tutorial

Windows App SDK

Lucky Lotto

Lucky Lotto shows how you can generate randomised lottery numbers and display these using a control from **NuGet** using the **Windows App SDK**.
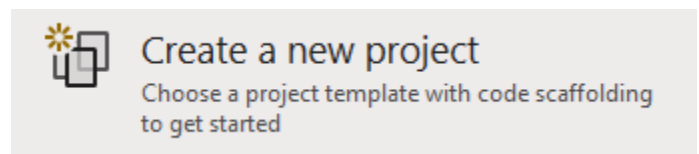
Step 1

Follow **Setup and Start** on how to get **Setup** and **Install** what you need for **Visual Studio 2022** and **Windows App SDK**.
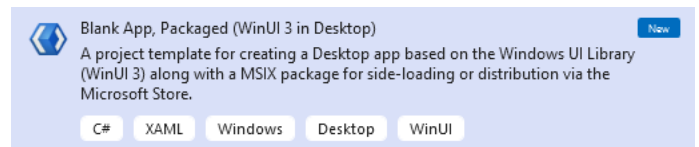
In **Windows 11** choose **Start** and then find or search for **Visual Studio 2022** and then select it.
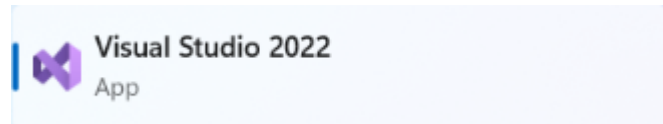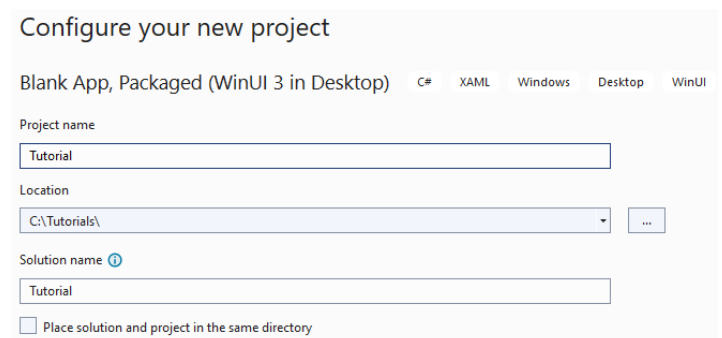

Once **Visual Studio 2022** has started select **Create a new project**.

Then choose the **Blank App, Packages (WinUI in Desktop)** and then select **Next**.

After that in **Configure your new project** type in the **Project name** as *LuckyLotto*, then select a Location and then select **Create** to start a new **Solution**.
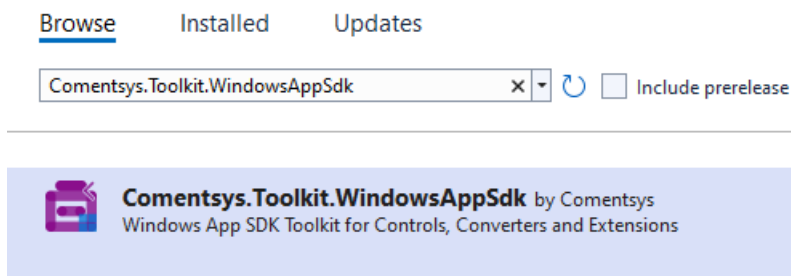
Step 2

Then in **Visual Studio** within **Solution Explorer** for the **Solution**, right click on the **Project** shown below the **Solution** and then select **Manage NuGet Packages...**
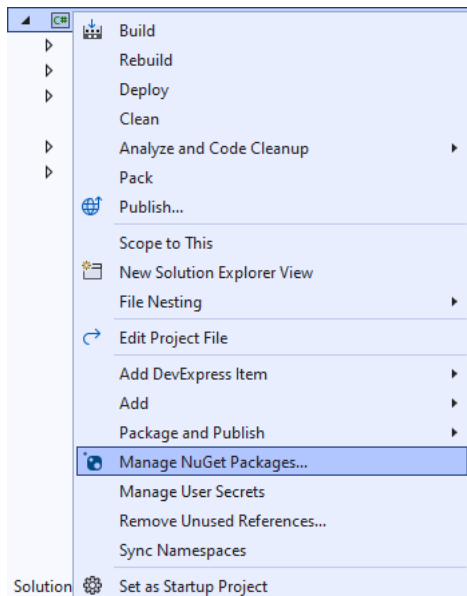
Step 3

Then in the **NuGet Package Manager** from the **Browse** tab search for **Comentsys.Toolkit.WindowsAppSdk** and then select **Comentsys.Toolkit.WindowsAppSdk by Comentsys** as indicated and select **Install**

This will add the package for **Comentsys.Toolkit.WindowsAppSdk** to your **Project**. If you get the **Preview Changes** screen saying **Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.** You can read the message and then select **OK** to **Install** the package, then you can close the **tab** for **Nuget: LuckyLotto** by selecting the **x** next to it.

