

# HOW TO –

## A. Creatie van WPF MVVM Applicatie

(Voorbeeldapplicatie van bieren, Brouwers en soorten bieren)

<https://github.com/CSharpSyntraWest/HOWTO-WPF-MVVM>

1. Maak een nieuwe WPF Applicatie aan (.Net Framework of .Net Core)
2. Maak 3 folders aan in het project: **Models**, **Views** en **ViewModels**
3. Versleep **MainView.xaml** naar de folder **Views**
4. Pas de **StartupUri** aan in **App.xaml**  
`StartupUri="Views/MainWindow.xaml"`
5. Maak een nieuwe folder **Utilities** en plaats hieronder de klasse **ObservableObject** :

```
public class ObservableObject : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    protected virtual void OnPropertyChanged([CallerMemberName] string propertyName = "")
    {
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
    }

    protected virtual bool OnPropertyChanged<T>(ref T backingField, T value, [CallerMemberName] string propertyName = "")
    {
        if (EqualityComparer<T>.Default.Equals(backingField, value))
            return false;

        backingField = value;
        OnPropertyChanged(propertyName);
        return true;
    }
}
```

6. Maak klassen **Bier**, **Brouwer** en **SoortBier** aan onder de folder **Models**. Deze klassen zijn afgeleid van **ObservableObject**, want we willen een 2-way binding kunnen leggen met objecten van deze klassen met de Views (UI). We vullen later de code verder aan.

```
public class Bier : ObservableObject
{
}

public class BierSoort : ObservableObject
{
}

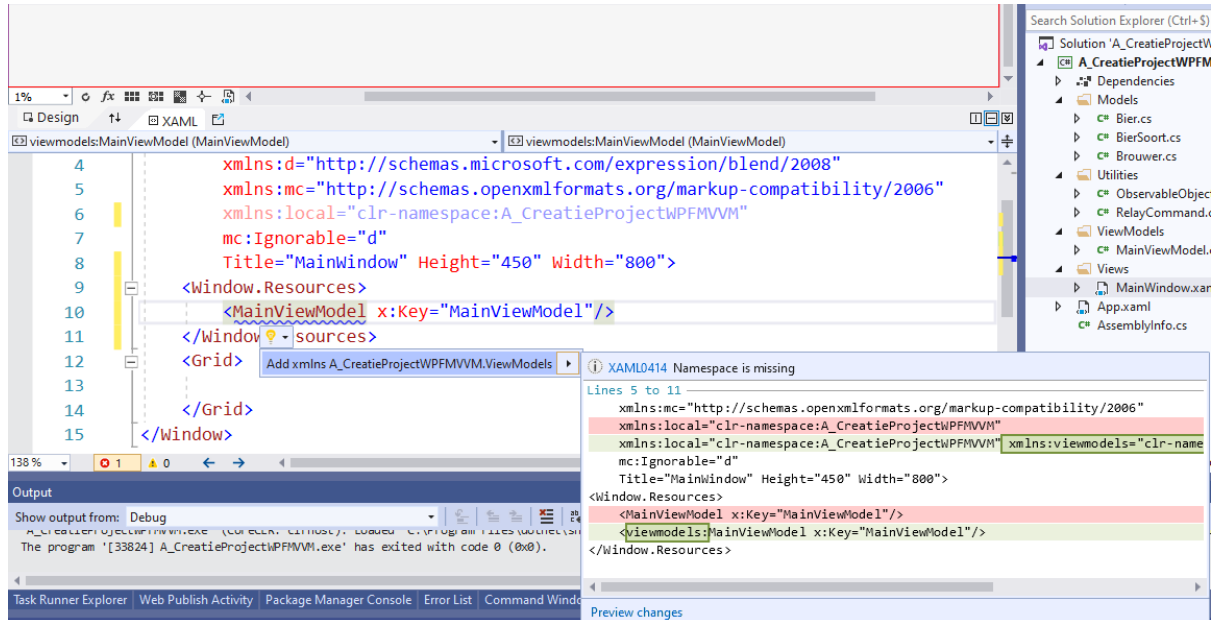
public class Brouwer : ObservableObject
{
}
```

