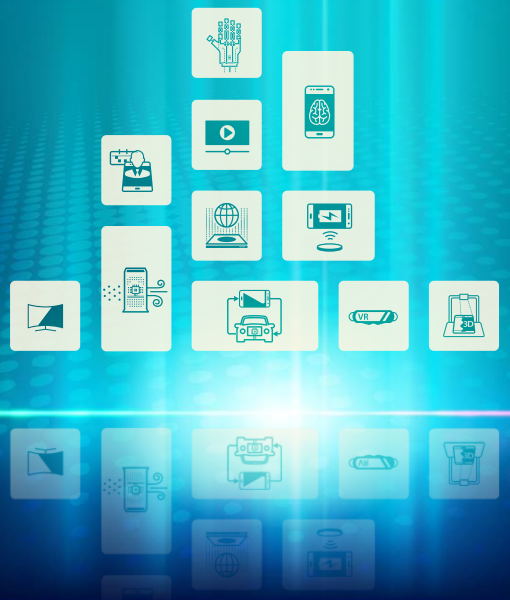
SAMSUNG

Szkolenie Xamarin 4 / 4

Wprowadzenie komponentu ListView

Adam Szczerbiak

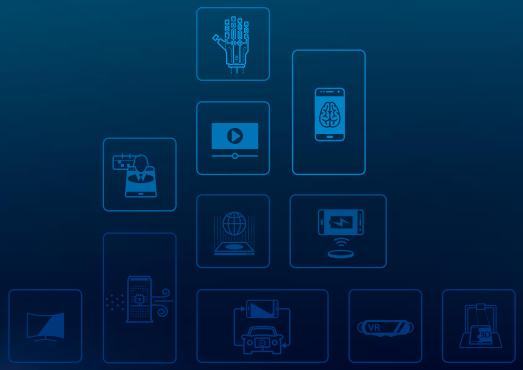


Apr 25, 2017 | Samsung R&D Institute Poland

Kim jesteśmy?

Robert

Adam



Plan warsztatów

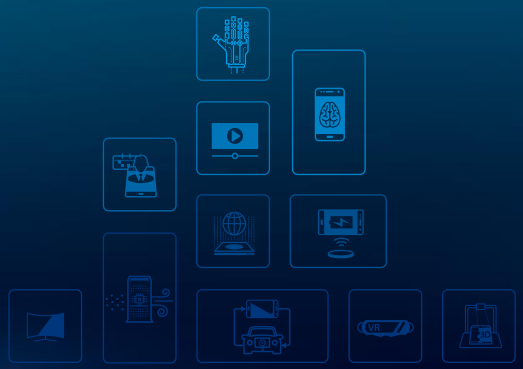
I. Wprowadzenie do Xamarin
Pierwsza aplikacja i XAML

