RSCG nr 8 : CommunityToolkit.Mvvm

Info

Nuget : https://www.nuget.org/packages/CommunityToolkit.Mvvm

You can find more details at : https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/overview

Author :Microsoft

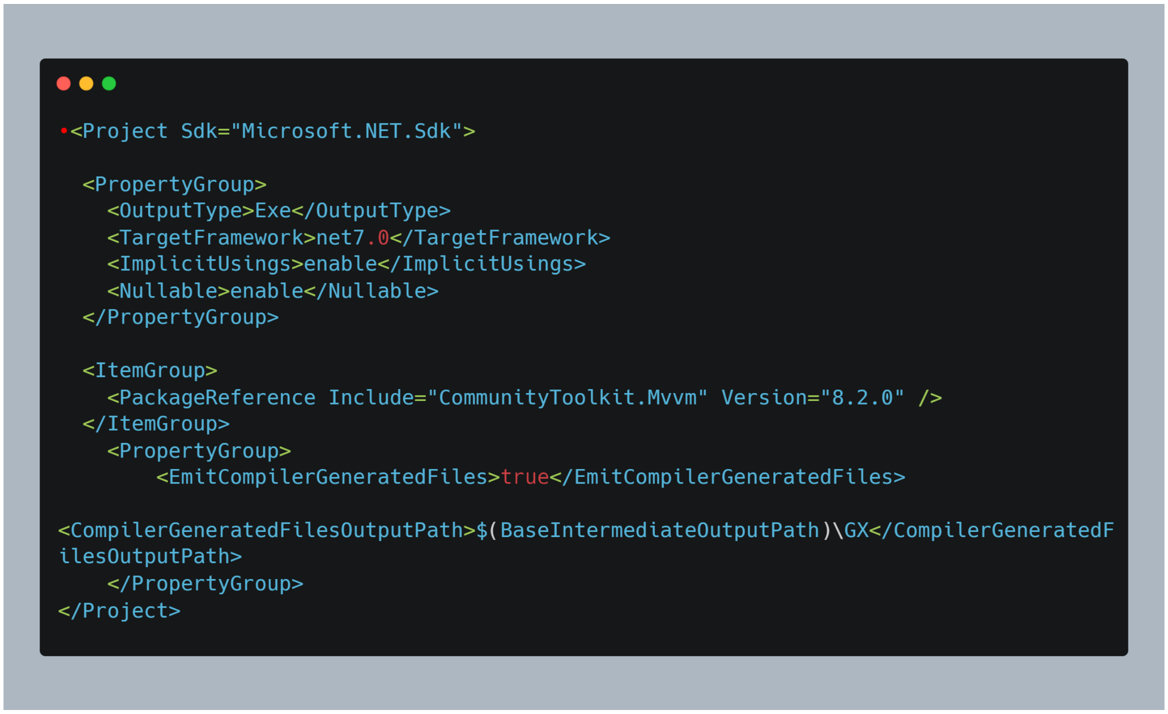
Source : https://github.com/CommunityToolkit/dotnet

About

Shows how to implement INotifyPropertyChanged,ObservableProperty and RelayCommandUnfortunately , not yet a separate package just for those.Also, this show that RSCG could generate multiple partial declarations