Step 4

Then in **Visual Studio** within **Solution Explorer** for the **Solution**, right click on the **Project** shown below the **Solution** and then select **Add** then **New Item...**

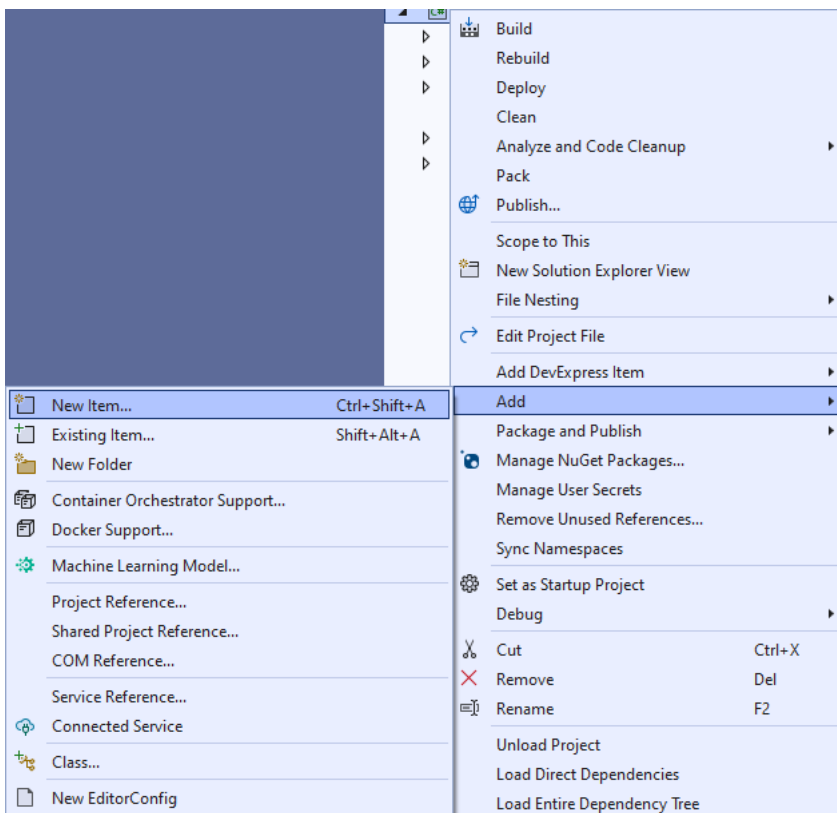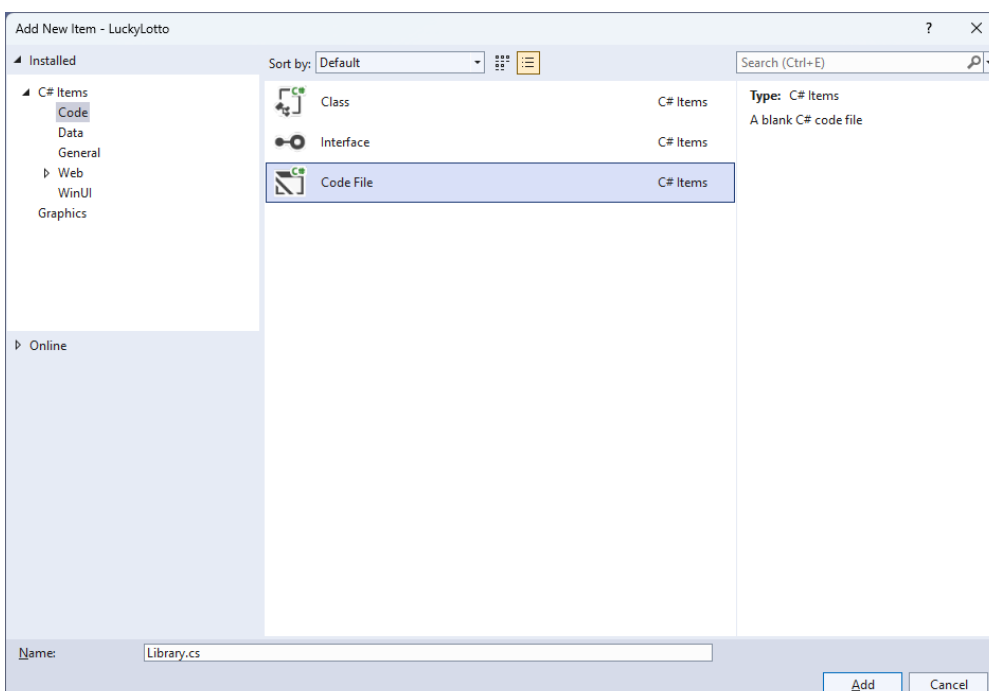## Step 5

