

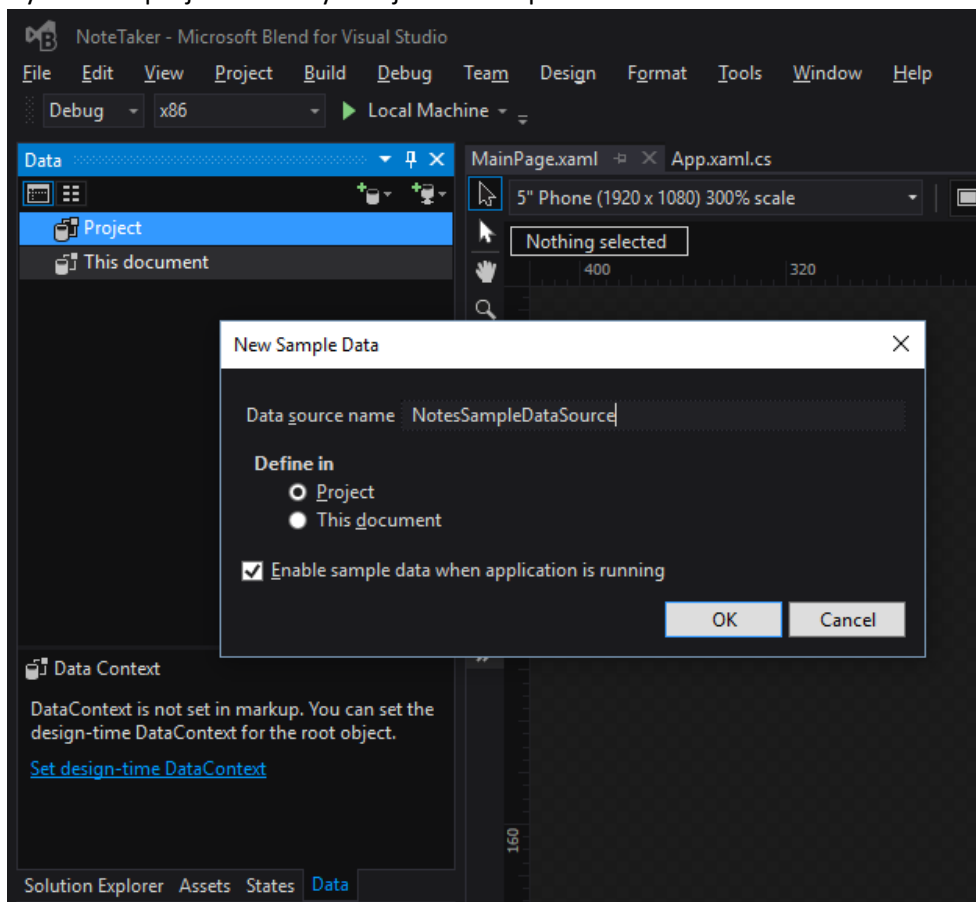
1. Vytvoření kostry projektu

1. Založte nový projekt **Universal Windows Blank App** se jménem **NoteTaker.Win10** a jménem solution **NoteTaker**. Zrušte **Show telemetry in the Windows Dev Center**
2. Cílem je vytvořit aplikaci na správu poznámek. Uživatelské rozhraní aplikace bude na nejvyšší úrovni tvořeno komponentou **SplitView**, která bude zobrazovat:
 - ve svém panelu *seznam poznámek*
 - ve zbytku obrazovky *detail vybrané poznámky*
3. Ponechte v obrazovce existující Grid a do něj přidejte **SplitView**.
4. V rámci SplitView umístěte do jeho části Pane komponentu ListView:

```
<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
  <SplitView>
    <SplitView.Pane>
      <ListView x:Name="NotesListView" />
    </SplitView.Pane>
  </SplitView>
</Grid>
```

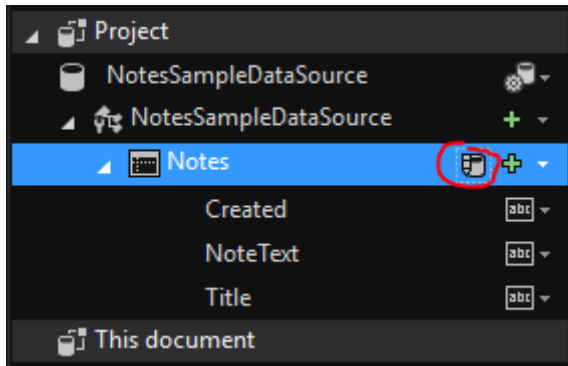
2. Příprava vzorových dat

1. Obrazovku MainPage.xaml otevřete v nástroji **Blend**
2. V panelu data vybereme projekt a dáme New Sample Data
3. Vytvoříte v projektu datový zdroj NotesSampleDataSource