How to use

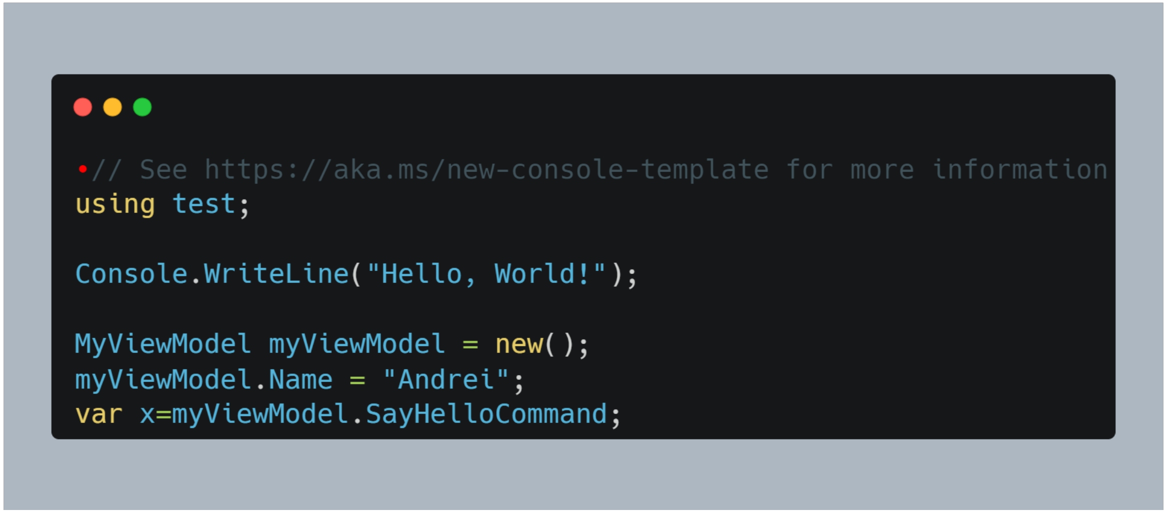
Add reference to the CommunityToolkit.Mvvm in the csproj



```xml
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net7.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="CommunityToolkit.Mvvm" Version="8.2.0" />
  </ItemGroup>
    <PropertyGroup>
        <EmitCompilerGeneratedFiles>true</EmitCompilerGeneratedFiles>

<CompilerGeneratedFilesOutputPath>$(BaseIntermediateOutputPath)\GX</CompilerGeneratedFilesOutputPath>
    </PropertyGroup>
</Project>
```

