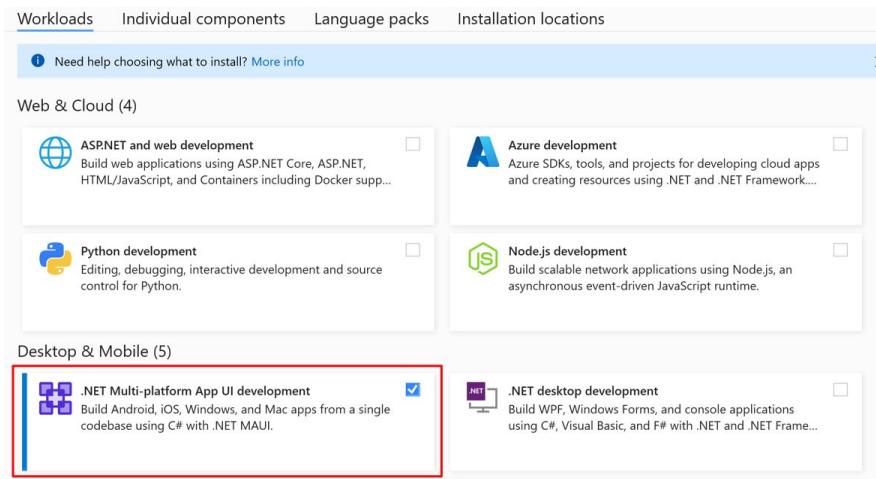
- Android 7.0 (API 24) veya üzeri,
- iOS 14 veya üzeri,
- Mac Catalyst kullanarak macOS 11 veya üzeri gerektirir,

.NET MAUI Blazor uygulamaları güncelleştirilmiş platforma özgü WebView denetimi gerektirir.

.NET MAUI Yükleme

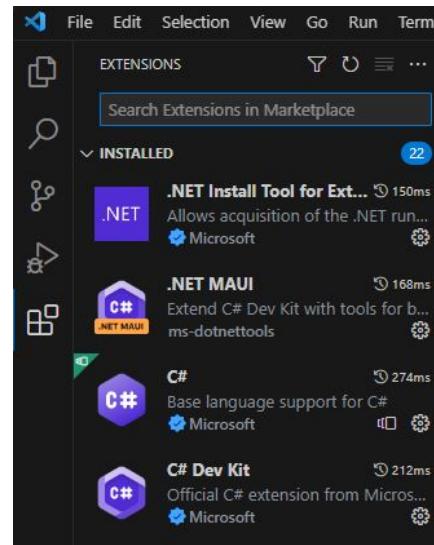
.NET MAUI uygulamaları geliştirmek için **Visual Studio 2022 17.8** veya üzeri ya da **.NET MAUI uzantısına sahip** en son **Visual Studio Code** gereklidir.

Visual Studio için; Visual Studio Installer'da aşağıdaki seçeneği seçmeniz yeterli olacaktır:



.NET MAUI Yükleme

Visual Studio Code için; Uzantılar sekmesinde .NET MAUI diye aratıp aşağıdaki görseldekileri indirebilirsiniz:



FmgLib.MauiMarkup Nedir?

FmgLib.MauiMarkup, .NET MAUI için özel olarak tasarlanmış bir kütüphanedir. Bu kütüphane, XAML dilini kullanma zorunluluğu olmaksızın doğrudan C# kodu ile kodlama yapmanızı sağlar.

Geliştiricilere, C# kodu kullanarak kullanıcı arayüzleri oluşturma konusunda basit ve esnek bir yaklaşım sunar. FmgLib.MauiMarkup ile artık uygulama arayüzlerini kod odaklı bir şekilde geliştirebilir, XAML dosyalarıyla ilgili karışıklıklarla uğraşmanın önüne geçebilirsiniz. Bu kütüphane, geliştirme sürecinizi hızlandırırken daha okunabilir ve yönetilebilir kod yazmanızı sağlar.

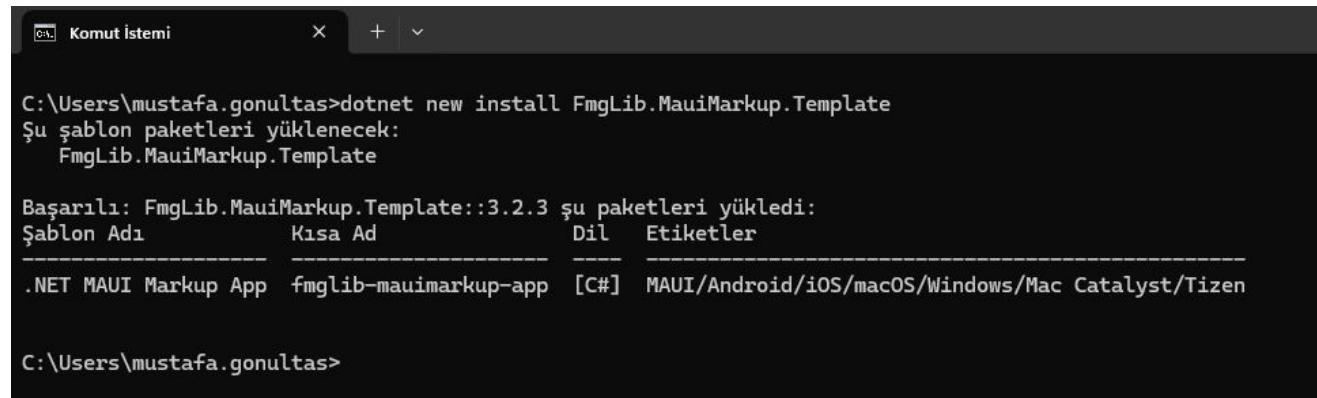
FmgLib.MauiMarkup, XAML tarafında bir Görünüm(View) için sağlanan tüm özellikler için uzantı metotları sunar.



FmgLib.MauiMarkup İle Yeni Proje

FmgLib, FmgLib.MauiMarkup ile yeni bir proje başlatmak için bir proje şablonu sağlar. Bu şablonu kullanabilmek için bilgisayarınızdan cmd(Komut İstemi)'yi açarak aşağıdaki dotnet komutunu çalıştırın:

```
dotnet new install FmgLib.MauiMarkup.Template
```



The screenshot shows a Windows Command Prompt window titled "Komut İstemi". The user has run the command "dotnet new install FmgLib.MauiMarkup.Template". The output indicates that the template "FmgLib.MauiMarkup.Template" is being installed. It shows the successful download of version 3.2.3, which includes the ".NET MAUI Markup App" template with the alias "fmglib-mauimarkup-app". The command prompt then ends with "C:\Users\mustafa.gonultas>".

```
C:\Users\mustafa.gonultas>dotnet new install FmgLib.MauiMarkup.Template
Şu şablon paketleri yüklenecek:
FmgLib.MauiMarkup.Template

Başarılı: FmgLib.MauiMarkup.Template::3.2.3 şu paketleri yükledi:
Şablon Adı           Kısa Ad           Dil   Etiketler
-----  -----  -----
.NET MAUI Markup App  fmglib-mauimarkup-app [C#]  MAUI/Android/iOS/macOS/Windows/Mac Catalyst/Tizen

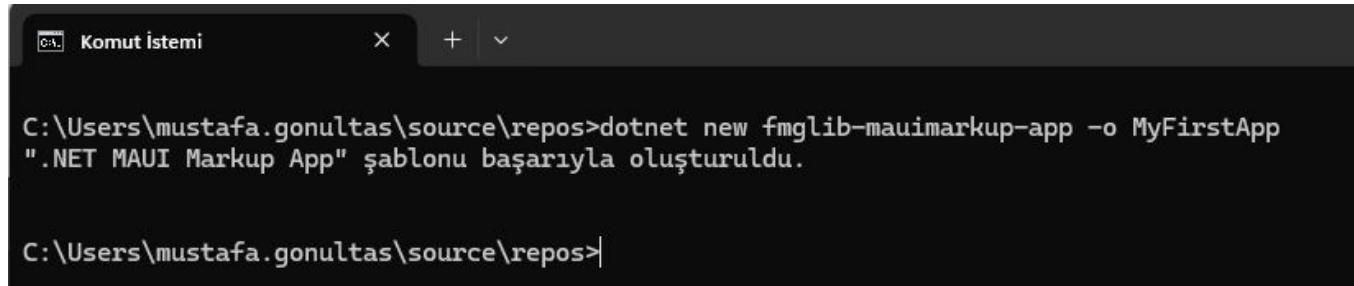
C:\Users\mustafa.gonultas>
```

Yeni bir proje oluşturmak için artık hazırız.

FmgLib.MauiMarkup İle Yeni Proje

FmgLib.MauiMarkup.Template şablonunu kullanabilmek için bilgisayarınızdan cmd(Komut İstemi)'yi açarak aşağıdaki dotnet komutunu çalıştırın:

```
dotnet new fmglib-mauimarkup-app -o MyFirstApp
```



```
C:\Users\mustafa.gonultas\source\repos>dotnet new fmglib-mauimarkup-app -o MyFirstApp
".NET MAUI Markup App" şablonu başarıyla oluşturuldu.

C:\Users\mustafa.gonultas\source\repos>
```

Projemiz belirlenen yol içeresine oluşturulmuş oldu. Fakat bunu böyle kullanmak zorunda değilsiniz. Visual Studio üzerinde rahat bir şekilde proje oluşturabilirsiniz.

FmgLib.MauiMarkup İle Yeni Proje

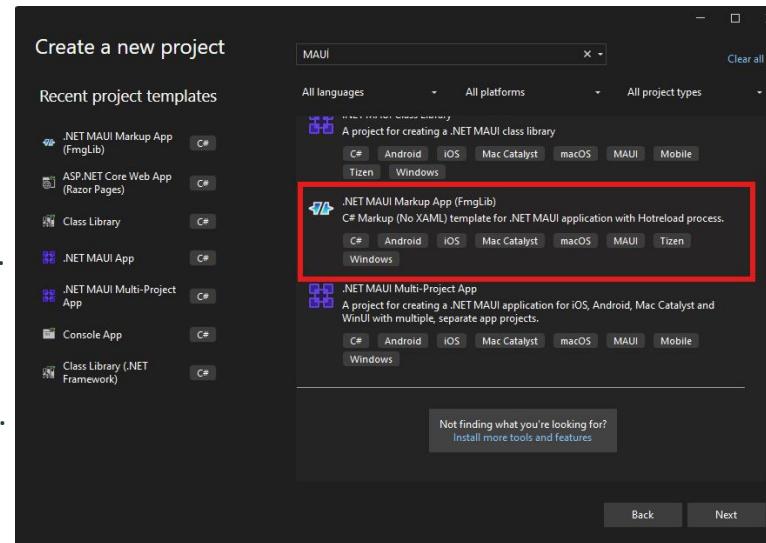
Bir önceki slaytta anlatılanları yaptıktan sonra Visual Studio'yu açtığınızda Yeni Proje Oluşturun seçeneğine tıkladığınızda yeni eklendiği için en üstte sizi karşılıyor olacaktır. aşağıdaki görselde belirtilen şablonu seçip FmgLib.MauiMarkup projesi oluşturabilirsiniz:

Proje oluşturulma adımları şu şekilde:

1- Şekildeki şablon seçilir.

2- Proje adı girilir ve Oluştur butonuna tıklanır.

Artık uygulama geliştirmeye hazır durumdayız.



FmgLib.MauiMarkup'ı Projeye Dahil Etme

MAUI uygulamanızı FmgLib.MauiMarkup NuGet paketini yüklemek için:

<https://www.nuget.org/packages/FmgLib.MauiMarkup> nuget.org üzerinden yardım alabilirsiniz. Visual Studio'da Nuget Package Manager üzerinden de rahatlıkla indirme sağlayabilirsiniz.

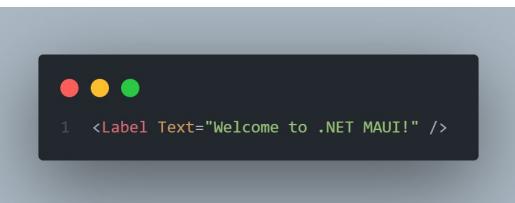
The screenshot displays two side-by-side windows. On the left is the NuGet.org package page for 'FmgLib.MauiMarkup'. It shows the package version 8.6.4, .NET 8.0 support, and a download count of 4.6K. Below the summary are tabs for .NET CLI, Package Manager, PackageReference, Paket CLI, Script & Interactive, and Cake. A command line snippet shows how to add the package to a project: `> dotnet add package FmgLib.MauiMarkup --version 8.6.4`. The 'Dependencies' tab is selected. On the right is a screenshot of the Visual Studio NuGet Package Manager interface. The search bar at the top contains 'FmgLib.MauiMarkup'. In the results list, the 'MyFirstApp' project is expanded, showing its dependencies, properties, platforms, resources, and several C# files: App.cs, AppShell.cs, Imports.cs, MainPage.cs, and MauiProgram.cs. Two packages are listed in the results: 'FmgLib.MauiMarkup' by FmgYazılım (4.59K downloads) and 'FmgLib.MauiMarkup.Template' by FmgYazılım (1.23K downloads).