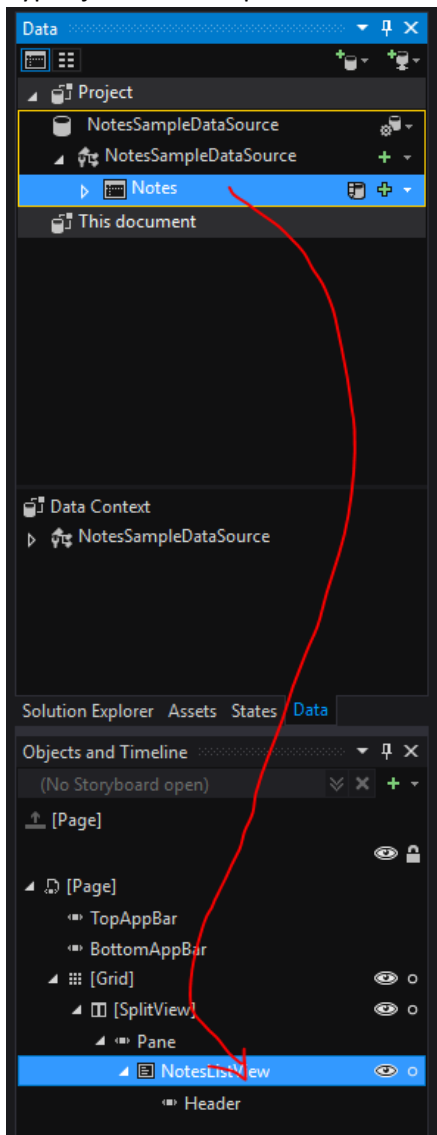


4. Z datového zdroje odebereme „**Groups**“
5. Vytvoříme novou kolekci **Notes** (Add collection property)

- Vytvořte Simple property
 - Created (formát datum)
 - NoteText (text, max. 25 slov)
 - Title (text, max. 8 slov)
- Přes příkaz Edit sample values si zkontrolujte, jaké hodnoty byly vygenerovány



- Máte připravená vzorová data, nyní vytvoříme vazbu mezi daty a ovládacím prvkem vypisujícím seznam položek: Přetáhněte myší kolekci **Notes** na ovládací prvek **NotesListView**



9. Měla by se v designéru ukázat vzorová data.
10. Uložte všechny změny v projektu v Blendu a zavřete jej.
11. Ve Visual Studiu potvrďte, že chcete znovu načíst všechny změněné soubory
12. **Spusťte aplikaci.**