7. Maak onder de folder **ViewModels** een nieuwe klasse **MainViewModel** aan die afgeleid is van de klasse **ObservableObject**

```
class MainViewModel:ObservableObject
{
}

}
```

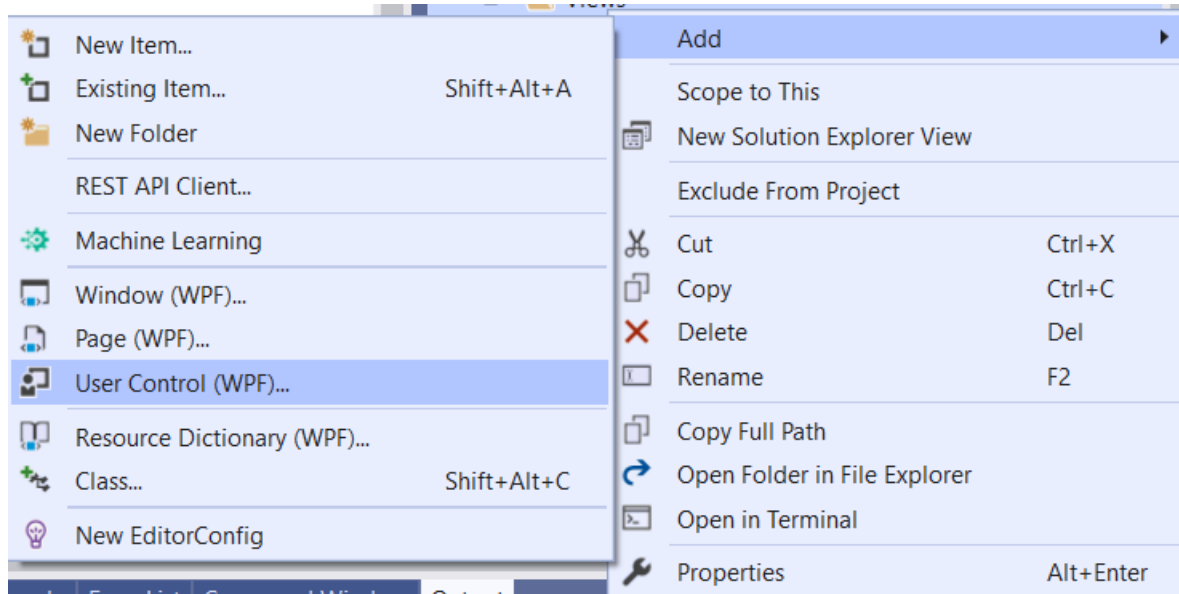
8. Open **MainView.xaml** en maak een Window.Resource aan voor de **MainViewModel** tussen de tags `<Window.Resources></Window.Resources>`
9. Rechtsklik met de muis op de **MainViewModel** tag en selecteer "Add xmlns.....ViewModels" om de namespace van de MainViewModel klasse bovenaan in de xaml toe te voegen :



Nadat je dit hebt gedaan, is er een xmlns toegevoegd aan MainWindow.xaml. De xmlns attribueert verwijst naar de namespace van de klasse MainViewModel. Deze kan gerefereerd worden in de xaml code via de prefix **viewmodels**:

```
<Window x:Class="A_CreatieProjectWPFMVVM.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:A_CreatieProjectWPFMVVM"
    xmlns:viewmodels="clr-namespace:A_CreatieProjectWPFMVVM.ViewModels"
    mc:Ignorable="d"
    Title="MainWindow" Height="450" Width="800">
    <Window.Resources>
        <viewmodels:MainViewModel x:Key="MainViewModel"/>
    </Window.Resources>
    <Grid>
    </Grid>
</Window>
```