.NET MAUI Bileşenleri - Label

Label, tek satırlı ve çok satırlı metin görüntüler. Label tarafından görüntülenen metin renklendirilebilir, aralıklı olabilir ve metin süslemeleri olabilir.

Örnek XAML/C#/MauiMarkup kodları şekildeki gibidir:

XAML



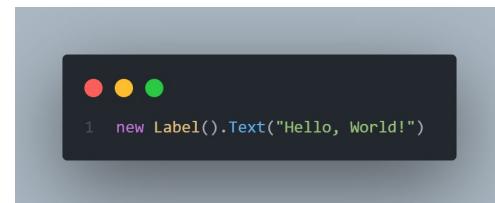
```
1 <Label Text="Welcome to .NET MAUI!" />
```

C#



```
1 new Label{ Text = "Hello, World!" };
```

FmgLib.MauiMarkup



```
1 new Label().Text("Hello, World!")
```

Biz bundan sonraki tüm süreçlerde sadece FmgLib.MauiMarkup ile ilerleyeceğiz. Görüldüğü üzere daha karmaşık kod yapılanmalarında bizi C# ile yazmanın zorluğundan kurtarmış olacak ve XAML kullanımına ve syntax yapısı öğrenme zorunluluğundan kurtarmış olacaktır.

.NET MAUI Bileşenleri - Label

TextColor: Metnin rengini ayarlamamanızı sağlar.



CharacterSpacing: Metnin karakter aralığını ayarlar. double değer alır.



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/label?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - Button

Button, bir görevi gerçekleştirmek için uygulamayı yönlendiren bir tıklama veya dokunuşa yanıt veren bir düğmedir.

Button, parmakla basıldığında veya fareyle tıklandığında o komutu başlatır.

Örnek Button kullanımı:

```
● ● ●  
1 new Button()  
2 .Text("Click me")  
3 .Center()  
4 .OnClicked(OnCounterClicked)
```

```
● ● ●  
1 new Button()  
2 .Text("Click me")  
3 .Center()  
4 .OnClicked((sender, e) =>{  
5     Console.WriteLine(".NET MAUI Markup");  
6 })
```

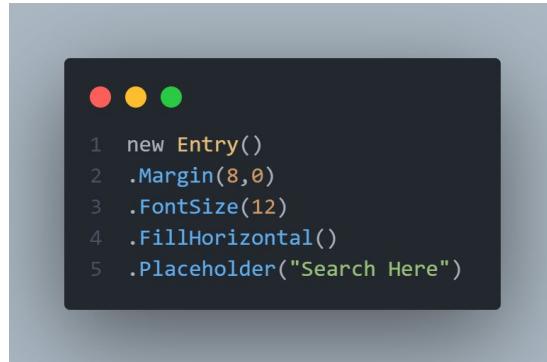
```
● ● ●  
1 new Button()  
2 .Text("OK")  
3 .TextColor(Black)  
4 .FontAttributes(Bold)  
5 .FontSize(15)  
6 .Row(2)  
7 .HeightRequest(30)  
8 .BackgroundColor(DeepSkyBlue)  
9 .CenterHorizontal()  
10 .Padding(0)  
11 .Command(e => e.Path("ClosePopupCommand"))
```

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/button?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - Entry

Entry, tek satırlık metin girmenize ve düzenlemenize olanak tanır. Ayrıca, Entry bir parola alanı olarak da kullanılabilir.

Örnek Entry kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/entry?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - Editor

Editor, birden çok metin satırı girmenize ve düzenlemenize olanak tanır.

Örnek Editor kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/editor?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - Image

Image, yerel bir dosyadan, bir URI'dan veya bir akıştan yüklenebilen bir resmi görüntüler. Standart platform resim formatları, animasyonlu GIF'ler dahil olmak üzere desteklenir ve yerel Scalable Vector Graphics (SVG) dosyaları da desteklenir.

Örnek Image kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/image?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - ProgressBar

ProgressBar, kullanıcılarla uygulamanın uzun bir işlemde ilerlediğini gösterir. ProgressBar, bir double değeri ile temsil edilen yüzdeye göre doldurulan yatay bir çubuktur.

Örnek ProgressBar kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/progressbar?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - ActivityIndicator

ActivityIndicator, uygulamanın uzun bir işlemle meşgul olduğunu göstermek için bir animasyon görüntüleri. ProgressBar'ın aksine, ActivityIndicator ilerleme hakkında herhangi bir bilgi vermez.

Örnek ActivityIndicator kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/activityindicator?view=net-maui-8.0>
bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - CheckBox

CheckBox, işaretlenebilen veya boş olabilen bir düğme türüdür. CheckBox işaretlendiğinde, açık (on) olarak kabul edilir. CheckBox boş olduğunda ise kapalı (off) olarak kabul edilir.

Örnek CheckBox kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/checkbox?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - RadioButton

RadioButton, kullanıcıların bir seçenek grubundan birini seçmesine olanak tanıyan bir button türüdür. Her seçenek bir radio button ile temsil edilir ve bir grup içinde yalnızca bir radio button seçilebilir.

Örnek RadioButton kullanımı:

```
1 new RadioButton()  
2 .IsChecked(true)  
3 .Content("Cat")
```

```
1 new RadioButton()  
2 .IsChecked(false)  
3 .Content(new Image().Source("cat.png"))
```

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/radiobutton?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - Border

Border, başka bir denetimin etrafına bir kenarlık, arka plan veya her ikisini çizen bir konteyner denetimidir. Border yalnızca bir Child(View) nesne içerebilir. Birden fazla nesnenin etrafına kenarlık koymak istiyorsanız, onları bir layout gibi bir konteyner nesnesine sarabilirsiniz.

Örnek Border kullanımı:



```
1 new Border()
2 .Background(Transparent)
3 .StrokeShape(new RoundRectangle().CornerRadius(6,0,0,6))
4 .StrokeThickness(1)
5 .Content(
6     new Image()
7     .Source("car.png")
8     .SizeRequest(165, 240)
9     .Aspect(Aspect.AspectFit)
10 )
```

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/border?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - Frame

Frame, bir görünümü veya layout'u renk, gölge ve diğer seçeneklerle yapılandırılabilen bir kenarlıkla sarmak için kullanılır. Frame'ler, denetimlerin etrafına kenarlık oluşturmak için kullanılabilir, ancak aynı zamanda daha karmaşık UI oluşturmak için de kullanılabilir.

Örnek Frame kullanımı:

```
1 new Frame()
2 .Padding(0)
3 .CornerRadius(20)
4 .HasShadow(false)
5 .SizeRequest(40)
6 .Opacity(10)
7 .Content(
8     new Image()
9     .Source("car.png")
10    .SizeRequest(40,40)
11    .Aspect(Aspect.AspectFit)
12    .Center()
13 )
```

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/frame?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - SearchBar

earchBar, bir arama başlatmak için kullanılan bir kullanıcı giriş denetimidir. SearchBar denetimi, yer tutucu metni, sorgu girişini, arama yürütme ve iptal işlemlerini destekler.

Örnek SearchBar kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/searchbar?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - Slider

Slider, sürekli bir aralıktan bir double değeri seçmek için manipüle edebileceğiniz yatay bir çubuktur.

Örnek Slider kullanımı:

```
1 new Slider()
2 .Maximum(50)
3 .Minimum(10)
4 .OnValueChanged((sender, e) =>{
5     Console.WriteLine("Test");
6 })
```

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/slider?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - ImageButton

ImageButton görünümü, içeriği bir resim olan bir button oluşturmak için Button ve Image görünümlerini birleştirir. ImageButton'a parmakla bastığınızda veya fare ile tıkladığınızda, uygulamayı bir görevi yerine getirmesi için yönlendirir. Ancak, Button'dan farklı olarak, ImageButton görünümünün metin ve metin görünümü kavramı yoktur.

Örnek ImageButton kullanımı:

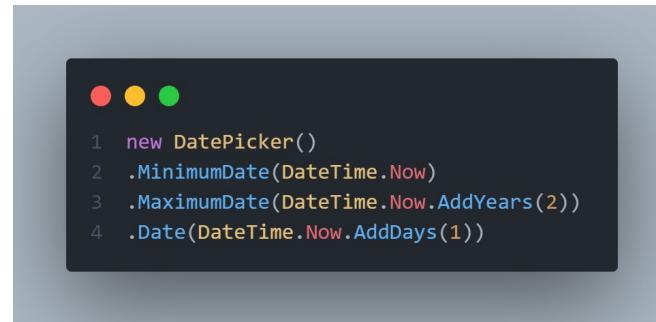


<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/imagebutton?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - DatePicker

DatePicker, platformun tarih seçici kontrolünü çağırır ve bir tarih seçmenizi sağlar.

Örnek DatePicker kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/datepicker?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - TimePicker

TimePicker, platformun zaman seçici kontrolünü çağırır ve bir zaman seçmenizi sağlar.

Örnek TimePicker kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/timepicker?view=net-maui-8.0&tabs=android> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - Picker

Picker, kullanıcıya seçim yapabilmesi için kısa bir öğe listesi gösterir.

Örnek Picker kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/picker?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - Stepper

Stepper, bir değer aralığından sayısal bir değer seçilmesini sağlar. Eksi ve artı işaretleriyle etiketlenmiş iki düğmeden oluşur. Bu düğmeler kullanıcı tarafından bir değer aralığından artımlı olarak bir çift değer seçmek için manipüle edilebilir.

Örnek Stepper kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/stepper?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - Switch

Switch, açık ve kapalı durumlar arasında geçiş yapabilmek için kullanıcı tarafından manipüle edilebilen yatay bir geçiş düğmesidir. Bu durumlar bir boolean değerle temsil edilir.

Örnek Switch kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/switch?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - BoxView

BoxView, belirtilen genişlik, yükseklik ve renkte basit bir dikdörtgen veya kare çizer.

Örnek BoxView kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/boxview?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - ScrollView

ScrollView, içeriğini kaydırabilme özelliğine sahip bir görünüm (view) 'dur. Varsayılan olarak, ScrollView içeriğini dikey olarak kaydırır. Bir ScrollView'un yalnızca tek bir alt ögesi olabilir, ancak bu alt öge başka düzenler (layout) olabilir.

Örnek ScrollView kullanımı:



```
1 new ScrollView()
2   .Content(
3     new Button()
4       .HeightRequest(750)
5       .Text("Example")
6       .FontAttributes(Bold)
7   )
```

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/scrollview?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - RefreshView

RefreshView, kaydırılabilir içerik için çekerek yenileme işlevi sağlayan bir kapsayıcı kontrolüdür. Bu nedenle, bir RefreshView'in alt ögesi ScrollView, CollectionView veya ListView gibi kaydırılabilir bir kontrol olmalıdır.

Örnek RefreshView kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/refreshview?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - ListView

ListView, seçilebilir veri öğelerinin dikey olarak kaydırılabilir bir listesini görüntüler.

ListView listenin görünümünü yönetirken, listedeki her öğenin görünümü, öğeleri göstermek için bir Cell kullanan bir DataTemplate tarafından tanımlanır.

.NET MAUI, metin ve resim kombinasyonlarını görüntülemek için hücre türleri içerir ve istediği üzere herhangi bir içeriği görüntüleyen özel hücreler de tanımlayabilirsiniz. ListView ayrıca başlık ve altbilgi, gruplandırılmış veri, refresh(çek-yenile) ve bağlam menü öğeleri gösterme desteği içerir.

Örnek ListView kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/listview?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

A screenshot of a .NET MAUI application window showing a ListView with multiple items. The items consist of two columns: one containing images and the other containing text. The code for this is shown in the foreground window:

```
1 new ListView()\n2 .ItemsSource(e => e.Path("Monkeys"))\n3 .ItemTemplate(new DataTemplate(() =>\n4     new ViewCell()\n5         .View(\n6             new Grid()\n7                 .RowDefinitions(e => e.Auto().Auto())\n8                 .ColumnDefinitions(e => e.Auto().Auto())\n9                 .Padding(10)\n10                .Children(\n11                    new Image()\n12                        .RowSpan(2)\n13                        .Source(e => e.Path("ImageUrl"))\n14                        .SizeRequest(60)\n15                        .Aspect(Aspect.AspectFill),\n16\n17                    new Label()\n18                        .Column(1)\n19                        .Text(e => e.Path("Name"))\n20                        .FontAttributes(Bold),\n21\n22                    new Label()\n23                        .Row(1)\n24                        .Column(1)\n25                        .Text(e => e.Path("Location"))\n26                        .FontAttributes.Italic)\n27                        .AlignBottom()\n28                )\n29            )\n30        ))
```

The code uses XAML-like syntax for defining the data template and item source.

