UGIdotNET

APP MODERNISATION DAY

09/04/2019 — MICROSOFT HOUSE, MILANO

Da Silverlight a Angular senza toccare il backend: fatto!

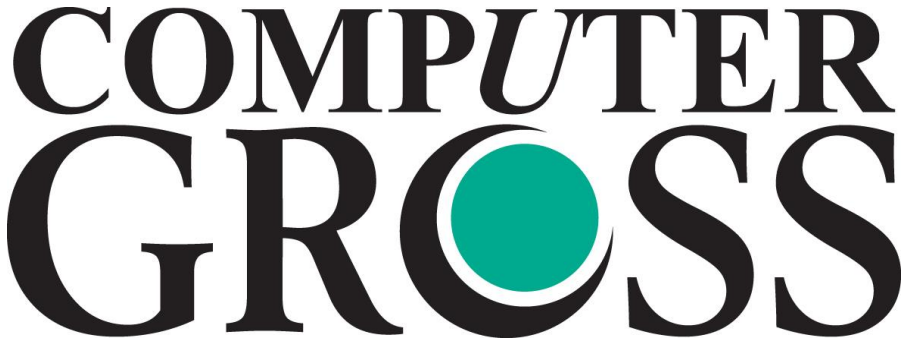
Michele Aponte- CEO/CTO @ Blexin Srl
michele.aponte@blexin.com
https://github.com/apomic80
Twitter: @apomic80

#APPMODERNISATION

#UGIDOTNET

Kudos

UGIdotNET



managed/designs

Un giorno ci chiama un cliente…

Requisiti

- **Back end e Database non si possono toccare**
- Vorremmo mantenere la **stessa UI e UX**
- La UI viene **generata dinamicamente**
- **XAML** (standard e verticale) sul **database**
- Bisogna usare il **nostro «framework»**
- Formare il team interno sulle tecnologie adottate
- Dobbiamo fare una demo tra **6 mesi** a un cliente

Requisiti

- **Back end e Database non si possono toccare**
- Vorremmo mantenere la **stessa UI e UX**
- La UI viene **generata dinamicamente**
- **XAML** (standard e verticale) sul **database**
- Bisogna usare il **nostro «framework»**
- Formare il team interno sulle tecnologie adottate
- Dobbiamo fare una demo tra **6 mesi** a un cliente

UGIdotNET

APP MODERNISATION DAY

09/04/2019 — MICROSOFT HOUSE, MILANO

Perché Angular?

- Dobbiamo realizzare una **Single Page Application**
- Elementi XAML -> Componenti Angular
- C# -> TYPESCRIPT
- XAML Binding -> RxJS e Observable
- Controlli Telerik -> Kendo UI

Kendo UI

THE ART OF WEB DEVELOPMENT

Lo XAML dal database al browser

```
<StackPanel x:Name="Root">
    <StackPanel.Resources>…
    </StackPanel.Resources>
    <StackPanel Margin="0,10,0,0" Orientation="Horizontal">
        <TextBlock Margin="0,5,0,0" Width="60" Text="Nome:" />
        <TextBox x:Name="txtNome" Text="{Binding Path=Nome, Mode=TwoWay}" Width="200" />
    </StackPanel>
    <StackPanel Orientation="Horizontal">…
    </StackPanel>
    <StackPanel Orientation="Horizontal">…
    </StackPanel>
    <StackPanel Orientation="Horizontal">
        <StackPanel Orientation="Horizontal">
            <TextBlock Margin="0,5,0,0" Width="60" Text="Città:" />
            <TextBox x:Name="txtCitta" Text="{Binding Path=Citta, Mode=TwoWay}" Width="200" />
        </StackPanel>
        <StackPanel Orientation="Horizontal">
            <TextBlock Margin="10,5,0,0" Width="60" Text="Provincia:" />
            <TextBox x:Name="txtProvincia" Text="{Binding Path=Provincia, Mode=TwoWay}" Width="70" />
        </StackPanel>
    </StackPanel>
    <StackPanel Orientation="Horizontal">…
    </StackPanel>
    <Grid Visibility="{Binding IsChecked, ElementName=chkMostraNote, Converter={StaticResource BooleanToVisibility}}">…
    </Grid>
</StackPanel>
```