Then in **Add New Item** from the **C# Items** list, select **Code** and then select **Code File** from the list next to this, then type in the name of *Library.cs* and then **Click** on **Add**.

Step 6

You will now be in the **View** for the **Code** of *Library.cs*, within this first type the following **Code**:

```csharp
using Comentsys.Toolkit.WindowsAppSdk;
using Microsoft.UI;
using Microsoft.UI.Xaml;
using Microsoft.UI.Xaml.Controls;
using Microsoft.UI.Xaml.Media;
using System;
using System.Collections.Generic;
using System.Linq;
using Windows.UI;

public class Library
{
    private static readonly Dictionary<int, Color> _style = new()
    {
        { 0, Colors.White },
        { 10, Colors.RoyalBlue },
        { 20, Colors.HotPink },
        { 30, Colors.MediumSpringGreen },
        { 40, Colors.Gold },
        { 50, Colors.Indigo }
    };
    private readonly Random _random = new((int)DateTime.UtcNow.Ticks);

    private List<int> Choose(int minimum, int maximum, int total) =>
        Enumerable.Range(minimum, maximum)
            .OrderBy(r => _random.Next(minimum, maximum))
                .Take(total).ToList();

    // Other Methods

}
```

The **Class** that has been defined in so far *Library.cs* has **using** for the package of
Comentsys.Toolkit.WindowsAppSdk amongst others needed. Then there is a **Dictionary** of **_style** to
represent the different **Colours** of **Lottery Balls** but it can be changed to match the numbers in your
Country, these are the ones used in the UK and **Random** which will be used to select randomised numbers
from. Then there is **Choose** which will generate an enumerable of numbers, in this case of **int** then it will
use **Random** to shuffle them randomly and then can just return the amount of numbers needed, using **Take**
and then convert this to a **List** that will be returned using Arrow Syntax with the **=>** for an Expression Body
based **Method**.

Step 7

While still in the **Class** for *Library.cs* and after the **Comment** of **`// Other Methods`** type in the following other **Methods**:

```csharp
private void Add(StackPanel panel, int value)
{
    Color style = _style.Where(w => value > w.Key)
        .Select(s => s.Value).LastOrDefault();
    var piece = new Piece()
    {
        Foreground = new SolidColorBrush(Colors.Black),
        Stroke = new SolidColorBrush(style),
        Value = value.ToString()
    };
    panel.Children.Add(piece);
}

public void New(StackPanel panel)
{
    panel.Children.Clear();
    panel.CornerRadius = new CornerRadius(10);
    panel.Background = new SolidColorBrush(Colors.WhiteSmoke);
    var numbers = Choose(1, 59, 6);
    numbers.Sort();
    foreach (int number in numbers)
    {
        Add(panel, number);
    }
}
```

The **Method** of **Add** will get a style as a **Color** using the **Dictionary** of **_style** and the **value** passed in then it creates a **Piece** which is a control from **Comentsys.Toolkit.WindowsAppSdk** and this is added to the **StackPanel** that is passed into the **Method**. Then there is the **Method** of **New** which takes a **StackPanel** and sets this up, then some **numbers** are chosen using **Choose** and are sorted with **Sort** then each of these are added to the **StackPanel** using **Add**.