.NET MAUI Bileşenleri - CollectionView

CollectionView, farklı düzenleme biçimlerini kullanarak veri listelerini göstermek için kullanılan bir görünüm (view) türüdür. ListView'a göre daha esnek ve performanslı bir alternatif olma amacı taşır.

Örnek CollectionView kullanımı:



```
1 new CollectionView()
2 .Row(1)
3 .ItemsSource(e => e.Path("Options"))
4 .ItemsLayout(new LinearItemsLayout(ItemsLayoutOrientation.Horizontal).ItemSpacing(10))
5 .ItemTemplate(
6     new DataTemplate(() =>
7         new Grid().SizeRequest(150,50).Children(
8             new RoundRectangle().CornerRadius(35).Fill("#242529".ToColor()),
9             new Label().Center().TextColor(White).Text(e => e.Path(".")))
10    )
11 )
12 )
```

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/collectionview/?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - CarouselView

CarouselView, kullanıcıların bir öğe koleksiyonunda gezinmek için kaydırabileceği, kaydırılabilir bir düzende veri sunmak için kullanılan bir görünüm (view) türüdür.

Varsayılan olarak, CarouselView öğelerini yatay olarak görüntüler. Ekranda tek bir öğe gösterilecek ve hızlı kaydırma hareketleri koleksiyon içindeki öğeler arasında ileri ve geri navigasyona neden olacaktır. Ek olarak, CarouselView'daki her öğeyi temsil eden göstergeler görüntülenebilir.

Örnek CarouselView kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/carouselview/?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

```
1 new CarouselView()
2 .ItemsSource(e => e.Path("Monkeys"))
3 .ItemTemplate(new DataTemplate() =>
4     new StackLayout()
5         .Children(
6             new Frame()
7                 .HasShadow(true)
8                 .BorderColor(DarkGray)
9                 .CornerRadius(5)
10                .Margin(20)
11                .HeightRequest(300)
12                .Center()
13                .Content(
14                    new StackLayout()
15                        .Children(
16                            new Label()
17                                .Text(e => e.Path("Name"))
18                                .FontAttributes(Bold)
19                                .FontSize(18)
20                                .Center(),
21
22                            new Image()
23                                .Source(e => e.Path("ImageUrl"))
24                                .Aspect(Aspect.AspectFill)
25                                .SizeRequest(150)
26                                .CenterHorizontal(),
27
28                            new Label()
29                                .Text(e => e.Path("Location"))
30                                .CenterHorizontal(),
31
32                            new Label()
33                                .Text(e => e.Path("Details"))
34                                .FontAttributes.Italic)
35                                .CenterHorizontal()
36                                .MaxLines(5)
37                                .LineBreakMode(TailTruncation)
38
39                )
40            )
41 )
```

.NET MAUI Bileşenleri - IndicatorView

IndicatorView, bir CarouselView içindeki öğe sayısını ve mevcut konumu gösteren işaretçileri görüntüleyen bir kontrol ögesidir.

Örnek IndicatorView kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/indicatorview?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - SwipeView

SwipeView, bir içerik öğesini saran ve çekme hareketiyle ortaya çıkarılan bağlam menüsü öğeleri sağlayan bir kapsayıcı kontrolüdür. Örnek SwipeView kullanımı:

- **LeftItems**: Kontrol sola doğru kaydırıldığında çağrılabilen kaydırma öğelerini temsil eden, Swipeltems türünden bir özelliktir.
- **RightItems**: Kontrol sağa doğru kaydırıldığında çağrılabilen kaydırma öğelerini temsil eden, Swipeltems türünden bir özelliktir.
- **TopItems**: Kontrol yukarıdan aşağıya doğru kaydırıldığında çağrılabilen kaydırma öğelerini temsil eden, Swipeltems türünden bir özelliktir.
- **BottomItems**: Kontrol aşağıdan yukarıya doğru kaydırıldığında çağrılabilen kaydırma öğelerini temsil eden, Swipeltems türünden bir özelliktir.
- **Threshold**: Kaydırma hareketini tetikleyerek kaydırma öğelerini tamamen göstermek için gereken cihaz bağımsız birim sayısını temsil eden, double türünden bir özelliktir.

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/swipeview?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.



```
1 new SwipeView()
2 .LeftItems(
3     new SwipeItem()
4         .Text("Delete")
5         .IconImageSource("delete.png")
6         .BackgroundColor(LightPink)
7         .Command(e => e.Path("DeleteCommand"))
8 )
9 .RightItems(
10    new SwipeItem()
11    .Text("Favorite")
12    .IconImageSource("favorite.png")
13    .BackgroundColor(LightGreen)
14    .Command(e => e.Path("FavoriteCommand")),
15
16    new SwipeItem()
17    .Text("Share")
18    .IconImageSource("share.png")
19    .BackgroundColor(LightYellow)
20    .Command(e => e.Path("FavoriteShareCommand"))
21 )
22 .Content(
23     new Grid()
24     .SizeRequest(300,60)
25     .BackgroundColor(LightGray)
26     .Children(
27         new Label()
28         .Text("Swipe")
29         .Center()
30     )
31 )
```

.NET MAUI Bileşenleri - TableView

TableView, kaydırılabilir öğelerden oluşan ve bölgelere ayrılabilen bir tablo görüntüler. Tipik olarak, her satırın farklı bir görünümüne sahip öğeleri göstermek için, örneğin ayarlar tablosu sunmak için kullanılır.

TableView tablonun görünümünü yönetirken, tablo içindeki her öğenin görünümü bir Cell tarafından tanımlanır. .NET MAUI, farklı veri kombinasyonlarını görüntülemek için kullanılan beş hücre türü içerir ve istediğiniz herhangi bir içeriği görüntüleyen özel hücreler de tanımlayabilirsiniz.

Örnek TableView kullanımı:



```
new TableView()
    .Intent(TableIntent.Menu)
    .Root(
        new TableSection("Chapter")
        {
            new TextCell()
                .Text("1. Introduction to .NET MAUI")
                .Detail("Learn about .NET MAUI and what it provides."),
            new TextCell()
                .Text("2. Anatomy of an app")
                .Detail("Learn about the visual elements in .NET MAUI"),
            new TextCell()
                .Text("3. Text")
                .Detail("Learn about the .NET MAUI controls that display text."),
            new TextCell()
                .Text("4. Dealing with sizes")
                .Detail("Learn how to size .NET MAUI controls on screen.")
        }
    )
}
```

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/tableview?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - ContentView

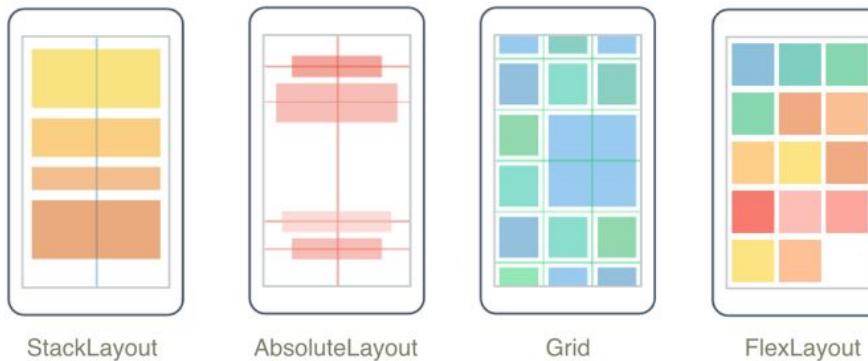
ContentView, özel ve yeniden kullanılabilir kontroller oluşturmanıza olanak tanıyan bir kontrol ögesidir.

ContentView sınıfı, ContentView'in içeriğini temsil eden View türünden bir Content özelliği tanımlar. Bu özellik, veri bağlamalarının hedeflenebilmesi ve stil uygulanabilmesi için BindableProperty nesnesi tarafından desteklenir.

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/contentview?view=net-maui-8.0> bu linkten diğer özelliklere erişebilirsiniz.

.NET MAUI Bileşenleri - Layout

Layout (düzen) sınıfları, uygulamanızdaki UI kontrollerini düzenlemenizi ve grüplamanızı sağlar. Bir layout sınıfı seçmek, layout'un alt öğeleri nasıl konumlandığıını ve boyutlandırdığını bilmeyi gerektirir. Ayrıca, istediğiniz düzeni oluşturmak için layout'ları iç içe yerleştirmeniz gerekebilir.

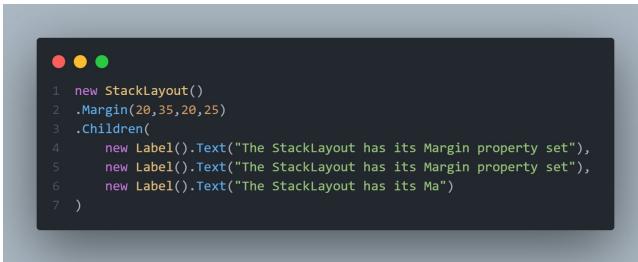


<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/layouts/?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - StackLayout

StackLayout öğeleri tek boyutlu bir yiğin (stack) içerisinde, yatay veya dikey olarak düzenler. **Orientation** özelliği, öğelerin yönünü belirtir ve varsayılan yön Dikey'dir. StackLayout genellikle bir sayfadaki kullanıcı arayüzünün bir alt bölümünü düzenlemek için kullanılır.

Örnek StackLayout kullanımı:



```
1 new StackLayout().Children(
2     new StackLayout()
3         .Orientation(StackOrientation.Horizontal)
4         .Children(
5             new Label().Text("Username:"),  
6             new Entry().Placeholder("Enter your username")
6         ),
7     new StackLayout()
8         .Orientation(StackOrientation.Horizontal)
9         .Children(
10            new Label().Text("Password:"),  
11            new Entry().Placeholder("Enter your password").IsPassword(true)
11        ),
12
13
14
15
16 )
```

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/layouts/stacklayout?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - HorizontalStackLayout

HorizontalStackLayout, alt görünümleri tek boyutlu yatay bir yığın içerisinde düzenler ve **StackLayout'a göre daha performanslı bir alternatif**tir. HorizontalStackLayout genellikle bir sayfadaki kullanıcı arayüzünün bir alt bölümünü düzenlemek için kullanılır.

Örnek HorizontalStackLayout kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/layouts/horizontalstacklayout?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - VerticalStackLayout

VerticalStackLayout, alt görünümleri tek boyutlu dikey bir yiğin içerisinde düzenler ve **StackLayout'a göre daha performanslı bir alternatifdir**. VerticalStackLayout genellikle bir sayfadaki kullanıcı arayüzünün bir alt bölümünü düzenlemek için kullanılır.

Örnek VerticalStackLayout kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/layouts/verticalstacklayout?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Grid

Grid, öğeleri satır ve sütunlarda görüntüleyen, bu satır ve sütunların orantılı veya mutlak boyutlara sahip olabildiği bir düzen (layout) ögesidir. Bir Grid'in satır ve sütunları, **RowDefinitions** ve **ColumnDefinitions** özellikleriyle belirtilir.

Belirli Grid hücrelerine öğeleri yerleştirmek için **Column** ve **Row** ekli (**attached**) özelliklerini kullanılır. Öğelerin birden fazla satır ve sütuna yayılması için **RowSpan** ve **ColumnSpan** ekli (**attached**) özelliklerini kullanılır.

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/layouts/grid?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

```
● ● ●
1 new ListView()
2 ..ItemsSource(e => e.Path("Monkeys"))
3 .ItemTemplate(new DataTemplate(() =>
4     new ViewCell()
5     .View(
6         new Grid()
7             .RowDefinitions(e => e.Auto().Auto())
8             .ColumnDefinitions(e => e.Auto().Auto())
9             .Padding(10)
10            .Children(
11                new Image()
12                    .RowSpan(2)
13                    .Source(e => e.Path("ImageUrl"))
14                    .SizeRequest(60)
15                    .Aspect(Aspect.AspectFill),
16
17                new Label()
18                    .Column(1)
19                    .Text(e => e.Path("Name"))
20                    .FontAttributes(Bold),
21
22                new Label()
23                    .Row(1)
24                    .Column(1)
25                    .Text(e => e.Path("Location"))
26                    .FontAttributes.Italic)
27                    .AlignBottom()
28            )
29        )
30    )))

```

.NET MAUI Bileşenleri - FlexLayout

FlexLayout, alt öğeleri yatay veya dikey olarak bir yığın şeklinde göstermesi bakımından StackLayout'a benzer. Ancak FlexLayout, tek bir satır veya sütuna sığmayacak kadar çok alt öğe olduğunda bunları sarma (wrap) özelliğine sahiptir ve ayrıca alt öğelerin boyut, yön ve hizalama özellikleri üzerinde daha ayrıntılı kontrol imkanı sunar.