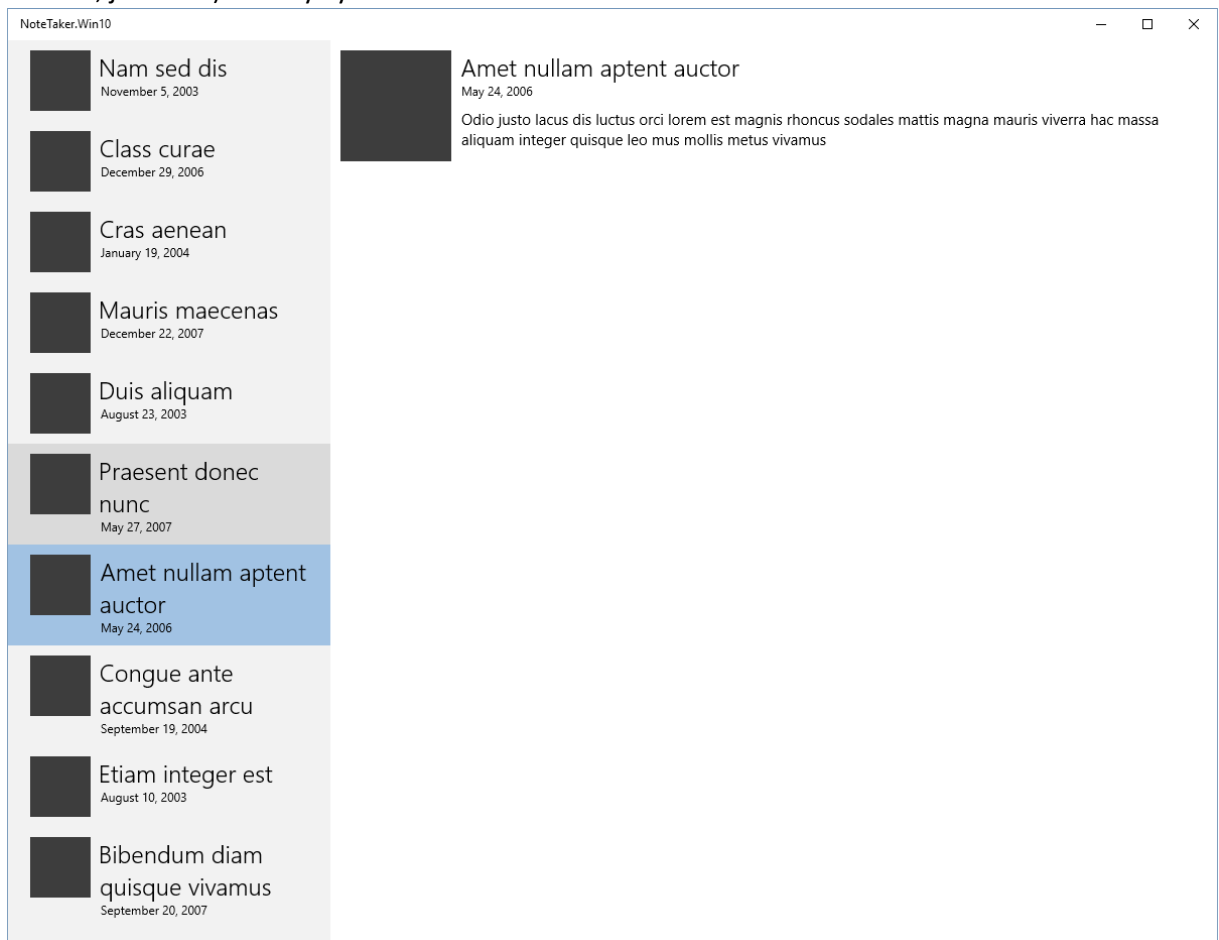
Je vidět prázdné okno protože panel je uzavřený, přidáním vlastnosti `IsPaneOpen="True"` do `SplitView` jej zobrazíte. Přidejte dále vlastnost `DisplayMode="Inline"`.

3. Úprava šablony položky

1. Blend předgeneroval šablonu položky v seznamu, seznamte se s ní.
2. Opravte bindingy tak, aby první řádek textu byla property **Title**, druhý řádek property **Created**.
3. Jako cvičení odstraňte Použitý **Grid a StackPanel** a nahradte jej **RelativePanelem**, aby se ale **nezměnil vzhled UI**. Doporučuji původní Grid pouze zakomentovat a udělat si screenshot obrazovky pro porovnání.
4. Zmenšete velikost obrázku na 60x60px a odstraňte pevnou výšku položky. Vyzkoušejte aktivovat zalamování textů.

4. Zobrazení detailu poznámky

1. Implementujte zobrazení detailu poznámky do pravé části okna aplikace.
2. Do `SplitView` umístěte `ScrollView` s `RelativePanelem` a v něm sestavte detail poznámky (bez editace, jen čtení). Možný výsledek:



3. Aby se zobrazoval detail vybrané položky, bude třeba vybranou položku nastavit jako datový kontext RelativePanelu, který tvoří detail poznámky. Toho docílíte nastavením property DataContext RelativePanelu: `DataContext="{Binding SelectedItem, ElementName=NotesListView}"`
4. Pro zobrazení klasických textů u vybrané položky, použijte klasické bindingy, např.: `Text="{Binding Title}"`
5. Pokud není vybrána žádná položka, zařídte ať je skrytý rámeček na obrázek. Za tímto účelem využijeme Converter. V projektu založte složku Converters a vytvořte v ní **veřejnou třídu NullToVisibilityConverter implementující rozhraní IValueConverter**
6. Pokud je vstupní hodnota value metody Convert null, potom vračejte `Windows.UI.Xaml.Visibility.Collapsed` v opačném případě `Visible`.
7. Instanci converteru vytvoříme jako aplikační resource. V souboru App.xaml přidejte using namespace do hlavičky: `xmlns:converters="using:NoteTaker.Win10.Converters"` a poté i celý klíč s danou resource:


```
<converters:NullToVisibilityConverter
x:Key="NullToVisibilityConverter"></converters:NullToVisibilityConverter>
```
8. Do ScrollViewu přidejte nastavení property Visibility s využitím converteru: `Visibility="{Binding SelectedItem, ElementName=NotesListView, Converter={StaticResource NullToVisibilityConverter}}"`
9. Vyzkoušejte, že zmizel šedý rámeček, pokud není vybrána žádná položka a při vybrání položky se stále zobrazí detail 😊