Lo XAML dal database al browser

```
{
    "StackPanel": {
        "@xmlns": "http://schemas.microsoft.com/client/2007",
        "@xmlns:x": "http://schemas.microsoft.com/winfx/2006/xaml",
        "@xmlns:conv": "clr-namespace:Silvelight2Angular.Silverlight.Converters;assembly=Silvelight2Angular.Silverlight",
        "@x:Name": "Root",
        "StackPanel.Resources": {},
        "StackPanel": [
            {
                "@Margin": "0,10,0,0",
                "@Orientation": "Horizontal",
                "TextBlock": {
                    "@Margin": "0,5,0,0",
                    "@Width": "60",
                    "@Text": "Nome:"
                },
                "TextBox": {
                    "@x:Name": "txtNome",
                    "@Text": "{Binding Path=Nome, Mode=TwoWay}",
                    "@Width": "200"
                }
            },
            {},
            {},
            {},
            {}
        ],
        "Grid": {
            "@Visibility": "{Binding IsChecked, ElementName=chkMostraNote, Converter={StaticResource BooleanToVisibility}}",
            "Grid.RowDefinitions": {},
            "Grid.ColumnDefinitions": {},
            "TextBlock": [],
            "TextBox": []
        }
    }
}
```

Lo XAML dal database al browser


```
{
    "StackPanel": {
        "@xmlns": "http://schemas.microsoft.com/client/2007",
        "@xmlns:x": "http://schemas.microsoft.com/winfx/2006/xaml",
        "@xmlns:conv": "clr-namespace:Silvelight2Angular.Silverlight.Converters;assembly=Silvelight2Angular.Silverlight",
        "@x:Name": "Root",
        "StackPanel.Resources": {},
        "StackPanel": [
            {
                "@Margin": "0,10,0,0",
                "@Orientation": "Horizontal",
                "TextBlock": {
                    "@Margin": "0,5,0,0",
                    "@Width": "60",
                    "@Text": "Nome:"
                },
                "TextBox": {
                    "@x:Name": "txtNome",
                    "@Text": "{Binding Path=Nome, Mode=TwoWay}",
                    "@Width": "200"
                }
            },
            {},
            {},
            {},
            {}
        ],
        "Grid": {
            "@Visibility": "{Binding IsChecked, ElementName=chkMostraNote, Converter={StaticResource BooleanToVisibility}}",
            "Grid.RowDefinitions": {},
            "Grid.ColumnDefinitions": {},
            "TextBlock": [],
            "TextBox": []
        }
    }
}
```