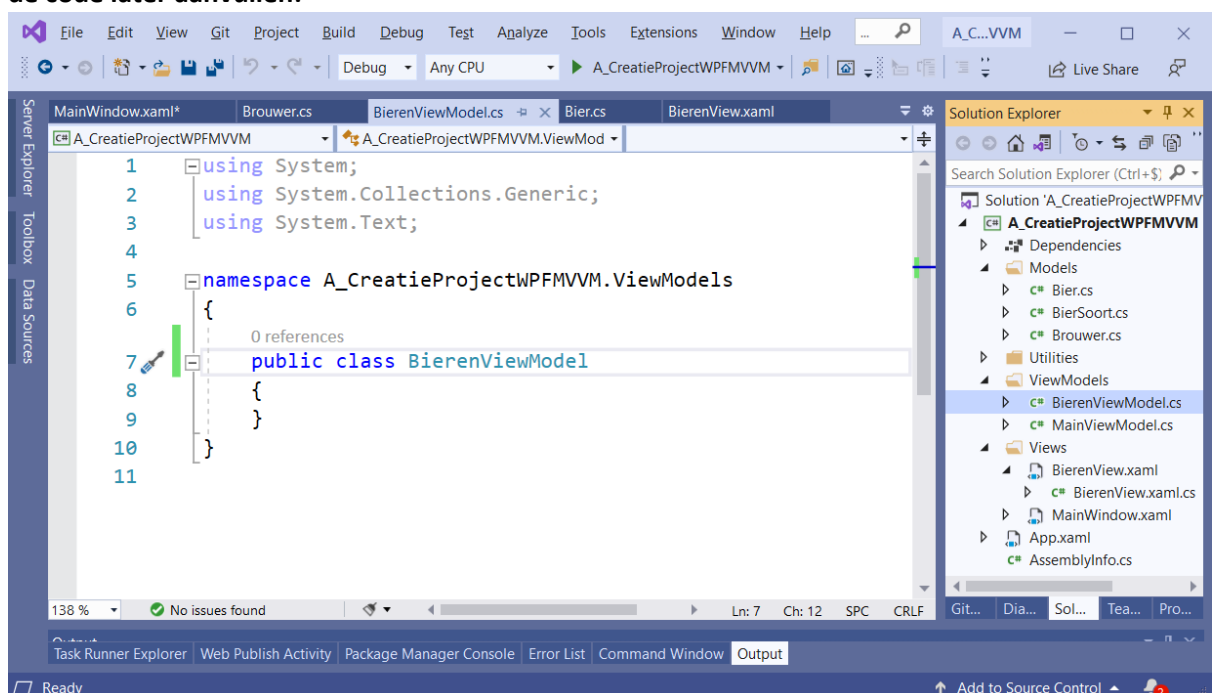
10. Maak onder de folder **Views** een **usercontrol** met naam **BierenView**:



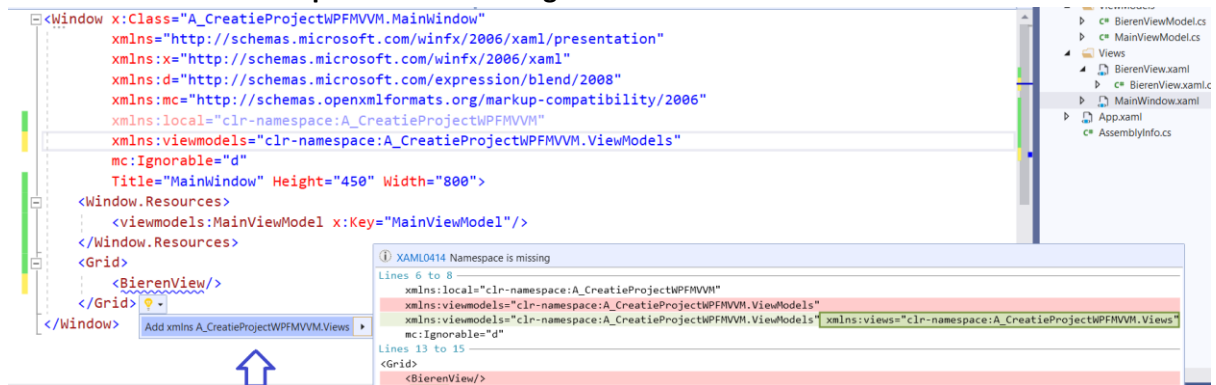
11. Open **BierenView.xaml** en zet een textblock met **Text="Hier komt de BierenView"**:



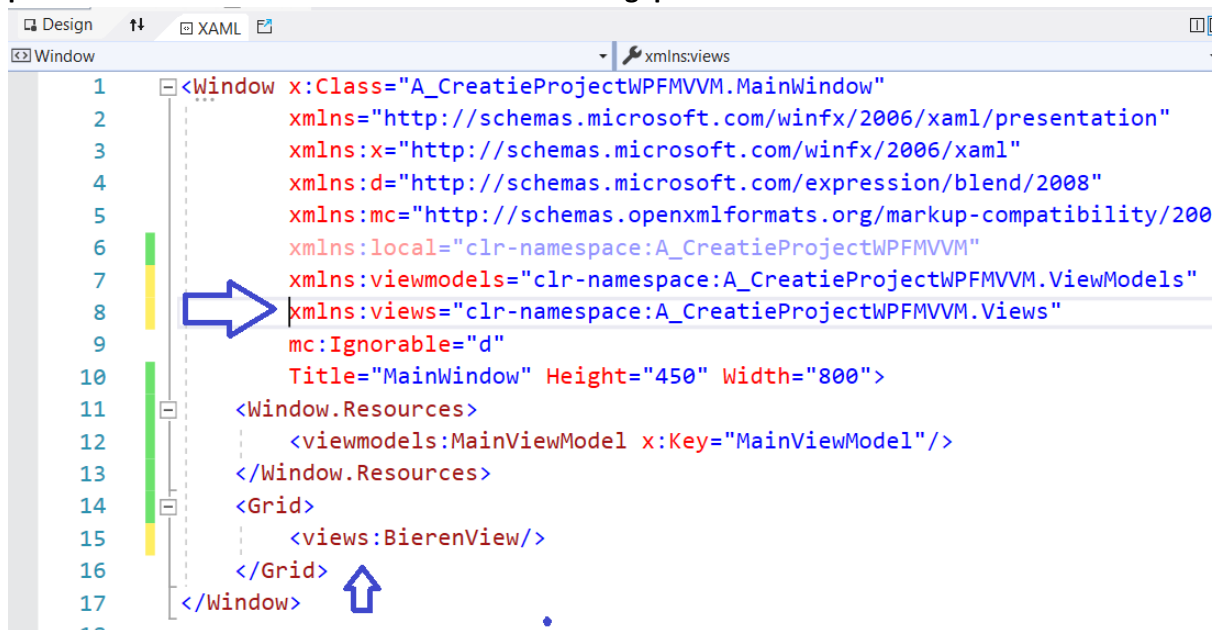
12. Maak een nieuwe klasse met naam **BierenViewModel** onder de folder **ViewModels** – we zullen de code later aanvullen:



13. Open MainWindow.xaml en plaats tussen de <Grid></Grid> tags je usercontrol BierenView.  
Rechtsklik met de muis op de BierenView tag en selecteer « Add xmlns....Views ».



14. Nu wordt de namespace van de BierenView toegevoegd bovenaan via een xmlns tag en de prefix views : wordt automatisch voor BierenView geplaatst :



15. Run de app, je krijgt het volgende scherm te zien :  
16.