5. Implementace přizpůsobení se velikosti obrazovky

1. Zkuste si zmenšit velikost okna aplikace a uvidíte, že aplikace se stává nepoužitelnou, protože UI se nijak nepřizpůsobí změně velikosti okna. Stejně tak můžete aplikaci zkusit nasadit na mobilní telefon či do emulátoru.
2. Navrhněte, jak by se mohlo uživatelské rozhraní měnit v závislosti na velikosti obrazovky, a tyto změny popište pomocí vizuálních stavů. Ukázka takového popisu je uvedena níže:

```
<VisualStateManager.VisualStateGroups>
  <VisualStateGroup>
    <VisualState>
      <!--VisualState to be triggered when window width is >=720 effective pixels-->
      <VisualState.StateTriggers>
        <AdaptiveTrigger MinWindowWidth="720" />
      </VisualState.StateTriggers>
      <VisualState.Setters>
        <Setter Target="MainSplitView.DisplayMode" Value="CompactInline"></Setter>
        <Setter Target="MainSplitView.IsPaneOpen" Value="True" />
        <Setter Target="itemHeader.(RelativePanel.RightOf)" Value="border"></Setter>
      </VisualState.Setters>
    </VisualState>
  </VisualStateGroup>
</VisualStateManager.VisualStateGroups>
```

3. Vyzkoušejte tyto změny stavů UI naimplementovat.

6. Implementace vrstvy model

1. Vytvořte v projektu složku **Models**
2. V této složce implementujte třídu Note, která bude ukládat všechna data o poznámce:

```
public class Note
{
  public int NoteID { get; set; }
  public string Title { get; set; }
}
```

```

    public string NoteText { get; set; }
    public DateTime Created { get; set; }
}

```

3. V modelu dále implementujte rozhraní, které bude předepisovat operace nad úložištěm poznámek:

```

public interface INotesManager
{
    System.Collections.Generic.List<Note> GetNotes();
    void InsertNote(Note note);
    void UpdateNote(Note note);
    void DeleteNote(Note note);
}

```

4. Implementujte třídu `public class TestNotesManager : INotesManager`, která bude vracet statický seznam poznámek, jako vzorová data pro testování aplikace

7. Implementace vrstvy View-Model

1. Vytvořte v projektu složku **ViewModels**
2. Pro zjednodušení práce s implementací `INotifyPropertyChanged` vytvoříme bázovou třídu, která bude obsahovat pomocný kód:

```

public abstract class ViewModelBase : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;
    protected void NotifyPropertyChanged([CallerMemberName]string propertyName =
    "")
    {
        if (PropertyChanged != null)
        {
            PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
        }
    }
}

```

3. Implementujte třídu **NoteViewModel**, která bude dědit z `ViewModelBase` a bude mít následující vlastnosti:

- a. V konstruktoru přijme instanci `Note`
- b. Pomocí veřejných property zpřístupní property třídy `Note`
- c. Pokud dojde ke změně property, vytvoří notifikaci:

```

public string Title
{
    get
    {
        return note.Title;
    }
    set
    {
        if (note.Title != value)
        {
            note.Title = value;
            NotifyPropertyChanged();
        }
    }
}

```

4. Implementujte třídu `MainPageViewModel`, která bude dědit z `ViewModelBase` a bude mít následující vlastnosti:

- a. V konstruktoru přijme instanci `INotesManager` a uloží si ji.
- b. Bude mít property implementující `INotifyPropertyChanged`:

- i. `public ObservableCollection<NoteViewModel> Notes`
 - ii. `public NoteViewModel SelectedNote`
- c. Bude mít metodu `public void LoadData()`, která načte data z `INotesManageru`.

7. Propojení View a View-Modelu

1. V `MainPage.xaml` odstraňte `NotesSampleDataSource` jako hlavní zdroj dat, ponechte jej jen pro design.
2. V třídě `MainPage` vytvořte private field uchovávající ViewModel stránky `MainPageViewModel viewModel`
3. V konstruktoru stránky vytvořte instanci ViewModelu a přiřadte jako `DataContext` celé obrazovky (`this.DataContext = viewModel;`)
4. Při navigaci na stránku zavolejte načtení dat nad ViewModelem:

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    base.OnNavigatedFrom(e);
    viewModel.SaveChangedNotes();
}
```

5. Zajistěte, aby vybraná položka v seznamu poznámek byla načtena do property `SelectedNote` ViewModelu: `SelectedItem="{Binding SelectedNote, Mode=TwoWay}"`
6. Detail poznámky nenačítejte podle `ListView`, ale z property `SelectedNote`