UGIdotNET

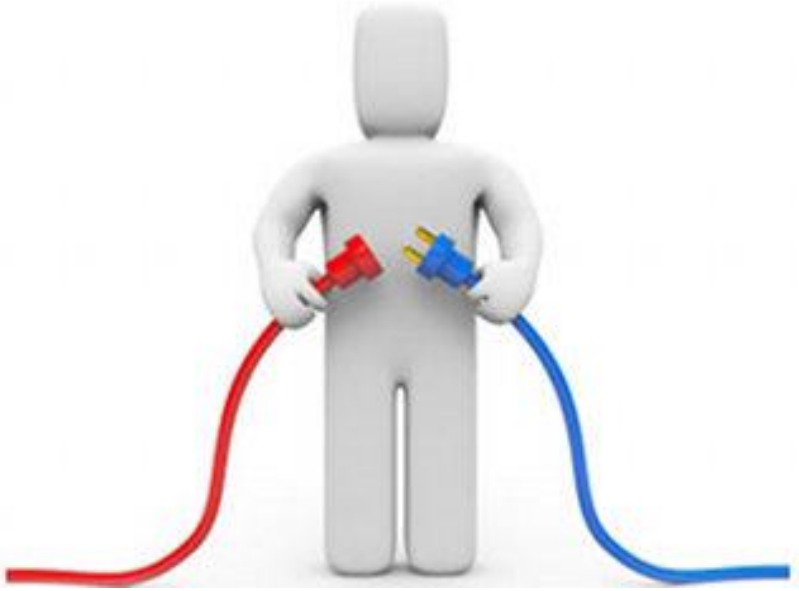
APP MODERNISATION DAY

09/04/2019 — MICROSOFT HOUSE, MILANO

Dal Binding XAML al Binding Angular

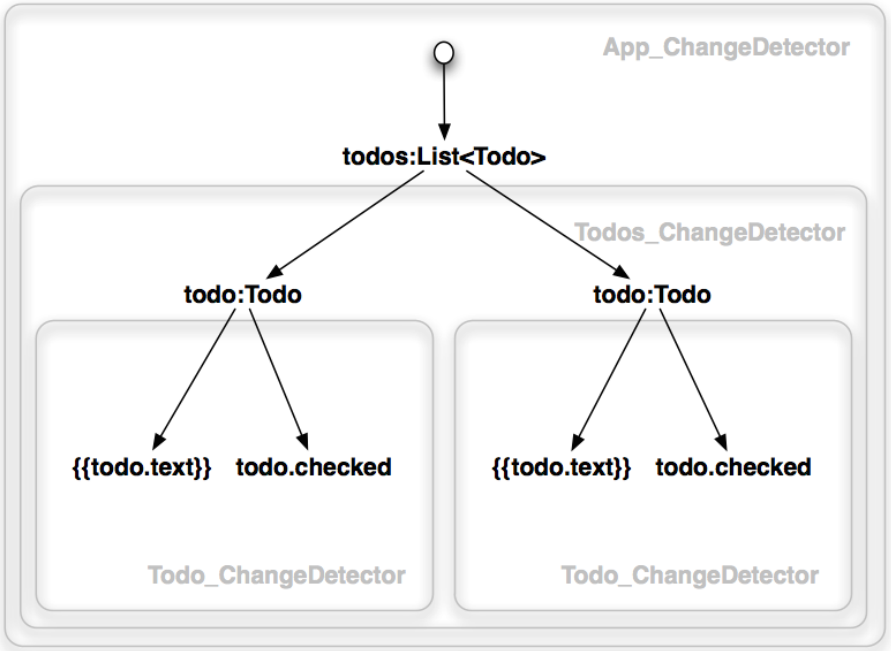
```
<TextBox x:Name="txtNome"
         Text="{Binding Path=Nome, Mode=TwoWay}"
         Width="200" />
```

```
"TextBox": {
    "@x:Name": "txtNome",
    "@Text": "{Binding Path=Nome, Mode=TwoWay}"
    "@Width": "200"
}
```

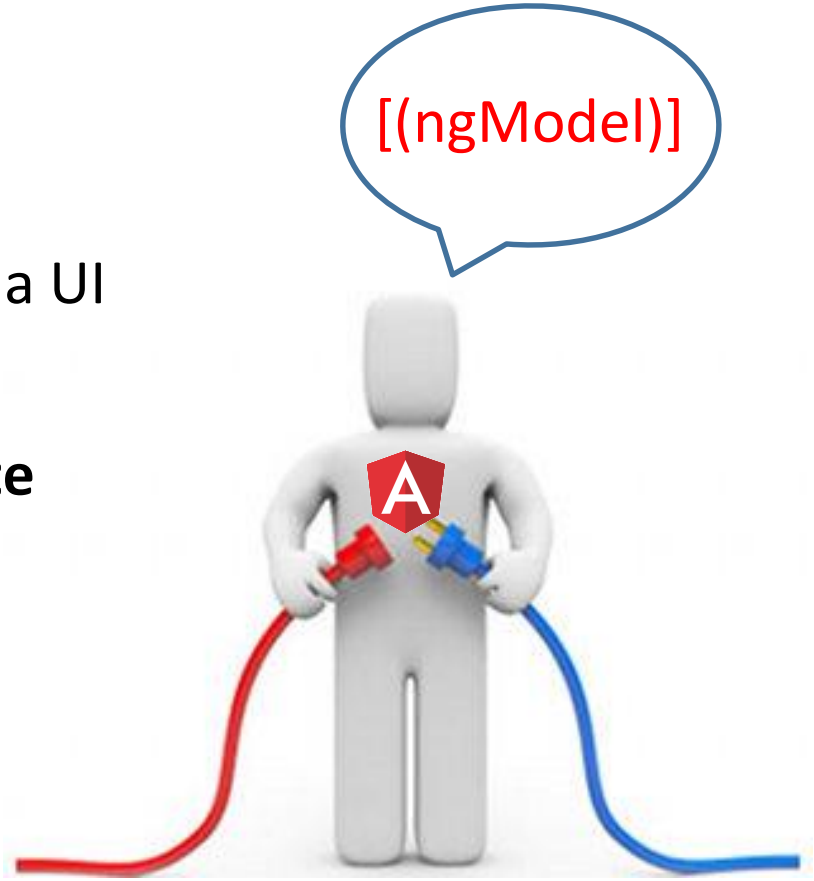


Dal Binding XAML al Binding Angular

Angular **si accorge delle modifiche** agli elementi bindati con la **Change Detection:**