II. Układy stron i nawigacja

III. Układ siatkowy - Grid

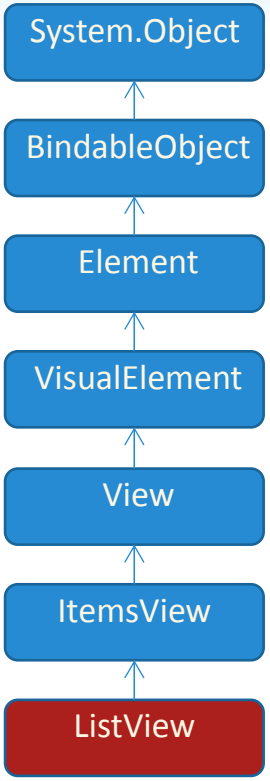
IV. ListView

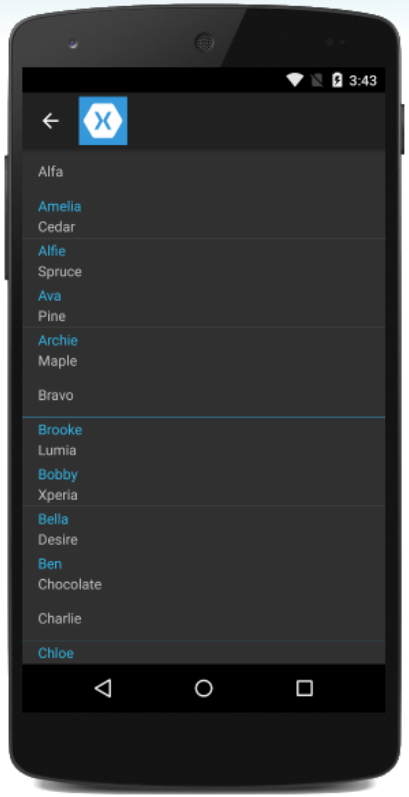
Wstęp

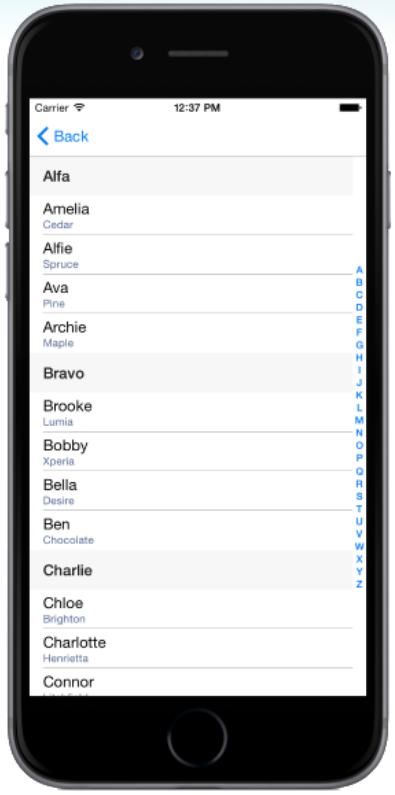


Klasa ListView – hierarchia

Położenie komponentu ListView w hierarchii klas widgetów biblioteki Xamarin.Forms:



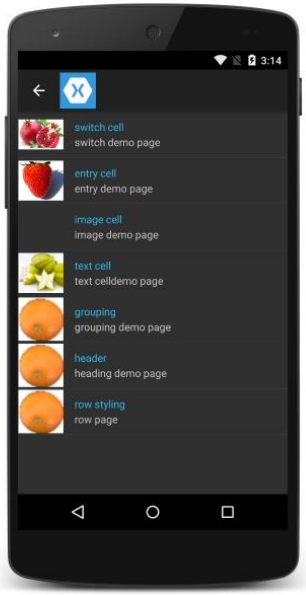


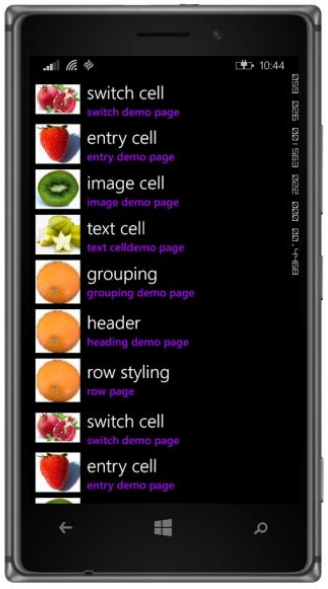


Źródło: https://developer.xamarin.com/guides/xamarin-forms/user-interface/listview/customizing-list-appearance/Images/grouping_depth.png (detale)

ListView – podstawy

- **Wyświetlanie kolekcji elementów**
  (tego samego typu, np. tekst, obraz, przełącznik)
- **Lista jest zawsze wertykalna (pionowa)**
- **W razie potrzeby automatycznie dodawane jest przewijanie**
  (programista nie musi manualnie dodawać komponentu ScrollView)
- **Możliwość zaznaczania jednego bądź wielu elementów**

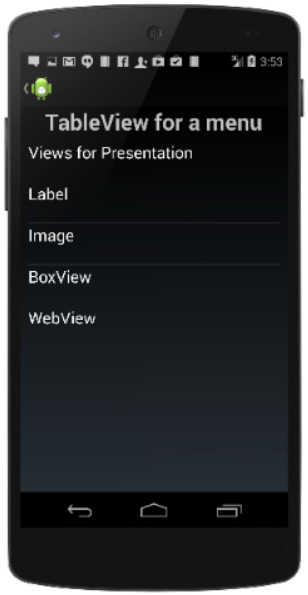




Źródło:
https://developer.xamarin.com/guides/xamarin-forms/user-interface/listview/Images/image_cell_default.png

ListView – ograniczenia

- Wyświetlanie przewijalnych list z danymi
- Źródło danych powinno być jednorodne
  (wszystkie dane muszą być tego samego typu)
- ListView czy TableView?

