



DATENBINDUNG

Vincent Uhlmann
IT-Akademie Dr. Heuer GmbH

WAS IST DATENBINDUNG

- Eine NET MAUI-Anwendung besteht aus einer oder mehreren Seiten, von denen jede in der Regel mehrere Benutzeroberflächenobjekte, so genannte Views, enthält
- Eine der Hauptaufgaben der Anwendung besteht darin, diese Views zu synchronisieren und die verschiedenen Werte oder Auswahlen, die sie darstellen, zu kontrollieren
- Um diese Änderungen erfolgreich verarbeiten zu können, muss die Anwendung über Änderungen in diesen Views oder den zugrunde liegenden Daten informiert werden
- Dies kann durch das definieren von Events erreicht werden, die angeben, wenn eine Änderung vorgenommen wurde
- Die Datenbindung automatisiert diesen Prozess und macht EventHandlerer überflüssig
- Datenbindung ist also die Technik der Verknüpfung der Eigenschaften zweier Objekte, so dass sich Änderungen in einer Eigenschaft automatisch in der anderen widerspiegeln

WAS IST DATENBINDUNG

- Eines der beiden an einer Datenbindung beteiligten Objekte ist immer ein von View abgeleitetes Element und gehört zur visuellen Schnittstelle einer Seite
- Das andere Objekt ist eines der folgenden:
- Eine andere View, normalerweise auf derselben Seite
- Ein Objekt in einer Codedatei

WAS IST DATENBINDUNG

- Datenbindung verknüpft ein Paar von Eigenschaften zwischen zwei Objekten
- Diese beiden Objekte sind das Ziel und die Quelle
- Das Ziel ist das Objekt (und die Eigenschaft), auf dem die Datenbindung festgelegt wird
- Die Quelle ist das Objekt (und die Eigenschaft), auf die von der Datenbindung verwiesen wird
- Im einfachsten Fall fließen Daten von der Quelle zum Ziel
- In einigen Fällen können Daten jedoch auch vom Ziel zur Quelle fließen oder sogar in beide Richtungen

BINDUNGSKONTEXT

- Die BindingContext-Eigenschaft stellt den Kontext für die Datenbindung bereit
- Sie kann jedoch auch weggelassen werden, wenn das Quellobjekt direkt im SetBinding-Aufruf oder in der Binding-Markup-Erweiterung angegeben wird
- Die Eigenschaft, auf die eine Bindung festgelegt werden soll, muss als BindableProperty implementiert sein
- In .NET MAUI sind diese BindableProperty-Objekte über den Klassennamen und nicht über die Instanz selbst aufrufbar

BINDUNGSKONTEXT

```
<Label  
  BindingContext="{x:Reference slider}"  
  FontSize="48"  
  HorizontalOptions="Center"  
  Rotation="{Binding Value}"  
  Text="TEXT"  
  VerticalOptions="Center" />
```

```
<Slider  
  x:Name="slider"  
  Maximum="360"  
  VerticalOptions="Center" />
```

BINDUNGSMODUS

- Der Bindungsmodus bestimmt, wie Daten zwischen der Quelle und dem Ziel der Datenbindung fließen
- Es gibt vier Hauptmodi: OneWay, OneWayToSource, TwoWay, und OneTime
- Der Modus OneWay ist der Standardmodus, bei dem Änderungen in der Quelle automatisch im Ziel reflektiert werden, aber nicht umgekehrt
- OneWayToSource ermöglicht die Umkehrung, wobei Änderungen im Ziel die Quelle aktualisieren
- TwoWay ermöglicht eine bidirektionale Datenbindung, bei der Änderungen in der Quelle und im Ziel sich gegenseitig aktualisieren
- OneTime stellt eine einmalige Datenbindung her, bei der die Daten nur einmal von der Quelle zum Ziel übertragen werden

BINDUNGSMODUS

```
<Label
  BindingContext="{x:Reference slider}"
  FontSize="48"
  HorizontalOptions="Center"
  Rotation="{Binding Value, Mode=TwoWay}"
  Text="TEXT"
  VerticalOptions="Center" />
```

```
<Slider
  x:Name="slider"
  Maximum="360"
  VerticalOptions="Center" />
```


BINDUNGSPFAD

- Der Bindungspfad definiert den Weg, der durch die Eigenschaften eines Objekts oder durch die Elemente einer Sammlung gefolgt wird, um eine bestimmte Eigenschaft zu erreichen
- Der Pfad kann auf eine einzelne Eigenschaft, eine Untereigenschaft (Eigenschaft einer Eigenschaft) oder auf ein Mitglied einer Sammlung verweisen
- Beispiel: Ein Slider-Element, das einen Wert zwischen 0 und 1 repräsentiert, kann mit einem Label verbunden werden

```
<StackLayout Padding="10,0">  
    <Slider x:Name="slider" />  
    <Label Text="{Binding Source={x:Reference slider}, Path=Value}" />  
</StackLayout>
```

ZEICHENFOLGENFORMATIERUNG

- Ermöglicht die Darstellung von Daten (strings, DateTime ...) in der Benutzeroberfläche anzupassen
- Verwendung von StringFormat in Datenbindungen
- Beispiel: Ein Slider-Element, das einen Wert zwischen 0 und 1 repräsentiert, kann mit einem Label verbunden werden, das den Wert mit zwei Dezimalstellen anzeigt

```
<StackLayout Padding="10,0">  
    <Slider x:Name="slider" />  
    <Label Text="{Binding Source={x:Reference slider}, Path=Value,  
        StringFormat='The slider value is {0:F2}}'" />  
</StackLayout>
```

KOMPILIERTE BINDUNGEN

- Kompilierte Bindungen in XAML verbessern die Leistung und Zuverlässigkeit der Datenbindung, indem sie die Bindungsausdrücke zur Kompilierzeit anstatt zur Laufzeit auflösen
- Um kompilierte Bindungen zu nutzen, wird das `x:DataType`-Attribut auf einem `VisualElement` auf den Typ des Objekts gesetzt, an das das `VisualElement` und seine untergeordneten Elemente gebunden werden sollen
- Dieses Attribut sollte auf der gleichen Ebene in der Ansichtshierarchie festgelegt werden, auf der die `BindingContext`-Eigenschaft festgelegt ist

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage
    x:Class="MauiApp1.MainPage"
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:viewModels="clr-namespace:MauiApp1.ViewModels"
    x:DataType="viewModels:MainViewModel" />
```