- Aggiorna automaticamente la UI
- **Bello, ma costoso**
- Si può **parzialmente disabilitare**



UGIdotNET

APP MODERNISATION DAY

09/04/2019 — MICROSOFT HOUSE, MILANO

Binding tra elementi e Converter

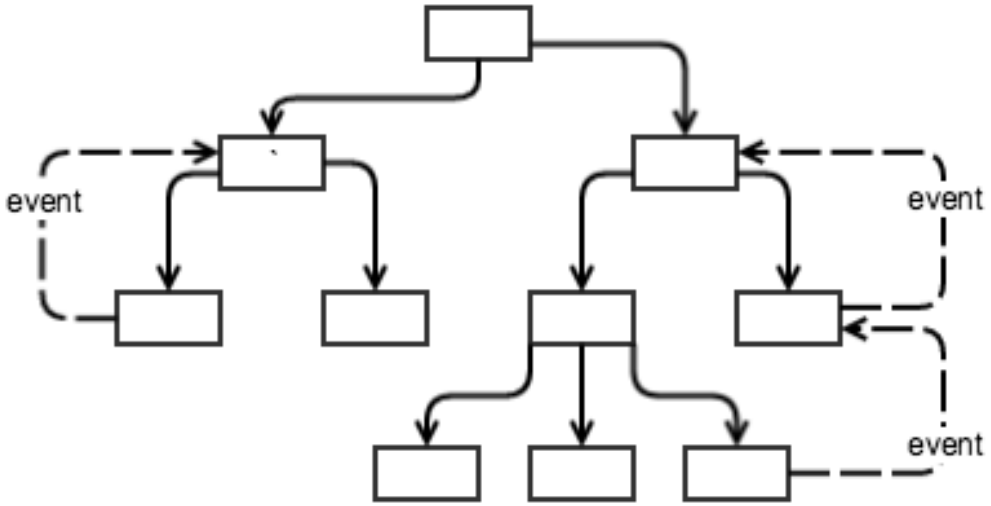
```
<StackPanel Orientation="Horizontal">
    <TextBlock Margin="0,5,0,0" Width="120" Text="Visualizza Note:" />
    <CheckBox x:Name="chkMostraNote" Margin="0,5,0,0" IsChecked="True" />
</StackPanel>
<Grid Visibility="{Binding IsChecked, ElementName=chkMostraNote, Converter={StaticResource BooleanToVisibility}}">…
</Grid>
```

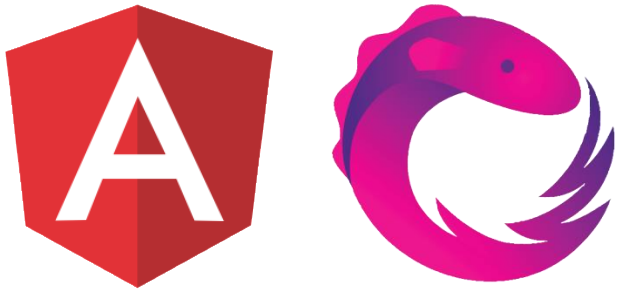
```
        "CheckBox": {
            "@x:Name": "chkMostraNote",
            "@Margin": "0,5,0,0",
            "@IsChecked": "True"
        }
    }
],
"Grid": {
    "@Visibility": "{Binding IsChecked, ElementName=chkMostraNote, Converter={StaticResource BooleanToVisibility}}",
    "Grid.RowDefinitions": {…
    },
```

Binding tra elementi e Converter

I componenti devono essere **autonomi** ma devono poter **comunicare** tra loro:

- **padre/figlio** => **@Input** e **@Output**
- **fratelli** => il **padre** come **mediatore** e **@Input** e **@Output**
- **relazione qualsiasi** => **Dependency Injection**, **Observable** e operatori **RxJS**





UGIdotNET

APP MODERNISATION DAY

09/04/2019 — MICROSOFT HOUSE, MILANO

UGIdotNET

APP MODERNISATION DAY

09/04/2019 — MICROSOFT HOUSE, MILANO

Grazie a tutti per la pazienza e …

… sparate pure sul pianista!

APP MODERNISATION DAY

09/04/2019 — MICROSOFT HOUSE, MILANO

L'EVENTO È STATO POSSIBILE GRAZIE A:

COMPUTER GROSS

managed/designs