Step 8

Then from **Solution Explorer** for the **Solution** double-click on **MainWindow.xaml** to see the **XAML** for the **Main Window**.
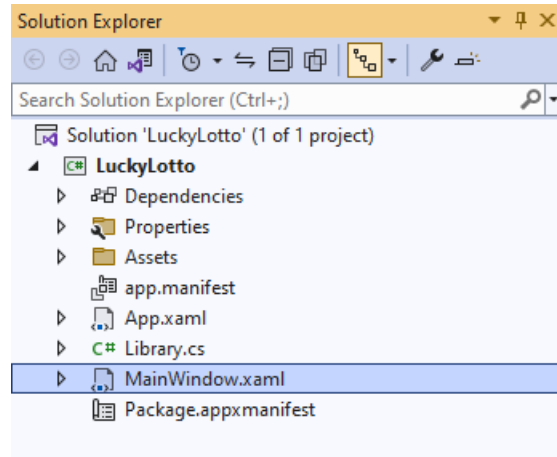
Step 9

In the **XAML** for **MainWindow.xaml** there be some **XAML** for a `StackPanel`, this should be **Removed** by removing the following:

```
<StackPanel Orientation="Horizontal"
HorizontalAlignment="Center" VerticalAlignment="Center">
    <Button x:Name="myButton" Click="myButton_Click">Click Me</Button>
</StackPanel>
```
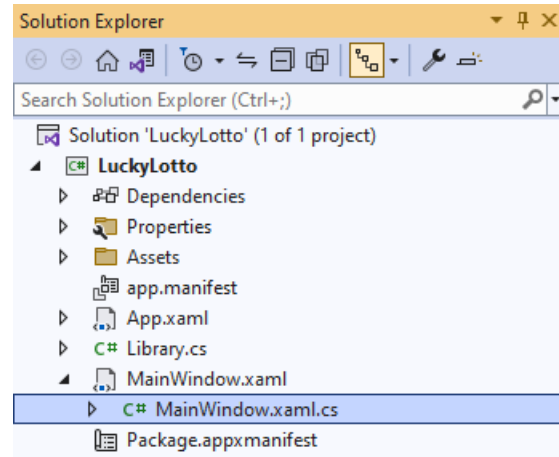
Step 10

While still in the **XAML** for **MainWindow.xaml** above `</Window>`, type in the following **XAML**:

```
<Grid>
    <Viewbox>
        <StackPanel Margin="50" Name="Display" Orientation="Horizontal"
        HorizontalAlignment="Center" VerticalAlignment="Center" Loaded="New"/>
    </Viewbox>
    <CommandBar VerticalAlignment="Bottom">
        <AppBarButton Icon="Page2" Label="New" Click="New"/>
    </CommandBar>
</Grid>
```

This **XAML** contains a `Grid` with a `Viewbox` which will scale a `StackPanel`. It has a `Loaded` event handler for `New` which is also shared by the `AppBarButton`.

Step 11

Then, within **Solution Explorer** for the **Solution** select the arrow next to **MainWindow.xaml** then double-click on **MainWindow.xaml.cs** to see the **Code** for the **Main Window**.

Step 12

In the **Code** for **MainWindow.xaml.cs** there be a **Method** of **myButton_Click(...)** this should be **Removed** by removing the following:

```
private void myButton_Click(object sender, RoutedEventArgs e)
{
    myButton.Content = "Clicked";
}
```

Step 13

Once **myButton_Click(...)** has been removed, type in the following **Code** below the end of the **Constructor** of **public MainWindow() { ... }**:

```
private readonly Library _library = new();

private void New(object sender, RoutedEventArgs e) =>
    _library.New(Display);
```

Here an **Instance** of the **Class** of **Library** is created then below this is the **Method** of **New** that will be used with **Event Handler** from the **XAML**, this **Method** uses Arrow Syntax with the **=>** for an Expression Body which is useful when a **Method** only has one line.
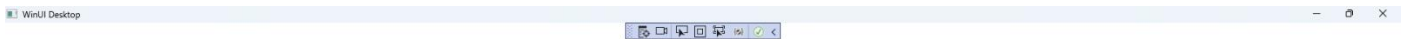
Step 14

That completes the **Windows App SDK** Application. In **Visual Studio 2022** from the **Toolbar** select **LuckyLotto (Package)** to **Start** the Application.

Step 15

Once running you should see the `Piece` elements showing some lottery number and you can select **New** to pick different numbers as many times as needed to get a set of numbers you like.

Step 16

To **Exit** the **Windows App SDK** Application, select the **Close** button from the top right of the Application as that concludes this **Tutorial** for **Windows App SDK** from tutorialr.com!