FlexLayout, esnek ve duyarlı (responsive) kullanıcı arayüzleri oluşturmak için kullanılan bir düzen yöneticisidir. FlexLayout, CSS Flexbox modeline benzer bir yapıya sahiptir ve esnek, dinamik yerleşimlerde kullanılmak üzere tasarlanmıştır. Bu düzen yöneticisi, çocuk (child) öğelerin düzenlenmesini ve hizalanmasını esnek bir şekilde yönetmek için çeşitli özellikler sunar.

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/layouts/flexlayout?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.



```
1 new FlexLayout()
2 .ItemsSources(BindingContext.Categories)
3 .Assign(out var flex)
4 .Wrap(FlexWrap.Wrap)
5 .FlexBasis(FlexBasis.Auto)
6 .ItemTemplates(new DataTemplate(() =>
7     new Frame()
8     .CornerRadius(15)
9     .BorderColor(LightGray)
10    .BackgroundColor(LightGray)
11    .MinimumHeightRequest(30)
12    .WidthRequest(180)
13    .Padding(8)
14    .Margin(1,0,5)
15    .FlexBasis(FlexBasis.Auto)
16    .Content(
17        new Grid()
18        .ColumnDefinitions(e => e.Star(3).Star(7))
19        .Padding(5)
20        .Children(
21            new Image()
22            .Source(e => e.Path("Icon"))
23            .SizeRequest(30)
24            .Column(0)
25            .CenterVertical(),
26
27            new Label()
28            .Text(e => e.Path("Name"))
29            .TextColor(CornflowerBlue)
30            .FontAttributes(Bold)
31            .FontSize(12)
32            .Column(1)
33            .FontAutoScalingEnabled(true)
34            .CenterVertical()
35        )
36    )
37 )))
```

.NET MAUI Bileşenleri - FlexLayout

FlexLayout Özellikleri:

1. **Direction:** Flex yönünü belirler. Bu özellik, Child öğelerin yatay veya dikey olarak düzenleneceğini belirler. Varsayılan değeri **Row**'dur.
 - a. **Row:** Child öğeler yatay olarak soldan sağa dizilir.
 - b. **Column:** Child öğeler dikey olarak yukarıdan aşağıya dizilir.
 - c. **RowReverse:** Child öğeler yatay olarak sağdan sola dizilir.
 - d. **ColumnReverse:** Child öğeler dikey olarak aşağıdan yukarıya dizilir.
2. **JustifyContent:** Child öğeler arasındaki hizalamanın nasıl olacağını belirtir. Varsayılan kullanım; **Start**
 - a. **Start:** Child öğeler başlangıç noktasında hizalanır.
 - b. **Center:** Child öğeler ortalanır.
 - c. **End:** Child öğeler bitiş noktasında hizalanır.
 - d. **SpaceBetween:** İlk ve son öğeler kenarlara hizalanır, aradaki öğeler eşit aralıklara yerleştirilir.
 - e. **SpaceAround:** Child öğeler arasında eşit boşluk bırakılır ve kenarlarda yarınluk boşluk bırakılır.
 - f. **SpaceEvenly:** Child öğeler arasında ve kenarlarda eşit boşluk bırakılır.

.NET MAUI Bileşenleri - FlexLayout

FlexLayout Özellikleri:

3. **AlignContent:** Kullanılan objelerin nasıl hizalanacağını belirtir. Varsayılan kullanım; **Stretch**
 - a. **Start:** Satırlar başlangıç noktasında hizalanır.
 - b. **Center:** Satırlar ortalanır.
 - c. **End:** Satırlar bitiş noktasında hizalanır.
 - d. **SpaceBetween:** Satırlar arasında eşit boşluk bırakılır.
 - e. **SpaceAround:** Satırlar arasında ve kenarlarda eşit boşluk bırakılır.
 - f. **Stretch:** Satırlar ekseni dolduracak şekilde genişletilir.
4. **AlignItems:** AlignContent yaklaşımından farkı çapraz olarak bu hizalamayı yönetir. Hizalamanın nerede olacağını belirtir. Varsayılan kullanım; **Stretch**
 - a. **Start:** Child öğeler başlangıç noktasında hizalanır.
 - b. **Center:** Child öğeler ortalanır.
 - c. **End:** Child öğeler bitiş noktasında hizalanır.
 - d. **Stretch:** Child öğeler ekseni dolduracak şekilde genişletilir.

.NET MAUI Bileşenleri - FlexLayout

FlexLayout Özellikleri:

5. **Position:** Objelerin dağılımının birbirine göre mi yoksa sabit bir değere göre mi olacağını belirtir.
Varsayılan kullanım: **Relative**
 - a. **Relative:** Child öğe, FlexLayout içinde normal akışta (flow) yer alır. Diğer öğelerle birlikte esnek düzen kurallarına göre hizalanır ve düzenlenir.
 - b. **Absolute:** Child öğe, FlexLayout içindeki normal akıştan çıkarılır ve sabit (absolute) bir pozisyonda yerleştirilir. Bu durumda, ögenin konumu AbsoluteLayout'ta olduğu gibi manuel olarak belirlenir. Diğer öğeler bu ögeyi dikkate almaz.
6. **AlignSelf:** Bir Child ögenin, ana FlexLayout'ta belirlenmiş olan AlignItems ayarından bağımsız olarak, kendi hizalanmasını belirlemenizi sağlar. Bu özellik, ögenin çapraz eksen (cross axis) üzerindeki hizalanmasını özelleştirmek için kullanılır. Varsayılan değer: **Auto**
 - a. **Auto:** Child öğe, ana FlexLayout'un AlignItems özelliğine göre hizalanır.
 - b. **Start:** Child öğe, çapraz eksenin başlangıcında hizalanır.
 - c. **Center:** Child öğe, çapraz eksen üzerinde ortalanır.
 - d. **End:** Child öğe, çapraz eksenin bitiş noktasında hizalanır.
 - e. **Stretch:** Child öğe, çapraz eksenin dolduracak şekilde genişletilir.

.NET MAUI Bileşenleri - FlexLayout

FlexLayout Özellikleri:

7. **Wrap:** Child öğelerin bir satır veya sütuna sığmadığında, otomatik olarak sarılmasını (wrap) sağlar.
Varsayılan değer: **NoWrap**
 - a. **NoWrap:** Child öğeler tek satırda/sütunda kalır, taşanlar kesilir.
 - b. **Wrap:** Child öğeler bir satır/sütuna sığmadığında otomatik olarak sarılır.
 - c. **ReverseWrap:** Child öğeler sarılır, ancak ters yönde yerleştirilir.
8. **Basis:** Child öğelerde kullanılır. Bir Child'ın temel boyutunu belirler. Esnekliği (flexibility) ve genişlemesini belirlemek için kullanılır. Varsayılan değer: **Auto**
9. **Grow:** Child öğelerde kullanılır. Child'ın Layout içerisindeki kaplayacağı alanı belirtir. Varsayılan değer: 0
10. **Shrink:** Child öğelerde kullanılır. Child'ın responsive olarak nasıl/ne kadar küçülmesi gerektiğini belirtir. Varsayılan değer: 1
11. **Order:** Child öğelerde kullanılır. Child öğelerin düzenlemeye sırasını belirler. Varsayılan değer: 0

.NET MAUI Bileşenleri - AbsoluteLayout

AbsoluteLayout, öğeleri açık değerler veya düzenin boyutuna göre göreli değerler kullanarak konumlandırmak ve boyutlandırmak için kullanılır. Konum, alt ögenin sol üst köşesi ile AbsoluteLayout'un sol üst köşesi arasındaki göreli konum olarak belirtilir.

AbsoluteLayout, yalnızca alt öğelere boyut uygulayabileceğiniz veya ögenin boyutu diğer alt öğelerin konumlandırma üzerinde etkisi olmadığındaki kullanılması gereken özel amaçlı bir düzen olarak düşünülmelidir. Bu düzenin standart bir kullanımı, diğer kontrollerle sayfayı kaplayan ve kullanıcının sayfadaki normal kontrollerle etkileşimini engellemek için kullanılabilen bir örtü (overlay) oluşturmaktır.

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/layouts/absolutelayout?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.



```
1 new AbsoluteLayout()
2 .Margin(40)
3 .Children(
4     new BoxView()
5         .Color(Red)
6         .AbsoluteLayoutFlags(AbsoluteLayoutFlags.PositionProportional)
7         .AbsoluteLayoutBounds(0.5, 0, 100, 100)
8         .Rotation(30),
9
10    new BoxView()
11        .Color(Green)
12        .AbsoluteLayoutFlags(AbsoluteLayoutFlags.PositionProportional)
13        .AbsoluteLayoutBounds(0.5, 0, 100, 100)
14        .Rotation(60)
15 )
```

.NET MAUI Bileşenleri - BindableLayout

BindableLayout, Layout sınıfından türetilen herhangi bir düzen (layout) sınıfının, öğelerin bir koleksiyonuna bağlanarak içeriğini oluşturmasını ve her ögenin görünümünü bir DataTemplate ile ayarlama seçeneği sağlayan bir özelliktir.

Bir bağlanabilir düzen (layout), ItemsSource özelliği herhangi bir IEnumerable uygulayan koleksiyona ayarlanarak ve bir Layout türevli sınıfına eklenerken veriyle doldurulur. Bağlanabilir düzendeki her ögenin görünümü, BindableLayout.ItemTemplate ekli özelliği bir DataTemplate'e ayarlanarak tanımlanabilir.

Örnek BindableLayout kullanımı:



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/layouts/bindablelayout?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - ContentPage

ContentPage, genellikle Grid veya StackLayout gibi bir düzen olan tek bir görünüm görüntüler ve en yaygın sayfa türüdür.

FmgLib.MauiMarkup kütüphansında HotReload işlemi gerçekleştirebilmek için FmgLibContentPage yapısı geliştirilmiştir. Bu sınıf mevcut ContentPage içerisinde HotReload özelliği sağlamaktadır. FmgLibContentPage ViewModel desteklidir. MVVM yapısı rahatlıkla kullanılabilmektedir. Tasarım kodları **Build** methodu içerisinde yazılır(HotReload özelliği kullanılabilmesi için şarttır). HotReload istenmiyorsa normal ContentPage kullanılabilir ve yapıçı method içerisinde tasarım kodları.

FmgLib.MauiMarkup 8.7.0 versiyonu ile birlikte artık direkt FmgLibContentPage kullanımına gerek kalmadan HotReload özelliği ile kodlama yapabilirsiniz. Ve sadece ContentPage'lerde değil bütün page türlerinde HotReload özelliğinden yararlanabileceksiniz. Yapılması gereken şey sayfanın **IFmgLibHotReload** interface'nden implemente edilmesi ve yapıçı method içerisinde **this.InitializeHotReload();** yazılmasıdır.



```
1 public partial class ExamplePage : ContentPage, IFmgLibHotReload
2 {
3     public ExamplePage()
4     {
5         this.InitializeHotReload();
6     }
7     public void Build()
8     {
9         this
10            .Content(
11                new Label().Text("FmgLib.MauiMarkup")
12            );
13    }
14 }
```

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/pages/contentpage?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - FlyoutPage

FlyoutPage, iki ilişkili bilgi sayfasını yöneten bir sayfadır: öğeleri sunan bir açılır (flyout) sayfa ve açılır sayfadaki öğelerle ilgili ayrıntıları sunan bir ayrıntı sayfası.

FlyoutPage'in iki düzen davranışları vardır:

- **Açılmış pencere düzeni:** Ayrıntı sayfası, açılır sayfayı kaplar veya kısmen kaplar. Açılır sayfadaki bir öğeyi seçmek, ilgili ayrıntı sayfasına götürür. Telefonlarda çalışan uygulamalar her zaman bu düzen davranışını kullanır.
- **Bölünmüş düzen:** Açılır sayfa solda, ayrıntı sayfası sağda görüntülenir. Tabletlerde veya masaüstü bilgisayarlarda çalışan uygulamalar bu düzen davranışını kullanabilir ve Windows varsayılan olarak bunu kullanır.



ContentPage



FlyoutPage



NavigationPage



TabbedPage

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/pages/flyoutpage?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - NavigationPage

NavigationPage, istediğiniz şekilde sayfalar arasında ileri ve geri gezinmemizi sağlayan hiyerarşik bir gezinme deneyimi sunar.