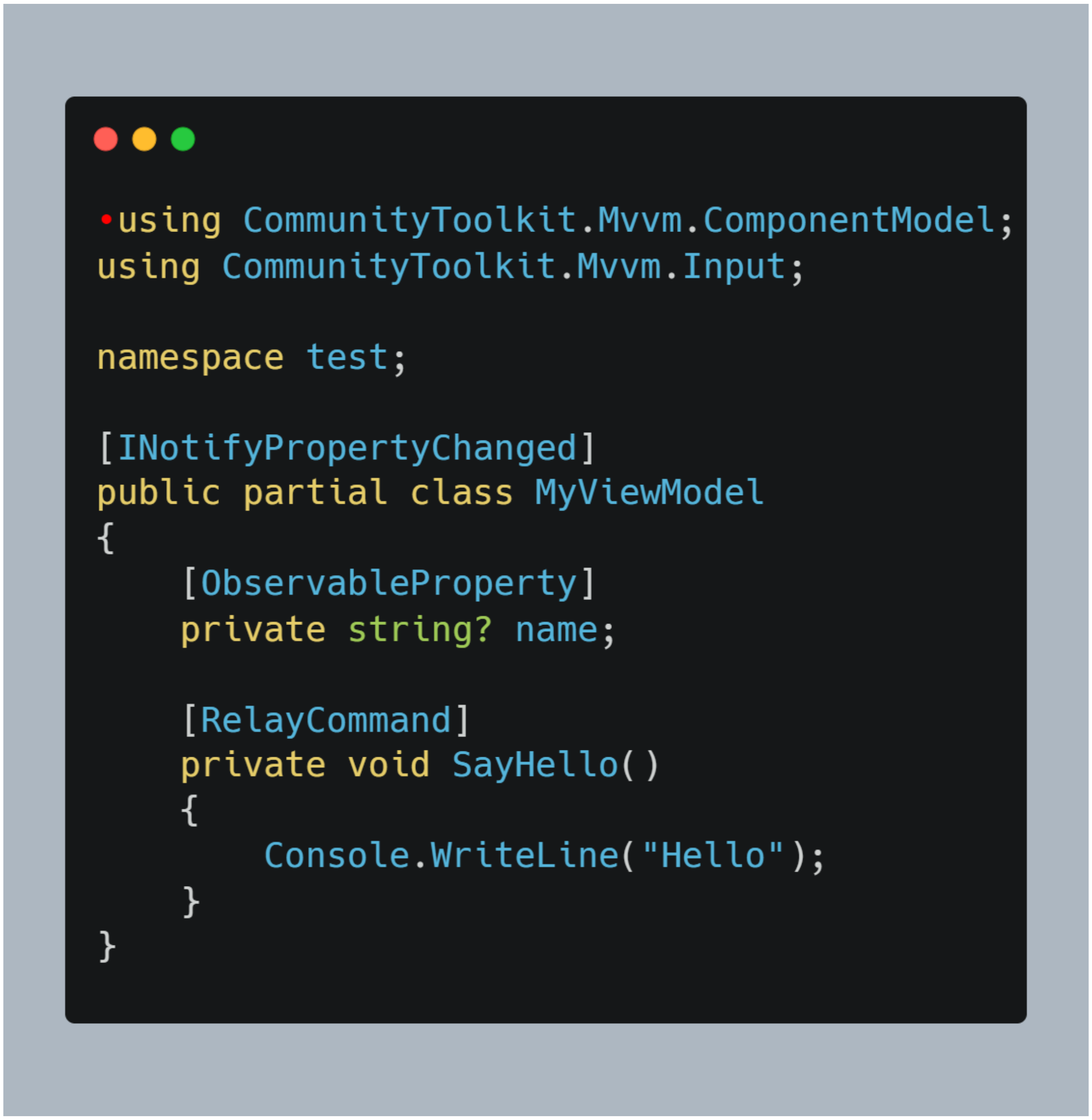
This was for me the starting code

I have **coded** the file Program.cs



```
// See https://aka.ms/new-console-template for more information
using test;

Console.WriteLine("Hello, World!");

MyViewModel myViewModel = new();
myViewModel.Name = "Andrei";
var x=myViewModel.SayHelloCommand;
```

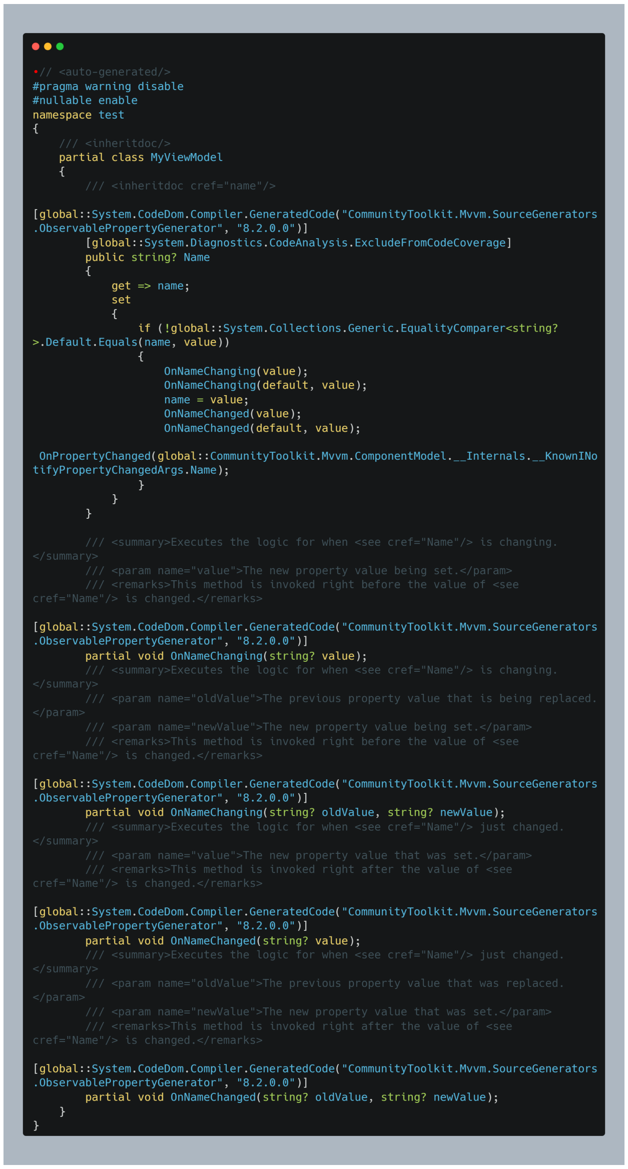
I have **coded** the file MyViewModel.cs



```csharp
using CommunityToolkit.Mvvm.ComponentModel;
using CommunityToolkit.Mvvm.Input;

namespace test;

[INotifyPropertyChanged]
public partial class MyViewModel
{
    [ObservableProperty]
    private string? name;

    [RelayCommand]
    private void SayHello()
    {
        Console.WriteLine("Hello");
    }
}
```

