

# GUI Lab: Agent Assignment 1

## Formål:

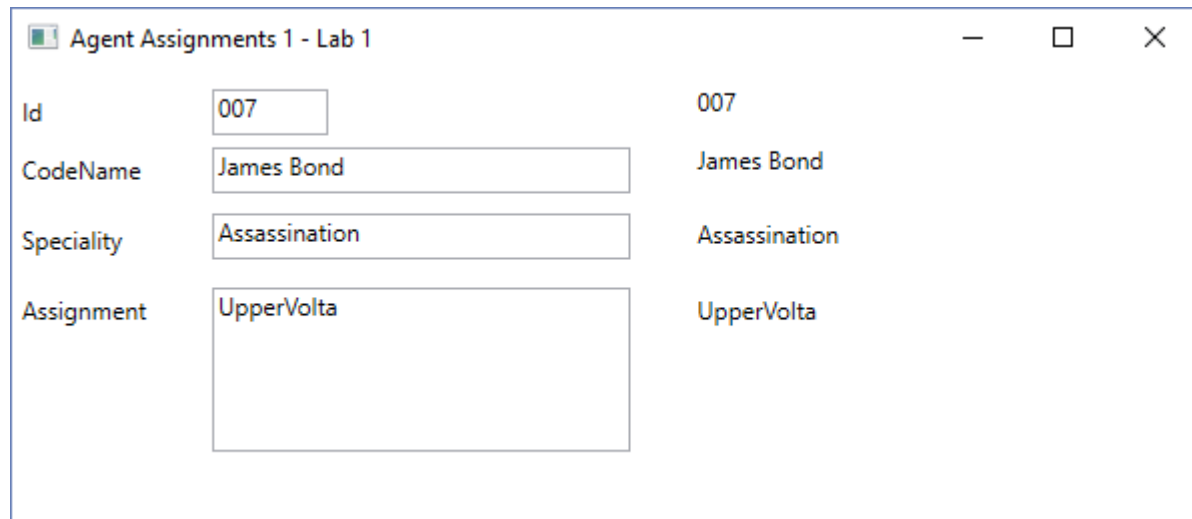
At opnå erfaring med databinding.

## Forudsætninger

At du har læst om data binding.

## Delopgave 1

Lav et WPF program som benytter data binding til at knytte nogle textbokse til attributterne i Agent-klassen (Agent-klassen er implementeret i filen Agent.cs og kan hentes fra Brightspace). For at teste at dataene i objektet virkelig bliver ændret, så tilføjes der også en række tekstblokke som bindes til de samme data.



Id	<input type="text" value="007"/>	007
CodeName	<input type="text" value="James Bond"/>	James Bond
Speciality	<input type="text" value="Assassination"/>	Assassination
Assignment	<input type="text" value="UpperVolta"/>	UpperVolta

## Delopgave 2

Lav et WPF program som benytter data binding til en List data source. I C# koden sættes DataContext til at referere en collection af agenter. Man vælger agent i listboksen. Brug f.eks. DisplayMemberPath til at vise navnet på agenten. Felterne for den valgte agent kan så editeres i tekstboksene til venstre for listboksen.

I denne delopgave skal man ikke kunne tilføje nye agenter, kun ændre properties på dem som listen oprettes med.

The screenshot shows a WPF application window titled "Agent Assignments 1 - Lab 2". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area is divided into two parts. On the left, there is a form with four text boxes, each with a label to its left: "Id" (containing "001"), "CodeName" (containing "Nina"), "Speciality" (containing "Assassination"), and "Assignment" (containing "UpperVolta"). On the right, there is a list box containing two items: "Nina" (which is selected and highlighted) and "James Bond".

## Delopgave 3

Lav et WPF program som benytter data binding til en List data source, således at man kan editere den aktuelle agent, og som indeholder knapper til at navigere gennem listen samt giver mulighed for tilføjelse af nye agenter. Sæt datakonteksten i XAML.

### HJÆLP:

1. Lav en MainWindowViewModel klasse som MainWindows DataContext initieres med i XAML.
2. MainWindowViewModel skal have en property som returnerer listen af agenter.  
Og denne liste skal være af typen ObservableCollection, så brugergrænsefladen automatisk opdateres når der tilføjes nye agenter.
3. ListBoxens ItemSource skal bindes til denne property.
4. Da DataContext nu peger på en ViewModel, så virker databindingen for tekstboksene ikke i den simple udgave, som var ok til lab 1 og 2. For at få dem til at virke igen, så tilføjes en property på MainWindowViewModel med navnet CurrentAgent og typen Agent, som implementerer get og set - og hvor der i set kaldes NotifyPropertyChanged.  
Til at styrer hvem der er CurrentAgent bruges listboksen, så listboksens property SelectedItem skal bindes til CurrentAgent.  
De forskellige TextBokse skal så have opdateret deres bindings, så den for eksempel for Id-tekstboksen komme til at hedde `Path="CurrentAgent.ID"`.
5. Man kan f.eks. navigere igennem listen ved at sætte selectedIndex på listBoxen. Dette gøres i en eventhandler for frem og tilbage knapperne.
6. Eventhandleren for "Add New" skal kalde en metode i MainWindowViewModel som tilføjer en ny agent til listen. Og efterfølgende skal denne nye agent automatisk vælges i listboksen, så brugeren kan indtaste data for den nye agent.

The screenshot shows a WPF application window titled "Agent Assignments 1 - Lab 3". The window is divided into two main sections. On the left, there is a form for editing an agent. It contains four text boxes: "Id" with the value "001", "CodeName" with the value "Nina", "Speciality" with the value "Assassination", and "Assignment" with the value "UpperVolta". Below these text boxes are three buttons: "<", ">", and "Add New". On the right, there is a list box containing two items: "Nina" and "James Bond". The item "Nina" is currently selected, indicated by a blue highlight.