NavigationPage, gezinmeyi Son Girilen İlk Çıkar (LIFO) bir Page nesne yiğini olarak sağlar.



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/pages/navigationpage?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - TabbedPage

TabPage, bir seferde yalnızca biri tamamen görünür olan Page türünden bir alt öğe koleksiyonunu yönetir. Her alt öğe, sayfanın üst veya alt kısmındaki bir dizi sekmeyle tanımlanır. Tipik olarak her alt öğe bir ContentPage olur ve sekmesi seçildiğinde sayfa içeriği görüntülenir.



<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/pages/tabbedpage?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Shell

.NET MAUI'de **Shell**, uygulamalar için modern bir gezinme yapısı sağlar ve özellikle karmaşık mobil ve masaüstü uygulamalarının gezinme ihtiyaçlarını basit ve verimli bir şekilde yönetmek için tasarlanmıştır. Shell, geleneksel sayfa tabanlı gezinme modelinin üzerine inşa edilmiştir ve daha temiz, düzenli bir kod yapısı sunar.

Shell'in Başlıca Özellikleri

- **Hiyerarşik Gezinme:** Shell, sayfalar arasında hiyerarşik bir yapı oluşturmanıza olanak tanır. Örneğin, bir ana sayfadan bir alt sayfaya geçmek gibi.
- **Tabbar ve Flyout Menüler:** Shell, tab bar (sekme çubuğu) ve flyout menü (yan menü) gibi yaygın gezinme yapılarını kolayca eklemenizi sağlar.
- **URI tabanlı Gezinme:** Shell, URI (Uniform Resource Identifier) tabanlı bir gezinme modelini destekler. Bu, belirli sayfalara URI kullanarak doğrudan erişim sağlamaayı kolaylaştırır.
- **Veri Şablonları:** Shell, veri şablonlarını destekler, böylece sayfa düzenlerini ve stillerini özelleştirebilirsiniz.

<https://learn.microsoft.com/tr-tr/dotnet/maui/fundamentals/shell/?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Shell

Shell'in Avantajları

- **Kolaylaştırılmış Gezinme:** Shell, karmaşık gezinme yapılarının daha basit bir şekilde oluşturulmasını sağlar. [NavigationPage](#) ile yapılan işlemleri, tek bir yapı altında toplayarak daha düzenli bir çözüm sunar.
- **Daha Az Kod:** Shell, sayfalar arası geçişleri yönetmek için daha az kod yazmayı gerektirir. Örneğin, tab bar veya flyout menüler gibi yaygın gezinme yapıları için ekstra kod yazmak gerekmez.
- **Esnek Yapı:** Shell, uygulamanın genel gezinme deneyimini özelleştirmek için geniş bir yapı sunar. URI tabanlı gezinme, sayfa geçişlerinde esneklik sağlar ve sayfalar arası veri aktarımı kolaylaştırır.
- **Yerleşik Animasyonlar:** Shell, sayfalar arasında yerleşik animasyonlar sunar, bu da kullanıcı deneyimini iyileştirir.

Örnek Shell kullanımı:



```
1 public partial class AppShell : Shell
2 {
3     public AppShell()
4     {
5         this
6             .Items(
7                 new TabBar()
8                     .Items(
9                         new Tab().Title("Ders1").Items(new ShellContent().ContentTemplate(() => new Ders1())),
10                        new Tab().Title("Ders2").Items(new ShellContent().ContentTemplate(() => new Ders2())))
11                 )
12             );
13     }
14 }
```

.NET MAUI Bileşenleri - DataTemplate

DataTemplate, bir veri kaynağının nasıl görüneceğini tanımlayan bir şablondur. Özellikle MVVM (Model-View-ViewModel) modelini kullanarak UI tasarımda veri bağlama işlemlerini basitleştirmek için kullanılır. MAUI'de, bir kontrolün öğelerini (örneğin, bir ListView veya CollectionView'daki öğeler) nasıl görselleştireceğinizi belirtmek için kullanılır. Veri şablonları, belirli bir veri türü için özel bir görselleştirme sağlar ve veri bağlama işlemi sayesinde veri ve görünüm arasında bir köprü kurar. Örneğin, bir liste kontrolünde her bir ögenin nasıl görüneceğini tanımlamak için bir DataTemplate kullanabilirsiniz.

Bir **DataTemplateSelector**, veri bağlanan bir özelliğin değerine dayalı olarak çalışma zamanında bir DataTemplate (veri şablonu) seçmek için kullanılabilir. Bu, aynı türdeki nesnelere birden fazla veri şablonu uygulanmasını ve çalışma zamanında görünümlerinin seçilmesini sağlar. Bir veri şablonu seçici, her bir nesnenin görünümünün çalışma zamanında seçilebildiği senaryoları etkinleştirir; örneğin, bir CollectionView veya CarouselView'un bir nesne koleksiyonuna bağlılığı ve veri şablonu seçicisinin belirli bir DataTemplate döndürerek her bir nesnenin görünümünü seçebildiği senaryolar gibi.

```
1  public class PersonDataTemplateSelector : DataTemplateSelector
2  {
3      public DataTemplate ValidTemplate { get; set; }
4      public DataTemplate InvalidTemplate { get; set; }
5
6      protected override DataTemplate OnSelectTemplate(object item, BindableObject container)
7      {
8          return ((Person)item).DateOfBirth.Year >= 1980 ? ValidTemplate : InvalidTemplate;
9      }
10 }
```

.NET MAUI Bileşenleri - ControlTemplate

ControlTemplate, ContentView'den türetilmiş özel kontrollerin ve ContentPage'den türetilmiş sayfaların görsel yapısını tanımlamanıza olanak tanır. Kontrol şablonları, bir özel kontrol veya sayfa için kullanıcı arayüzü (UI), kontrol veya sayfayı uygulayan mantıktan ayırrı. Ayrıca, şablonlanmış özel kontrole veya şablonlanmış sayfaya önceden tanımlanmış bir konumda ek içerik de eklenebilir.

Örneğin, bir kontrol şablonu, bir özel kontrol tarafından sağlanan kullanıcı arayüzü yeniden tanımlamak için oluşturulabilir. Daha sonra, bu kontrol şablonu, gereken özel kontrol örneği tarafından kullanılabilir. Alternatif olarak, bir uygulamadaki birden fazla sayfa tarafından kullanılacak ortak bir kullanıcı arayüzü tanımlayan bir kontrol şablonu oluşturulabilir. Bu kontrol şablonu daha sonra birden fazla sayfa tarafından kullanılabilir ve her sayfa yine de kendine özgü içeriğini gösterebilir.

<https://learn.microsoft.com/tr-tr/dotnet/maui/fundamentals/controltemplate?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Shadow

Shadow sınıfı, bir düzenin (layout) veya görünümün (view) etrafına gölge çizer. VisualElement sınıfı, herhangi bir düzen veya görünümüne gölge eklenmesine olanak tanıyan, Shadow türünde bir Shadow bindable (bağlanabilir) özelliğine sahiptir.

Shadow sınıfı aşağıdaki özelliklerini tanımlar:

- **Radius** (float türünde): Gölgeyi oluşturmak için kullanılan bulanıklığın yarıçapını tanımlar. Bu özelliğin varsayılan değeri 10'dur.
- **Opacity** (float türünde): Gölgenin opaklığını (saydamlığını) belirtir. Bu özelliğin varsayılan değeri 1'dir.
- **Brush** (Brush türünde): Gölgeyi renklendirmek için kullanılan fırçayı temsil eder.
- **Offset** (Point türünde): Gölge için ofseti belirtir, bu da gölgeyi oluşturan ışık kaynağının konumunu temsil eder.



```
1 new Label()
2 .Text("FmgLib.MauiMarkup")
3 .CharacterSpacing(2)
4 .FontSize(30)
5 .FontAttributes(Italic)
6 .TextColor(Green)
7 .TextCenter()
8 .Shadow(
9     new Shadow()
10    .Radius(15)
11    .Opacity(0.7f)
12    .Brush(Gray)
13    .Offset(new Point(5,5))
14 )
```

.NET MAUI Bileşenleri - Resource Dictionary

ResourceDictionary, bir .NET MAUI uygulaması tarafından kullanılan kaynaklar için bir depodur. ResourceDictionary'de depolanan tipik kaynaklar arasında stiller, kontrol şablonları, veri şablonları, dönüştürücüler ve renkler bulunur.

```
1  public class AppStyles
2  {
3      public static ResourceDictionary Default => new ResourceDictionary {
4
5          // "ActivityIndicator"
6
7          new Style<ActivityIndicator>(e => e
8              .Color(e => e.OnLight(AppColors.Primary).OnDark(White)),
9
10         // "IndicatorView"
11
12         new Style<IndicatorView>(e => e
13             .IndicatorColor(e => e.OnLight(AppColors.Gray200).OnDark(AppColors.Gray500))
14             .SelectedIndicatorColor(e => e.OnLight(AppColors.Gray950).OnDark(AppColors.Gray100)),
15
16         // "Border"
17
18         new Style<Border>(e => e
19             .Stroke(e => e.OnLight(AppColors.Gray200).OnDark(AppColors.Gray500))
20             .StrokeShape(new Rectangle())
21             .StrokeThickness(1))
22     };
23 }
```

.NET MAUI Bileşenleri - Styles

.NET MAUI'de **Style** sınıfı, birden fazla görsel elemana uygulanabilecek bir stil oluşturmak için kullanılır. Stiller, UI (Kullanıcı Arayüzü) öğelerinin belirli bir şekilde görünmesini ve davranışmasını sağlamak amacıyla, özellik değerlerini merkezi bir yerde tanımlar. Bu yaklaşım, kod tekrarlamasını en aza indirir ve uygulamanın stil ve tema yönetimini daha basit ve esnek hale getirir.

.NET MAUI CSS desteklidir. <https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/styles/css?view=net-maui-8.0>



```
1  public class AppStyles
2  {
3      public static ResourceDictionary Default => new ResourceDictionary {
4
5          // "ActivityIndicator"
6
7          new Style<ActivityIndicator>(e => e
8              .Color(e => e.OnLight(AppColors.Primary).OnDark(White))),
9
10         // "IndicatorView"
11
12         new Style<IndicatorView>(e => e
13             .IndicatorColor(e => e.OnLight(AppColors.Gray200).OnDark(AppColors.Gray500))
14             .SelectedIndicatorColor(e => e.OnLight(AppColors.Gray950).OnDark(AppColors.Gray100))),
15
16         // "Border"
17
18         new Style<Border>(e => e
19             .Stroke(e => e.OnLight(AppColors.Gray200).OnDark(AppColors.Gray500))
20             .StrokeShape(new Rectangle())
21             .StrokeThickness(1))
22     };
23 }
```

.NET MAUI Bileşenleri - Fonts

Varsayılan olarak, .NET Multi-platform App UI (.NET MAUI) uygulamaları her platformda Open Sans fontunu kullanır. Ancak, bu varsayılan ayar değiştirilebilir ve bir uygulamada kullanılmak üzere ek yazı tipleri (fontlar) kaydedilebilir.

Metin görüntüleyen tüm kontroller, yazı tipi görünümünü değiştirmek için ayarlanabilecek özellikler tanımlar:

- **FontFamily:** `string` türünde bir özellik olup, kullanılan yazı tipini belirtir.
- **FontAttributes:** `FontAttributes` türünde bir özellik olup, üç üyesi vardır: `None`, `Bold` (Kalın) ve `Italic` (İtalik). Bu özelliğin varsayılan değeri `None`'dur.
- **FontSize:** `double` türünde bir özellik olup, yazı tipinin boyutunu belirtir.
- **FontAutoScalingEnabled:** `bool` türünde bir özellik olup, bir uygulamanın kullanıcı arayüzünün, işletim sisteminde ayarlanan metin ölçeklendirme tercihlerine göre değişip değimeyeceğini tanımlar. Bu özelliğin varsayılan değeri `true`'dur.

Bu özellikler `BindableProperty` nesneleriyle desteklenir, bu da onların veri bağlamalarının hedefi olabileceği ve stillendirilebileceği anlamına gelir.

Metin görüntüleyen tüm kontroller, otomatik olarak yazı tipi ölçeklendirmesini kullanır; bu, bir uygulamanın kullanıcı arayüzünün, işletim sisteminde ayarlanan metin ölçeklendirme tercihlerine göre değiştiği anlamına gelir.

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/fonts?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Visual States

Visual State Manager (Görsel Durum Yöneticisi), kullanıcı arayüzünde koddan görsel değişiklikler yapmanın yapılandırılmış bir yolunu sunar. Çoğu durumda, bir uygulamanın kullanıcı arayüzü XAML'de tanımlanır ve bu XAML, Görsel Durum Yöneticisinin kullanıcı arayüzünün görselliğini nasıl etkilediğini tanımlayan işaretlemeleri içerebilir.