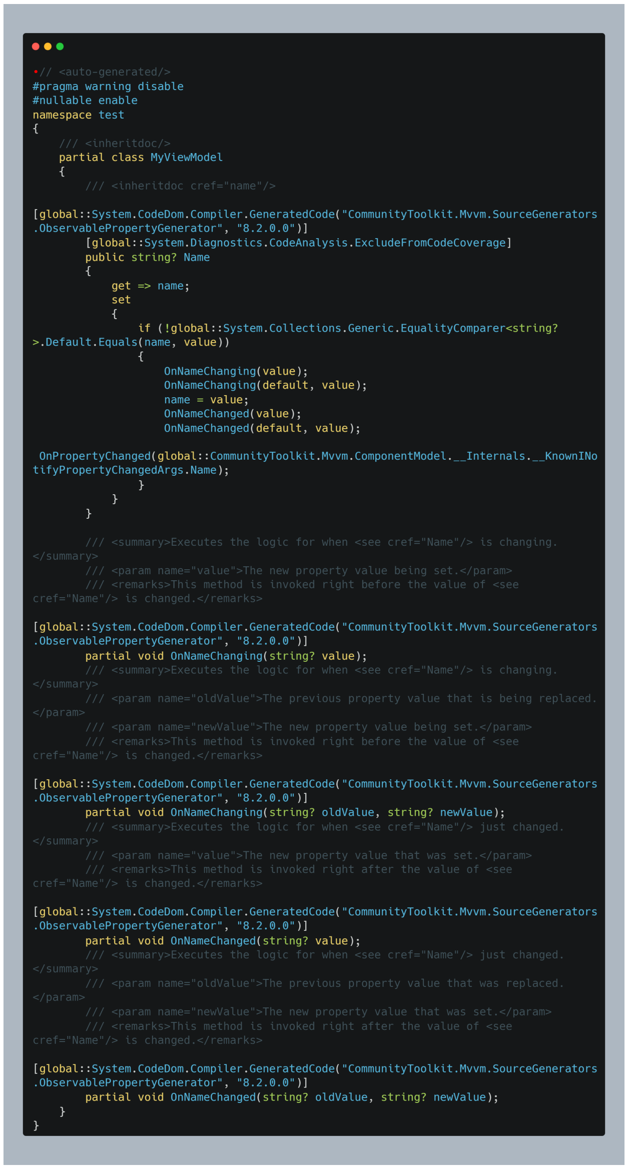
And here are the *generated* files

The file *generated* is test.MyViewModel.g.cs



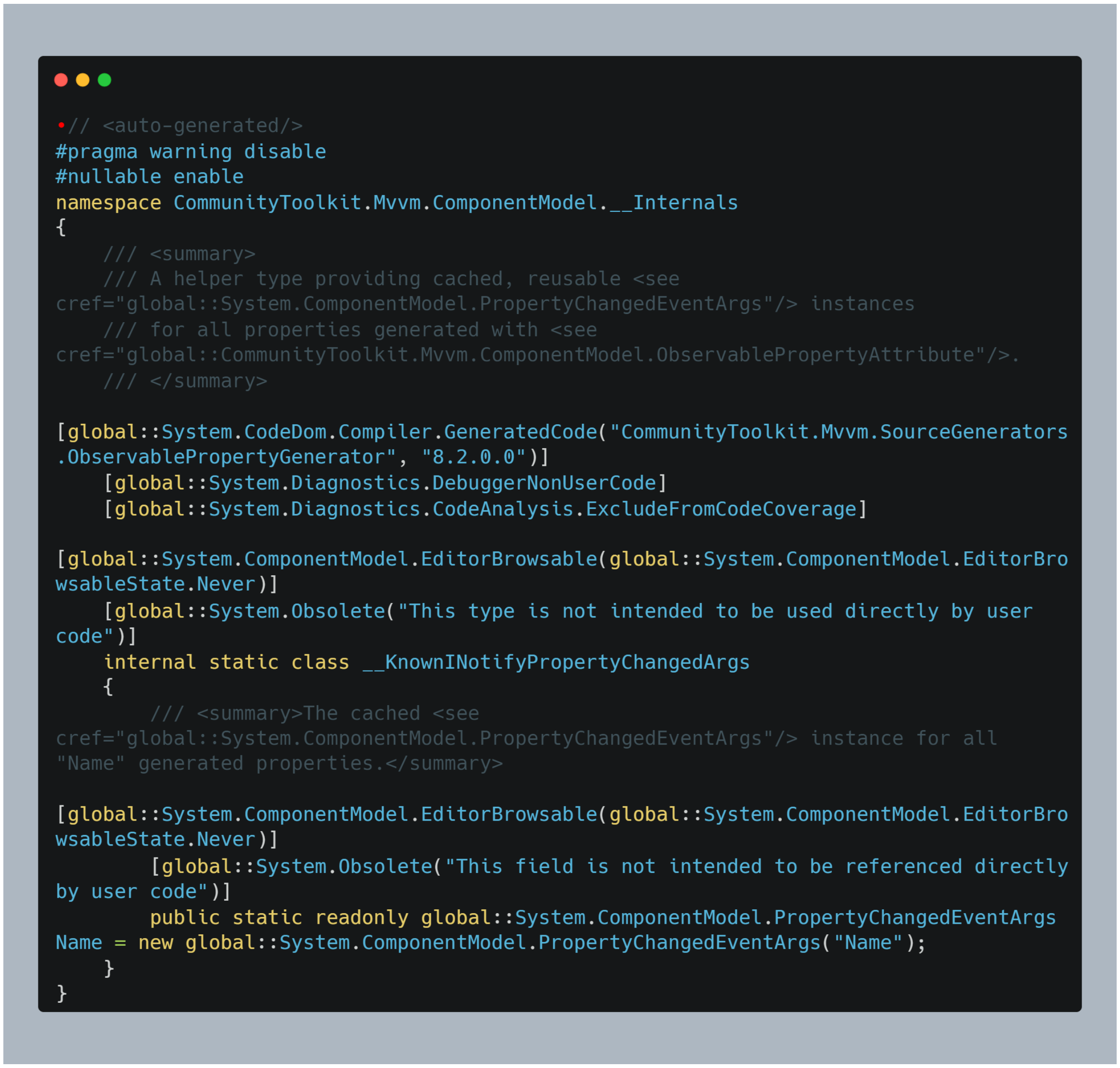
```
// <auto-generated/>
#pragma warning disable
#nullable enable
namespace test
{
    /// <inheritdoc/>
    partial class MyViewModel
    {
        /// <inheritdoc cref="name"/>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.ObservablePropertyGenerator", "8.2.0.0")]
        [global::System.Diagnostics.CodeAnalysis.ExcludeFromCodeCoverage]
        public string? Name
        {
            get => name;
            set
            {
                if (!global::System.Collections.Generic.EqualityComparer<string?
>.Default.Equals(name, value))
                {
                    OnNameChanging(value);
                    OnNameChanging(default, value);
                    name = value;
                    OnNameChanged(value);
                    OnNameChanged(default, value);

 OnPropertyChanged(global::CommunityToolkit.Mvvm.ComponentModel.__Internals.__KnownINo
tifyPropertyChangedArgs.Name);
                }
            }
        }

        /// <summary>Executes the logic for when <see cref="Name"/> is changing.
</summary>
        /// <param name="value">The new property value being set.</param>
        /// <remarks>This method is invoked right before the value of <see
cref="Name"/> is changed.</remarks>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.ObservablePropertyGenerator", "8.2.0.0")]
        partial void OnNameChanging(string? value);
        /// <summary>Executes the logic for when <see cref="Name"/> is changing.
</summary>
        /// <param name="oldValue">The previous property value that is being replaced.
</param>
        /// <param name="newValue">The new property value being set.</param>
        /// <remarks>This method is invoked right before the value of <see
cref="Name"/> is changed.</remarks>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.ObservablePropertyGenerator", "8.2.0.0")]
        partial void OnNameChanging(string? oldValue, string? newValue);
        /// <summary>Executes the logic for when <see cref="Name"/> just changed.
</summary>
        /// <param name="value">The new property value that was set.</param>
        /// <remarks>This method is invoked right after the value of <see
cref="Name"/> is changed.</remarks>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.ObservablePropertyGenerator", "8.2.0.0")]
        partial void OnNameChanged(string? value);
        /// <summary>Executes the logic for when <see cref="Name"/> just changed.
</summary>
        /// <param name="oldValue">The previous property value that was replaced.
</param>
        /// <param name="newValue">The new property value that was set.</param>
        /// <remarks>This method is invoked right after the value of <see
cref="Name"/> is changed.</remarks>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.ObservablePropertyGenerator", "8.2.0.0")]
        partial void OnNameChanged(string? oldValue, string? newValue);
    }
}
```