Visual State Manager, görsel durumlar kavramını tanır. Örneğin, bir .NET MAUI görünümü (view) olan Button (Buton), altında yatan duruma bağlı olarak — devre dışı, basılmış ya da odaklanmış olup olmadığı gibi — birkaç farklı görsel görünüme sahip olabilir. Bunlar, butonun durumlarıdır. Görsel durumlar, görsel durum gruplarında toplanır. Bir görsel durum grubundaki tüm görsel durumlar birbirini dışlar. Hem görsel durumlar hem de görsel durum grupları basit metin dizeleriyle tanımlanır.

.NET MAUI Visual State Manager, şu görsel durumlarla birlikte bir CommonStates (Ortak Durumlar) adlı bir görsel durum grubu tanımlar:

- Normal
- Disabled (Devre Dışı)
- Focused (Odaklanmış)
- PointerOver (İmleç Üzerinde)

Normal, Disabled, Focused ve PointerOver görsel durumları, VisualElement sınıfından türeyen tüm sınıflarda desteklenir. VisualElement, View ve Page sınıflarının temel sınıfıdır. Buna ek olarak, kendi görsel durum gruplarınızı ve görsel durumlarınızı da tanımlayabilirsiniz.

Visual State Manager kullanarak görünümü tanımlamanın avantajı, görsel öğelere doğrudan kod-behind'den (arkaplan kodundan) erişmek yerine, görsel öğelerin farklı durumlara nasıl tepki vereceğini tamamen XAML'de kontrol edebilmenizdir. Bu, tüm kullanıcı arayüzü tasarımını tek bir yerde tutar.

.NET MAUI Bileşenleri - Visual States

VisualStateGroup: `VisualStateGroup`, bir grup görsel durumu tanımlar. Bu örnekte, `CommonStates` adlı bir grup kullanıyoruz.

VisualState: Her bir `VisualState`, bir kontrolün belirli bir durumu için görsel özellikleri tanımlar. Örneğin, `Normal` durumu için `BackgroundColor` ve `TextColor` özelliklerini ayarlıyoruz.

Setter: `Setter` öğeleri, belirli bir görsel durum için ayarlanan özellik değerlerini belirtir.

Avantajları:

- **Merkezi Yönetim:** Tüm UI değişiklikleri tek bir yerde tanımlanır (XAML dosyası), bu da kodun bakımını kolaylaştırır.
- **Temiz Kod:** Görsel durumların kod arka planında değil, XAML'de tanımlanması, daha temiz ve anlaşılır bir kod yapısı sağlar.
- **Kolay Değişiklik:** Görsel durumlar ve bu durumların davranışları kolayca güncellenebilir ve değiştirilebilir.

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/visual-states?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Brushes

.NET Multi-platform App UI (.NET MAUI) brush (fırça), bir alanı boyamak için, örneğin bir kontrolün arka planını boyamak için farklı yaklaşımalar kullanmanıza olanak tanır.

Brush sınıfı, bir alanı çıktısıyla boyayan soyut bir sınıfıdır. Brush sınıfından türeyen sınıflar, bir alanı boyamanın farklı yollarını tanımlar. Aşağıdaki liste, .NET MAUI'de mevcut olan farklı fırça türlerini açıklamaktadır:

- **SolidColorBrush**: Bir alanı düz (tek renkli) bir renkle boyar. Daha fazla bilgi için Solid color brushes (Düz renk fırçalar) kısmına bakın.
- **LinearGradientBrush**: Bir alanı doğrusal bir gradyanla boyar. Daha fazla bilgi için Linear gradient brushes (Doğrusal gradyan fırçalar) kısmına bakın.
- **RadialGradientBrush**: Bir alanı radyal bir gradyanla boyar. Daha fazla bilgi için Radial gradient brushes (Radyal gradyan fırçalar) kısmına bakın.

Bu fırça türlerinin örnekleri, bir şeklin (Shape) Stroke ve Fill özelliklerine, bir sınırın (Border) Stroke özelliğine, bir gölgenin (Shadow) Brush özelliğine ve bir görsel elemanın (VisualElement) Background özelliğine atanabilir.

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/brushes/?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Shapes

.NET MAUI, çeşitli şekiller oluşturmanıza olanak tanıyan zengin bir şekil koleksiyonuna sahiptir. Şekiller, arayüzün herhangi bir yerine eklenebilir ve genellikle arka planlar, dekoratif öğeler veya etkileşimli bileşenler olarak kullanılır.

1. Shape Elemanları:

- **Line (Çizgi):** İki nokta arasında düz bir çizgi çizer.
- **Rectangle (Dikdörtgen):** Belirtilen genişlik ve yükseklikte bir dikdörtgen çizer.
- **Ellipse (Elips):** Çember veya elips çizer. Yükseklik ve genişlik eşit olduğunda bir daire çizer.
- **Polygon (Çokgen):** Belirtilen noktaları birleştirerek kapalı bir şekil çizer.
- **Polyline (Çoklu Çizgi):** Belirtilen noktaları birleştirerek açık bir çizgi çizer.
- **Path (Yol):** Daha karmaşık şekiller ve çizimler için kullanılır. Bezier eğrileri gibi gelişmiş çizim komutlarını içerir.

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/controls/shapes/?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Shapes

2. Shape Özellikleri:

- **Aspect:** `Stretch` türünde bir özellik olup, şeklin tahsis edilen alanını nasıl doldurduğunu açıklar. Bu özelliğin varsayılan değeri `Stretch.None`'dur.
- **Fill:** `Brush` türünde bir özellik olup, şeklin içini boyamak için kullanılan fırçayı belirtir.
- **Stroke:** `Brush` türünde bir özellik olup, şeklin dış çizgisini boyamak için kullanılan fırçayı belirtir.
- **StrokeDashArray:** `DoubleCollection` türünde olup, bir şekli çevreleyen çizgilerin çizgi ve boşluk desenini belirten double değerlerin koleksiyonunu temsil eder.
- **StrokeDashOffset:** `double` türünde olup, çizgi deseni içinde bir çizginin başladığı mesafeyi belirtir. Bu özelliğin varsayılan değeri `0.0`'dır.
- **StrokeDashPattern:** `float[]` türünde olup, bir şeklin çizgisini çizerken kullanılan çizgi ve boşluk desenini belirtir.
- **StrokeLineCap:** `PenLineCap` türünde olup, bir çizginin veya segmentin başlangıç ve bitiş şekillerini tanımlar. Bu özelliğin varsayılan değeri `PenLineCap.Flat`'tir.
- **StrokeLineJoin:** `PenLineJoin` türünde olup, bir şeklin köşelerinde kullanılan birleştirme türünü belirtir. Bu özelliğin varsayılan değeri `PenLineJoin.Miter`'dır.
- **StrokeMiterLimit:** `double` türünde olup, bir şeklin dış çizgisinin kalınlığının yarısına oranla miter uzunluğunun sınırını belirtir. Bu özelliğin varsayılan değeri `10.0`'dır.
- **StrokeThickness:** `double` türünde olup, şeklin dış çizgi kalınlığını belirtir. Bu özelliğin varsayılan değeri `1.0`'dır.

.NET MAUI Bileşenleri - Animations

Animations, görsel elemanların farklı özelliklerini hedef alır ve tipik bir temel animasyon, bir özelliği zaman içinde bir değerden diğerine kademeli olarak değiştirir.

Temel animasyonlar, **VisualElement** nesneleri üzerinde çalışan **ViewExtensions** sınıfı tarafından sağlanan genişletme metodlarıyla oluşturulabilir:

- **CancelAnimations**: Tüm animasyonları iptal eder.
- **FadeTo**: Bir **VisualElement** nesnesinin **Opacity** (opaklık) özelliğini animasyonlu bir şekilde değiştirir.
- **RelScaleTo**: Bir **VisualElement** nesnesinin **Scale** (ölçek) özelliğinde artımlı bir artış veya azalma uygular.
- **RotateTo**: Bir **VisualElement** nesnesinin **Rotation** (dönme) özelliğini animasyonlu bir şekilde değiştirir.
- **RelRotateTo**: Bir **VisualElement** nesnesinin **Rotation** (dönme) özelliğinde artımlı bir artış veya azalma uygular.
- **RotateXTo**: Bir **VisualElement** nesnesinin **RotationX** özelliğini animasyonlu bir şekilde değiştirir.
- **RotateYTo**: Bir **VisualElement** nesnesinin **RotationY** özelliğini animasyonlu bir şekilde değiştirir.
- **ScaleTo**: Bir **VisualElement** nesnesinin **Scale** (ölçek) özelliğini animasyonlu bir şekilde değiştirir.
- **ScaleXTo**: Bir **VisualElement** nesnesinin **ScaleX** özelliğini animasyonlu bir şekilde değiştirir.
- **ScaleYTo**: Bir **VisualElement** nesnesinin **ScaleY** özelliğini animasyonlu bir şekilde değiştirir.
- **TranslateTo**: Bir **VisualElement** nesnesinin **TranslationX** ve **TranslationY** özelliklerini animasyonlu bir şekilde değiştirir.

Varsayılan olarak, her animasyon 250 milisaniye sürecektir. Ancak, animasyon oluşturulurken her animasyon için bir süre belirtilebilir.

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/animation/basic?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Animations

Animasyonların çalışırken nasıl hızlanıp yavaşlayacağını kontrol eden bir aktarım fonksiyonu belirtmenize olanak tanıyan bir **Easing** sınıfı içerir.

Easing sınıfı, animasyonlar tarafından kullanılabilenek bir dizi yumusatma fonksiyonu tanımlar:

- **BounceIn** yumusatma fonksiyonu, animasyonun başlangıcında bir sıçrama efekti oluşturur.
- **BounceOut** yumusatma fonksiyonu, animasyonun sonunda bir sıçrama efekti oluşturur.
- **CubicIn** yumusatma fonksiyonu, animasyonu yavaşça hızlandırır.
- **CubicInOut** yumusatma fonksiyonu, animasyonu başlangıçta hızlandırır ve sonunda yavaşlatır.
- **CubicOut** yumusatma fonksiyonu, animasyonu hızlı bir şekilde yavaşlatır.
- **Linear** yumusatma fonksiyonu sabit bir hız kullanır ve varsayılan yumusatma fonksiyonudur.
- **SinIn** yumusatma fonksiyonu, animasyonu düzgün bir şekilde hızlandırır.
- **SinInOut** yumusatma fonksiyonu, animasyonu başlangıçta düzgün bir şekilde hızlandırır ve sonunda düzgün bir şekilde yavaşlatır.
- **SinOut** yumusatma fonksiyonu, animasyonu düzgün bir şekilde yavaşlatır.
- **SpringIn** yumusatma fonksiyonu, animasyonu sonuna doğru çok hızlı bir şekilde hızlandırır.
- **SpringOut** yumusatma fonksiyonu, animasyonu sonuna doğru hızlı bir şekilde yavaşlatır.

In ve **Out** son ekleri, yumusatma fonksiyonunun etkisinin animasyonun başlangıcında mı, sonunda mı yoksa her ikisinde de mi belirgin olduğunu gösterir.

<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/animation/easing?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Popups

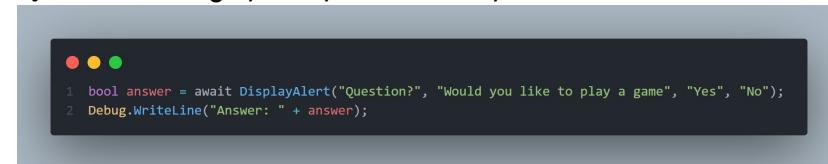
Bir uyarı görüntülemek, kullanıcının bir seçim yapmasını istemek veya bir bilgi girişi istemek, yaygın bir kullanıcı arayüzü (UI) görevidir. .NET MAUI, kullanıcıyla bir açılır pencere aracılığıyla etkileşim kurmak için [Page](#) sınıfında üç yöntem sunar: [DisplayAlert](#), [DisplayActionSheet](#) ve [DisplayPromptAsync](#). Açılmış pencereler, her platformda yerel kontrollerle işlenir.

Bir Uyarı Görüntüleme

.NET MAUI tarafından desteklenen tüm platformlar, kullanıcıyı uyarmak veya onlara basit sorular sormak için bir açılır pencere içerir. Uyarıları görüntülemek için herhangi bir [Page](#) üzerinde [DisplayAlert](#) yöntemini kullanabilirsiniz. Aşağıdaki örnek, kullanıcıya basit bir mesaj gösterir:



[DisplayAlert](#) yöntemi, iki düğme gösterip bir bool döndürerek kullanıcının yanıtını yakalamak için de kullanılabilir. Bir uyarıdan yanıt almak için her iki düğme için de metin sağlayın ve yöntemi bekleyin:



.NET MAUI Bileşenleri - Popups

Bir eylem sayfası, kullanıcının bir görevi nasıl sürdüreceğini dair bir dizi alternatif sunar. Bir eylem sayfasını görüntülemek için herhangi bir **Page** üzerinde **DisplayActionSheet** yöntemini kullanın ve mesajı ve düğme etiketlerini dize olarak geçirin:

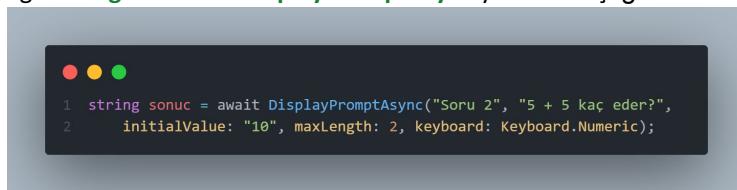


Eylem sayfaları ayrıca yıkıcı davranışı temsil eden bir yıkım düğmesini destekler. Yıkım düğmesi, **DisplayActionSheet** metoduna üçüncü dize argümanı olarak belirtilebilir veya null bırakılabilir. Aşağıdaki örnek bir yıkım düğmesini belirtir:



.NET MAUI Bileşenleri - Popups

Bir istem görüntülemek için, herhangi bir **Page** üzerinde **DisplayPromptAsync** yöntemini çağırın ve başlık ve mesajı dize argümanları olarak geçirin:



DisplayPromptAsync yönteminin tam argüman listesi şunlardır:

- **title**, türü **string**, istemde görüntülenecek başlıktır.
- **message**, türü **string**, istemde görüntülenecek mesajdır.
- **accept**, türü **string**, kabul düğmesinin metnidir. Bu, varsayılan değeri **OK** olan istege bağlı bir argümandır.
- **cancel**, türü **string**, iptal düğmesinin metnidir. Bu, varsayılan değeri **Cancel** olan istege bağlı bir argümandır.
- **placeholder**, türü **string**, istemde görüntülenecek yer tutucu metindir. Bu, varsayılan değeri **null** olan istege bağlı bir argümandır.
- **maxLength**, türü **int**, kullanıcı yanıtının maksimum uzunluğudur. Bu, varsayılan değeri **-1** olan istege bağlı bir argümandır.
- **keyboard**, türü **Keyboard**, kullanıcı yanıtı için kullanılacak klavye türündür. Bu, varsayılan değeri **Keyboard.Default** olan istege bağlı bir argümandır.
- **initialValue**, türü **string**, gösterilecek ve düzenlenebilir olan önceden tanımlanmış bir yanittır. Bu, varsayılan değeri boş bir dize olan istege bağlı bir argümandır.

.NET MAUI Bileşenleri - ToolBar

ToolbarItem sınıfı, .NET MAUI uygulamalarında bir **Page** nesnesinin **ToolbarItems** koleksiyonuna eklenebilen özel bir tür düğmedir. **Shell** sınıfı **Page** sınıfından türediği için, **ToolbarItem** nesneleri de bir **Shell** nesnesinin **ToolbarItems** koleksiyonuna eklenebilir. Her bir **ToolbarItem** nesnesi, uygulamanın navigasyon çubuğunda bir düğme olarak görünür. Bir **ToolbarItem** nesnesi bir simgeye sahip olabilir ve birincil veya ikincil bir öğe olarak görünebilir. **ToolbarItem** sınıfı, **MenuItem** sınıfından türetilmiştir.

ToolbarItem sınıfı şu özellikleri tanımlar:

- **Order:** Türü **ToolbarItemOrder** olan bu özellik, **ToolbarItem** nesnesinin birincil veya ikincil menüde görüntülenip görüntülenmeyeceğini belirler.
- **Priority:** Türü **int** olan bu özellik, **ToolbarItems** koleksiyonundaki öğelerin görüntülenme sırasını belirler.

ToolbarItem sınıfı, **MenuItem** sınıfından şu tipik olarak kullanılan özellikleri devralır:

- **Command:** Türü **ICommand** olan bu özellik, kullanıcı eylemlerini (örneğin, dokunma veya tıklama) bir.viewmodel'de tanımlanan komutlara bağlamaya olanak tanır.
- **CommandParameter:** Türü **object** olan bu özellik, **Command'a** iletilmesi gereken parametreyi belirtir.
- **IconImageSource:** Türü **ImageSource** olan bu özellik, bir **ToolbarItem** nesnesinde görüntülenecek simgeyi belirler.
- **Text:** Türü **string** olan bu özellik, bir **ToolbarItem** nesnesinde görüntülenecek metni belirler.

Bu özellikler, **ToolbarItem**'in kullanıcı arayüzünde nasıl göründüğünü ve nasıl etkileşimde bulunulduğunu kontrol eder, bu da geliştiricilere esnek bir tasarım ve kullanıcı deneyimi sunar.

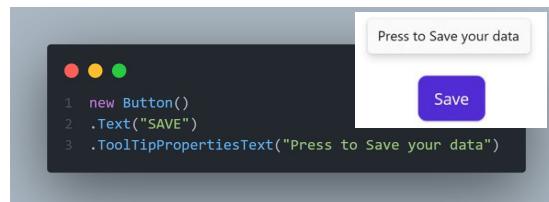
<https://learn.microsoft.com/tr-tr/dotnet/maui/user-interface/toolbaritem?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - ToolTip

.NET MAUI uygulamalarında kullanılan **tooltip** (ipucu), kullanıcının fare imlecini bir görünümün (view) üzerine getirdiğinde o görünümün amacını kısaca açıklayan küçük dikdörtgen bir açılır penceredir. Tooltips (ipuçları) genellikle kullanıcı imleci ilgili görünümün üzerinde durduğunda otomatik olarak görüntülenir:

- **Android'de**, ipuçları kullanıcının görünüme uzun süre basmasıyla görüntülenir. Uzun basma bırakıldıktan sonra ipuçları birkaç saniye daha görünür kalır.
- **iOS'ta**, bir ipucunu göstermek için uygulamanızın bir Mac üzerinde Apple silicon ile çalışan bir iPhone veya iPad uygulaması olması gereklidir. Bu kriter sağlandığında, imleç birkaç saniye boyunca görünümün üzerinde durduğunda ipuçları görüntülenir ve imleç görünümden uzaklaşana kadar görünür kalır. iOS'taki ipuçları iOS 15.0+ kullanımını gerektirir.
- **Mac Catalyst'te**, ipuçları imleç birkaç saniye boyunca görünümün üzerinde durduğunda görüntülenir ve imleç görünümden uzaklaşana kadar görünür kalır. Mac Catalyst üzerindeki ipuçları, Mac Catalyst 15.0+ kullanımını gerektirir.
- **Windows'ta**, ipuçları imleç görünümün üzerinde durduğunda görüntülenir. İpuçları birkaç saniye boyunca ya da imleç görünüm üzerinde durmayı bırakana kadar görünür kalır.

Tooltips (ipuçları), bir görünümde **ToolTipProperties.Text** attached property'sine bir metin atanarak tanımlanır. Örneğin:



.NET MAUI Bileşenleri - WebView

WebView, uygulamalar içinde uzaktaki web sayfalarını, yerel HTML dosyalarını ve HTML dizelerini görüntülemeye yarar. WebView ile görüntülenen içerik, Cascading Style Sheets (CSS) ve JavaScript desteği içerir. Varsayılan olarak, .NET MAUI projeleri, bir WebView'in uzaktaki bir web sayfasını görüntülemesi için gerekli platform izinlerini içerir.

WebView şu özellikleri tanımlar:

- **Cookies**: CookieContainer türünde olup cerezlerin depolanmasını sağlar.
- **CanGoBack**: bool türünde olup kullanıcının önceki sayfalara gidip gidemeyeceğini belirtir. Sadece okunabilir bir özellikir.
- **CanGoForward**: bool türünde olup kullanıcının sonraki sayfalara gidip gidemeyeceğini belirtir. Bu da sadece okunabilir bir özellikir.
- **Source**: WebViewSource türünde olup WebView'in görüntüleyeceği kaynağı temsil eder.
- **UserAgent**: string türünde olup kullanıcı aracısını temsil eder. Varsayılan değer, alt platform tarayıcısının kullanıcı aracısıdır, belirlenemiyorsa null değerini alır.

Bu özellikler, BindableProperty ile desteklenir, bu da onların veri bağlamaları için hedef olabilecekleri ve stil uygulanabilecekleri anlamına gelir.

Source özelliği, UrlWebViewSource veya HtmlWebViewSource türlerine ayarlanabilir. **UrlWebViewSource**, bir URL ile belirtilen web sayfasını yüklemek için kullanılırken, **HtmlWebViewSource** ise yerel bir HTML dosyasını veya HTML dizelerini yüklemek için kullanılır.

WebView ayrıca şu olayları tanımlar:

- **Navigating**: Sayfa gezintisi başladığında tetiklenir.
- **Navigated**: Sayfa gezintisi tamamlandığında tetiklenir.

Navigating olayıyla birlikte gelen **WebNavigatingEventArgs** nesnesinde, gezintiyi iptal etmek için kullanılabilen bir **Cancel** özelliği bulunur. **Navigated** olayıyla birlikte gelen **WebNavigatedEventArgs** nesnesi ise gezintinin sonucunu belirten bir **Result** özelliğine sahiptir.

.NET MAUI Bileşenleri - Menus

.NET MAUI'de sağ tıklama menüsü olarak bilinen context menu, tıklanan kontolle ilgili özel komutlar sunar. Bu menü, Mac Catalyst ve Windows üzerinde çalıştırılan ve **Element** sınıfından türeyen tüm kontrol, sayfa, düzen ve görenümlere eklenebilir.

Context menu, **MenuFlyout** ile tanımlanır ve şu alt öğelerden oluşabilir:

- **MenuFlyoutItem**: Tıklanabilir bir menü öğesini temsil eder.
- **MenuFlyoutSubItem**: Tıklanabilir bir alt menü öğesini temsil eder.
- **MenuFlyoutSeparator**: Menünün öğeleri arasında yatay bir ayraç çizer.

MenuFlyoutSubItem, **MenuFlyoutItem**'dan, o da **MenuItem** sınıfından türetilir. **MenuItem**, menü öğesinin görünümünü ve davranışını belirlemek için çeşitli özellikler tanımlar. Bir menü veya alt menü öğesinin görünümü, **Text** ve **IconImageSource** özellikleri ayarlanarak tanımlanabilir. Tıklama olayına yanıt ise **Clicked**, **Command** ve **CommandParameter** özellikleri kullanılarak yapılandırılabilir.

.NET MAUI Bileşenleri - Semantic Props

.NET MAUI, platformun erişilebilirlik özelliklerine erişmek için iki yaklaşımı destekler: **Semantic** özellikler ve **Automation** özellikleri. Semantic özellikler, uygulamalarda erişilebilirlik değerlerini belirtmek için önerilen yaklaşımındır.

Semantik özellikler, hangi kontrollerin erişilebilirlik odağı alacağını ve hangi metinlerin kullanıcıya sesli olarak okunacağını belirlemek için kullanılır. Bu özellikler, her platformun kendi erişilebilirlik deneyimini kullanır ve platformlar arası davranış birliği sağlamak yerine, platformların sunduğu erişilebilirlik işlevlerine dayanır.

SemanticProperties sınıfı, şu özellikleri tanımlar:

- **Description:** Ekran okuyucunun öğeyi tanıtmak için kullanacağı kısa açıklama.
- **Hint:** Description'a ek olarak kontrolün amacı gibi ekstra bağlam sağlar.
- **HeadingLevel:** Ögeyi başlık olarak işaretleyerek UI'yi düzenler ve gezinmeyi kolaylaştırır. Başlık seviyeleri, 1'den 9'a kadar olabilir.

Semantic focus: Belirli bir elemana odaklanmak için ekran okuyucusunu zorlayabilirsiniz.

Semantic screen reader: **ISemanticScreenReader** arayüzü ile ekran okuyucusu belirli bir metni seslendirmesi için yönlendirebilirsiniz.

Sınırlamalar: Metinlerin sesli okunabilmesi için varsayılan platform ekran okuyucusunun etkin olması gereklidir.

.NET MAUI Bileşenleri - Automation Props

Automation Properties özellikleri, bir ögenin platformun erişilebilirlik çerçevesine nasıl raporlandığını belirlemek için kullanılır. Bu özellikler her platform için erişilebilirlik değerlerini ayarlayarak ekran okuyucuların ögeyi sesli olarak tanıtmasını sağlar.