The file *generated* is test.MyViewModel.g.cs



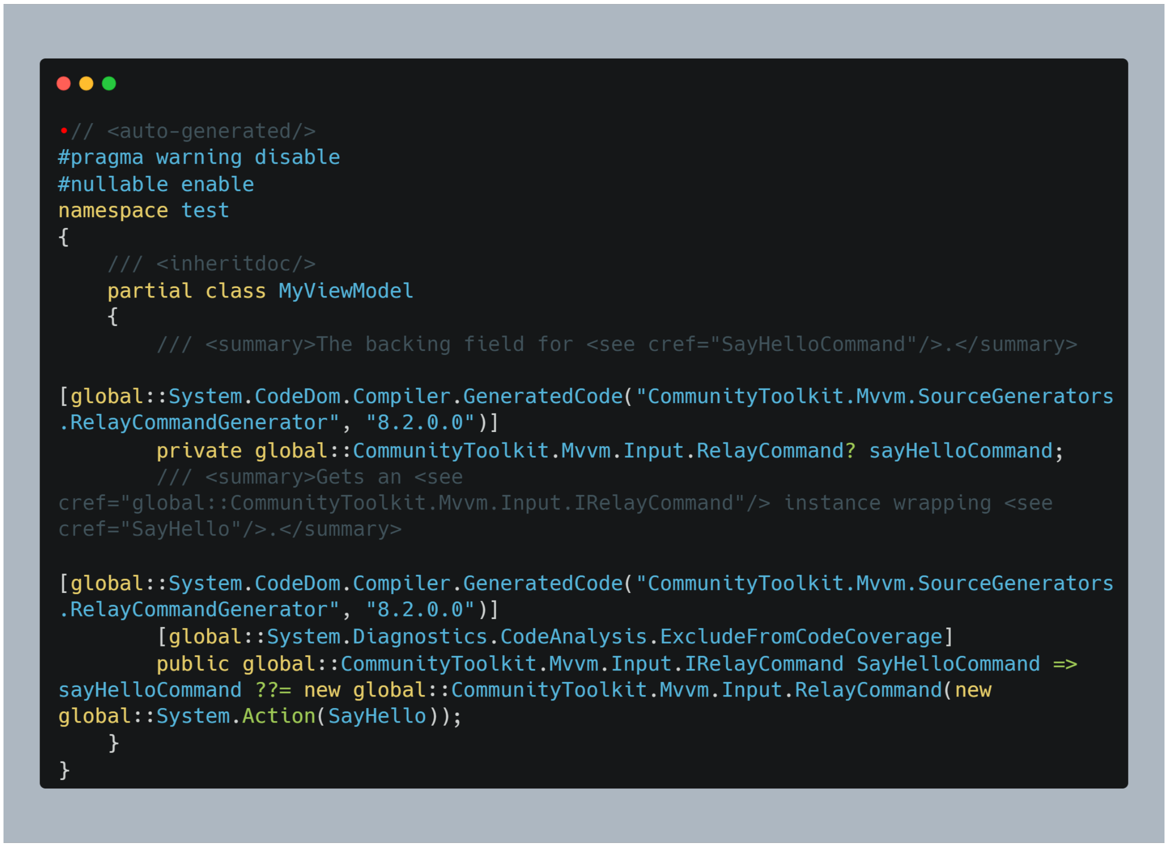
```
// <auto-generated/>
#pragma warning disable
#nullable enable
namespace test
{
    /// <inheritdoc/>
    partial class MyViewModel
    {
        /// <inheritdoc cref="name"/>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.ObservablePropertyGenerator", "8.2.0.0")]
        [global::System.Diagnostics.CodeAnalysis.ExcludeFromCodeCoverage]
        public string? Name
        {
            get => name;
            set
            {
                if (!global::System.Collections.Generic.EqualityComparer<string?
>.Default.Equals(name, value))
                {
                    OnNameChanging(value);
                    OnNameChanging(default, value);
                    name = value;
                    OnNameChanged(value);
                    OnNameChanged(default, value);

 OnPropertyChanged(global::CommunityToolkit.Mvvm.ComponentModel.__Internals.__KnownINo
tifyPropertyChangedArgs.Name);
                }
            }
        }

        /// <summary>Executes the logic for when <see cref="Name"/> is changing.
</summary>
        /// <param name="value">The new property value being set.</param>
        /// <remarks>This method is invoked right before the value of <see
cref="Name"/> is changed.</remarks>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.ObservablePropertyGenerator", "8.2.0.0")]
        partial void OnNameChanging(string? value);
        /// <summary>Executes the logic for when <see cref="Name"/> is changing.
</summary>
        /// <param name="oldValue">The previous property value that is being replaced.
</param>
        /// <param name="newValue">The new property value being set.</param>
        /// <remarks>This method is invoked right before the value of <see
cref="Name"/> is changed.</remarks>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.ObservablePropertyGenerator", "8.2.0.0")]
        partial void OnNameChanging(string? oldValue, string? newValue);
        /// <summary>Executes the logic for when <see cref="Name"/> just changed.
</summary>
        /// <param name="value">The new property value that was set.</param>
        /// <remarks>This method is invoked right after the value of <see
cref="Name"/> is changed.</remarks>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.ObservablePropertyGenerator", "8.2.0.0")]
        partial void OnNameChanged(string? value);
        /// <summary>Executes the logic for when <see cref="Name"/> just changed.
</summary>
        /// <param name="oldValue">The previous property value that was replaced.
</param>
        /// <param name="newValue">The new property value that was set.</param>
        /// <remarks>This method is invoked right after the value of <see
cref="Name"/> is changed.</remarks>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.ObservablePropertyGenerator", "8.2.0.0")]
        partial void OnNameChanged(string? oldValue, string? newValue);
    }
}
```