AutomationProperties sınıfı şu özellikleri tanımlar:

- **ExcludedWithChildren**: Bir öge ve alt ögelerinin erişilebilirlik ağacından çıkarılıp çıkarılmayacağını belirler.
- **IsInAccessibleTree**: Ögenin ekran okuyucular tarafından algılanıp algılanmadığını gösterir.
- **Name**: Ögeyi tanımlayan kısa bir açıklama.
- **HelpText**: Ögeye ilişkin daha uzun bir açıklama, genellikle tooltip olarak kullanılabilir.
- **LabeledBy**: Başka bir ögenin, mevcut ögenin erişilebilirlik bilgilerini tanımlamasına olanak tanır.

.NET 8'den itibaren bazı özellikler yerine **SemanticProperties** kullanılması önerilmektedir:

- **Name** yerine **Description**,
- **HelpText** yerine **Hint**,
- **LabeledBy** yerine **SemanticProperties.Description** kullanılmalıdır.

.NET MAUI Bileşenleri - Behaviors

.NET MAUI Behaviors ile kullanıcı arayüzü (UI) kontrollerine ek işlevsellik ekleyebilirsiniz. Bu işlevsellik, kontrole eklenmiş gibi görünse de aslında ayrı bir behavior (davranış) sınıfında uygulanır ve alt sınıf oluşturmaya gerek kalmaz. Behaviors, kodunuzu kontrolün API'stle doğrudan etkileşime giren, kod-behind yerine daha modüler ve tekrar kullanılabilir bir yapı sunar.

Behaviors kullanarak şunları yapabilirsiniz:

- Bir girdi kutusuna e-posta doğrulayıcı eklemek.
- Dokunma hareketleriyle bir derecelendirme kontrolü oluşturmak.
- Animasyonları kontrol etmek.

.NET MAUI, üç farklı behavior türünü destekler:

1. **Attached behaviors:** Statik sınıflarda tanımlanan bir veya daha fazla attached property içerir. Bunlar, XAML'de kontrol üzerine eklenebilir.
2. **.NET MAUI behaviors:** Behavior veya Behavior<T> sınıfından türeyen sınıflardır.
3. **Platform behaviors:** Yerel kontrollerle ilgili olaylara yanıt verir ve PlatformBehavior<TView> veya PlatformBehavior<TView, TPlatformView> sınıfından türetilir.

.NET MAUI Bileşenleri - Behaviors

Attached Behaviors

- Attached behaviors, attached property'ler aracılığıyla kontrole ek işlevsellik kazandırır.
- Bir attached property'nin **PropertyChanged** delegesi, property'sinin değeri değiştiğinde tetiklenir ve kontrol referansı üzerinde işlem yapar.

Bir attached behavior oluşturmak için statik bir sınıf ve attached property tanımlamanız yeterlidir.

.NET MAUI Behaviors

- Bir behavior, Behavior veya Behavior<T> sınıfından türetilir.
- **OnAttachedTo** metodu, behavior eklenir eklenmez çağrılr ve burada gerekli kurulumlar yapılır.
- **OnDetachingFrom** metodu, behavior kaldırıldığından temizleme işlemleri için kullanılır.

Bir behavior, hedef kontrolün Behaviors koleksiyonuna eklenerek kullanılabilir.

.NET MAUI Bileşenleri - DataBinding

.NET Multi-platform App UI (MAUI) uygulamaları, birden fazla sayfa ve kullanıcı arayüzü öğelerinden oluşur. Bu öğeler, genellikle bir veri kaynağındaki değerleri temsil eder ve kullanıcıların bu değerleri değiştirmesine olanak tanır. Veri bağlama, iki nesne arasındaki özellikleri eşleştirerek bu değişikliklerin otomatik olarak yansıtılmasını sağlar. Veri kaynağı ve hedef arasında tek yönlü, çift yönlü ya da iki nesne arasında karşılıklı veri akışı sağlanabilir.

MAUI'de veri bağlama, kod tekrarını azaltmak için etkin bir yöntemdir. Örneğin, bir Slider'ın değerini bir Label'da göstermek için veri bağlama kullanılabilir. BindableProperty sınıfı, varsayılan bağlama modunu belirler ve genellikle bu mod tek yönlüdür. Ancak, ihtiyaçlara göre çift yönlü veri akışı da sağlanabilir.

Veri bağlama sırasında tür dönüşümleri gerekebilir. Bu durumlarda, **IValueConverter** arayüzü ile özel dönüştürücüler yazılabilir. Ayrıca, bağlamanın başarısız olması durumunda yedek değerler (FallbackValue) ve null değerler için yedek değerler (TargetNullValue) tanımlanabilir.

MAUI'de bağlamalar, performansı artırmak amacıyla derleme zamanında doğrulanabilir ve çözülür. Derleme zamanında yapılan doğrulama hataları, uygulamanın çalışmadan önce tespit edilmesini sağlar.

<https://learn.microsoft.com/tr-tr/dotnet/maui/fundamentals/data-binding/?view=net-maui-8.0> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Gestures

.NET MAUI'de **Gesture** (jest), kullanıcıların dokunmatik ekranlarla etkileşimini sağlar ve kullanıcıların çeşitli hareketlerini algılayarak bu hareketlere karşılık belirli işlemler gerçekleştirilmesine olanak tanır. Özellikle mobil cihazlar gibi dokunmatik ekranlı cihazlar için önemlidir. .NET MAUI, çeşitli **GestureRecognizer** sınıfları ile bu gestures(jestleri) destekler.

.NET MAUI Gesture Çeşitleri:

1. TapGestureRecognizer

- **Kullanım:** Kullanıcının bir ekrana dokunmasını algılar.
- **Özellikler:**
 - Tek dokunuş veya çift dokunuş gibi farklı sayıda dokunmayı algılayabilir.
 - **NumberOfTapsRequired** özelliği ile kaç kez dokunulması gerektiği belirtilebilir.
- **Örnek:** Bir butona dokunarak işlemi başlatmak.

2. PinchGestureRecognizer

- **Kullanım:** Kullanıcının iki parmakla ekrana dokunarak yaklaştırılma veya uzaklaştırma hareketini (zoom in/out) algılar.
- **Özellikler:**
 - **Scale:** Yakınlaştırma miktarını belirtir.
 - **ScaleOrigin:** Yakınlaştırmmanın hangi noktadan yapıldığını belirtir.
- **Örnek:** Bir resim üzerinde yakınlaştırma yapmak.

.NET MAUI Bileşenleri - Gestures

3. SwipeGestureRecognizer

- **Kullanım:** Kullanıcının ekran üzerinde parmağını bir yönde kaydırmasını algılar.
- **Özellikler:**
 - **Direction:** Kaydırma yönünü belirtir (Yukarı, Aşağı, Sağ, Sol).
- **Örnek:** Bir sayfa değiştirme hareketini algılamak.

4. PanGestureRecognizer

- **Kullanım:** Kullanıcının parmağını ekranda sürüklemesini (pan) algılar.
- **Özellikler:**
 - **TotalX** ve **TotalY:** Parmağın ekrandaki hareketini temsil eder.
- **Örnek:** Ekran üzerinde bir öğeyi sürükleyip bırakmak.

5. DragGestureRecognizer

- **Kullanım:** Bir öğeyi sürükleme ve bırakma işlemlerini algılar.
- **Özellikler:**
 - **DragStarting:** Sürüklemenin başlangıç aşamasını belirtir.
 - **Drop:** Sürüklenen ögenin bırakıldığı yeri belirtir.
- **Örnek:** Öğeleri bir yerden bir yere sürükleme işlemi.

.NET MAUI Bileşenleri - Triggers

.NET MAUI'de **Trigger** (tetikleyici), kullanıcı arayüzündeki değişikliklere yanıt vererek, belirli koşullar sağlandığında UI özelliklerinin otomatik olarak güncellenmesine olanak tanır. Triggers, genellikle belirli olayların veya koşulların karşılanması durumunda görsel öğelerdeki özelliklerin dinamik olarak değişimini sağlar.

.NET MAUI Trigger Çeşitleri:

1. Property Trigger

- **Kullanım:** Belirli bir bindable property'nin (bağlanabilir özellik) değerinin değişmesi durumunda çalışır. Örneğin, bir **Button**'ın **IsEnabled** değeri **false** olduğunda renginin değişimini sağlayabilir.
- **Özellikler:**
 - **Property:** İzlenecek özellik.
 - **Value:** Özelliğin belirlenen değeri.
 - **Setters:** Özelliğin tetiklendiğinde değiştireceği özellikler.

2. Data Trigger

- **Kullanım:** Veriye bağlı olan özellikler için kullanılır. Belirli bir bağlama (binding) koşulunun karşılanması durumunda çalışır.
- **Özellikler:**
 - **Binding:** İzlenecek veri kaynağı.
 - **Value:** Koşul sağlandığında tetiklenecek değer.
 - **Setters:** Koşul karşılandığında uygulanacak değişiklikler.

.NET MAUI Bileşenleri - Triggers

3. Event Trigger

- **Kullanım:** Belirli bir olay (event) gerçekleştiğinde tetiklenir. Örneğin, bir öğeye tıklandığında veya odaklandığında (focused) çalışabilir.
- **Özellikler:**
 - **Event:** İzlenecek olay.
 - **Actions:** Olay tetiklendiğinde yapılacak eylemler (genellikle `TriggerAction` kullanılır).

4. Multi Trigger

- **Kullanım:** Birden fazla koşulun aynı anda sağlandığı durumlarda çalışır. İki veya daha fazla özellik değeri izlenebilir ve hepsi belirli bir değere sahip olduğunda tetiklenir.
- **Özellikler:**
 - **Conditions:** Birden fazla koşul tanımlanabilir.
 - **Setters:** Tüm koşullar karşılandığında uygulanacak değişiklikler.

Trigger Kullanımı:

Triggers, genellikle stil, animasyon veya görünümle ilgili hızlı ve dinamik değişiklikler yapmak için kullanılır

.NET MAUI Bileşenleri - Localization

.NET MAUI projesinde yerelleştirme yapmak için, **MauiProgram.cs** dosyasına şu eklenmelidir:

builder.UseMauiMarkupLocalization(); veya

builder.UseMauiMarkupLocalization("Localization1.json", "/Languages/Temp1.json");

Ana projede bir JSON dil dosyası bulunmalıdır. Eğer dil dosyalarının yolu belirtilmezse, **Localization.json** adlı bir dosya ana dizinde aranacaktır.

Önemli nokta: JSON dosyalarının **Build Action**'ı **MauiAsset** olarak ayarlanmalıdır.

Doğru JSON formatı şu şekildedir:

```
json
{
  "Hello": {
    "tr-TR": "Merhaba Dünya!",
    "en-US": "Hello World!"
  },
  "Msg": {
    "tr-TR": "Deneme amaçlı yapılmıştır.",
    "en-US": "It was made for testing purposes."
  }
}
```

<https://fmgyazilim.com/tr/fmglibmauimarkup#localization-multi-lang-support-section> bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - Preferences

.NET MAUI'de **IPreferences** arayüzü, uygulama ayarlarını anahtar/değer çiftleri ile saklamaya yarar.

Varsayılan implementasyon **Preferences.Default** özelliği ile sağlanır.

IPreferences ve **Preferences** sınıfları da **Microsoft.Maui.Storage** ad alanında bulunur.

Preferences anahtarları bir String tipi olmalıdır ve değerler şu veri türlerinden biri olabilir:

- Boolean
- Double
- Int32
- Single
- Int64
- String
- DateTime

DateTime değerleri 64-bit binary formatında saklanır. Saklama işlemi için **ToBinary** ve **FromBinary** metodları kullanılır. UTC formatında olmayan **DateTime** değerlerinde bazı düzenlemeler gerekebilir.

<https://learn.microsoft.com/tr-tr/dotnet/maui/platform-integration/storage/preferences?view=net-maui-8.0&tabs=android>
bu linkten detaylara ulaşabilirsiniz.

.NET MAUI Bileşenleri - SecureStorage

.NET MAUI'de **ISecureStorage** arayüzü, basit anahtar/değer çiftlerini güvenli bir şekilde saklamaya yarar.

Bu arayüzün varsayılan implementasyonu **SecureStorage.Default** özelliği ile sağlanır.

ISecureStorage ve **SecureStorage** sınıfları **Microsoft.Maui.Storage** ad alanında yer alır.

Örnek kullanım:

```
await SecureStorage.Default.SetAsync("oauth_token", "secret-oauth-token-value");
string oauthToken = await SecureStorage.Default.GetAsync("oauth_token");
bool success = SecureStorage.Default.Remove("oauth_token");
SecureStorage.Default.RemoveAll();
```

<https://learn.microsoft.com/tr-tr/dotnet/maui/platform-integration/storage/secure-storage?view=net-maui-8.0&tabs=macos> bu linkten detaylara ulaşabilirsiniz.