The file *generated* is __KnownINotifyPropertyChangedArgs.g.cs



```
•// <auto-generated/>
#pragma warning disable
#nullable enable
namespace CommunityToolkit.Mvvm.ComponentModel.__Internals
{
    /// <summary>
    /// A helper type providing cached, reusable <see
cref="global::System.ComponentModel.PropertyChangedEventArgs"/> instances
    /// for all properties generated with <see
cref="global::CommunityToolkit.Mvvm.ComponentModel.ObservablePropertyAttribute"/>.
    /// </summary>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.ObservablePropertyGenerator", "8.2.0.0")]
    [global::System.Diagnostics.DebuggerNonUserCode]
    [global::System.Diagnostics.CodeAnalysis.ExcludeFromCodeCoverage]

[global::System.ComponentModel.EditorBrowsable(global::System.ComponentModel.EditorBro
wsableState.Never)]
    [global::System.Obsolete("This type is not intended to be used directly by user
code")]
    internal static class __KnownINotifyPropertyChangedArgs
    {
        /// <summary>The cached <see
cref="global::System.ComponentModel.PropertyChangedEventArgs"/> instance for all
"Name" generated properties.</summary>

[global::System.ComponentModel.EditorBrowsable(global::System.ComponentModel.EditorBro
wsableState.Never)]
        [global::System.Obsolete("This field is not intended to be referenced directly
by user code")]
        public static readonly global::System.ComponentModel.PropertyChangedEventArgs
Name = new global::System.ComponentModel.PropertyChangedEventArgs("Name");
    }
}
```

The file *generated* is test.MyViewModel.SayHello.g.cs



```
// <auto-generated/>
#pragma warning disable
#nullable enable
namespace test
{
    /// <inheritdoc/>
    partial class MyViewModel
    {
        /// <summary>The backing field for <see cref="SayHelloCommand"/>.</summary>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.RelayCommandGenerator", "8.2.0.0")]
        private global::CommunityToolkit.Mvvm.Input.RelayCommand? sayHelloCommand;
        /// <summary>Gets an <see
cref="global::CommunityToolkit.Mvvm.Input.IRelayCommand"/> instance wrapping <see
cref="SayHello"/>.</summary>

[global::System.CodeDom.Compiler.GeneratedCode("CommunityToolkit.Mvvm.SourceGenerators
.RelayCommandGenerator", "8.2.0.0")]
        [global::System.Diagnostics.CodeAnalysis.ExcludeFromCodeCoverage]
        public global::CommunityToolkit.Mvvm.Input.IRelayCommand SayHelloCommand =>
sayHelloCommand ??= new global::CommunityToolkit.Mvvm.Input.RelayCommand(new
global::System.Action(SayHello));
    }
}
```

You can download the code and this page as pdf from https://ignatandrei.github.io/RSCG_Examples/v2/docs/CommunityToolkit.Mvvm

You can see the whole list at https://ignatandrei.github.io/RSCG_Examples/v2/docs/List-of-RSCG