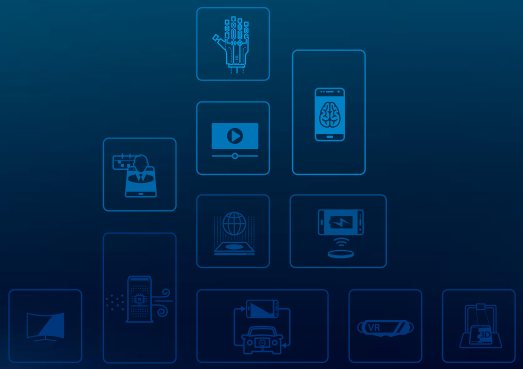


Źródło:
https://developer.xamarin.com/api/resource?id=source-id:10:TableView.TripleScreenShot.png

ListView – konspekt

- **Źródło danych: w jaki sposób przekazać dane do listy?**
  (pole ItemsSource – kontrola manualna bądź za pomocą *data binding* zgodnie z MVVM)
- **Kontrola nad wyglądem komórek**
  (komórki wbudowane; komórki niestandardowe)
- **Dostosowanie wyglądu listy**
  (Ustawianie nagłówków i stopek, tworzenie grup komórek, kontrola nad wysokością elementów)
- **Obsługa zdarzeń i interaktywność**
  (Obsługa dotknięć, akcji kontekstowych, implementacja pull-to-refresh)
- **Typowe pułapki**
  (jakich błędów nie popełniać)

ListView
Źródło danych



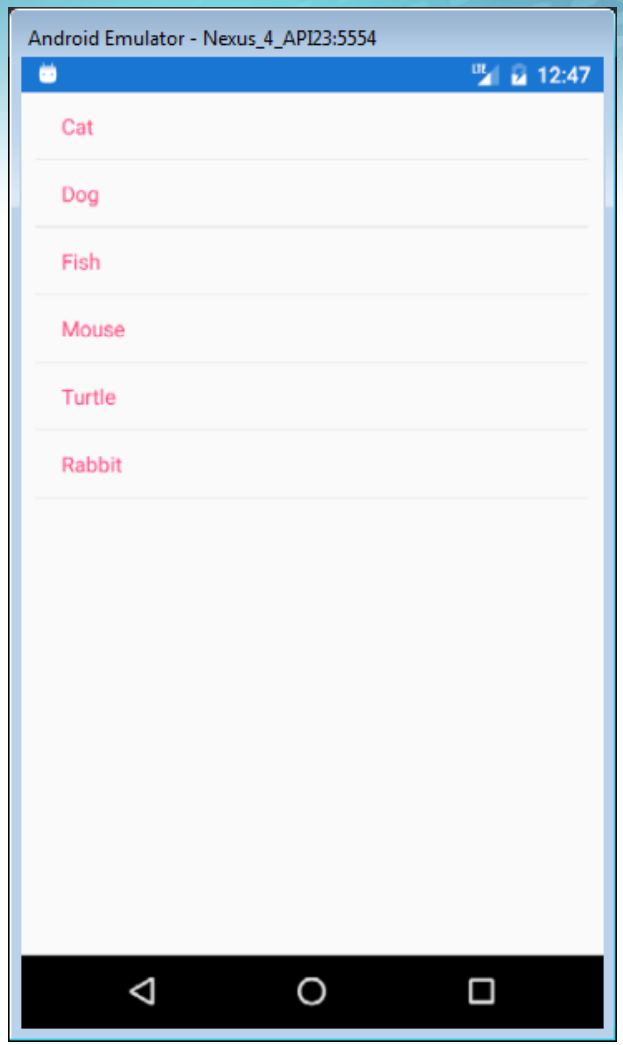
Populacja listy danymi

- **Metoda manualna:**

  wykorzystanie pola ItemsSource

C#

```
var listView = new ListView();
listView.ItemsSource = new string[] {
    "Cat",
    "Dog",
    "Fish",
    "Mouse",
    "Turtle",
    "Rabbit"
};
```



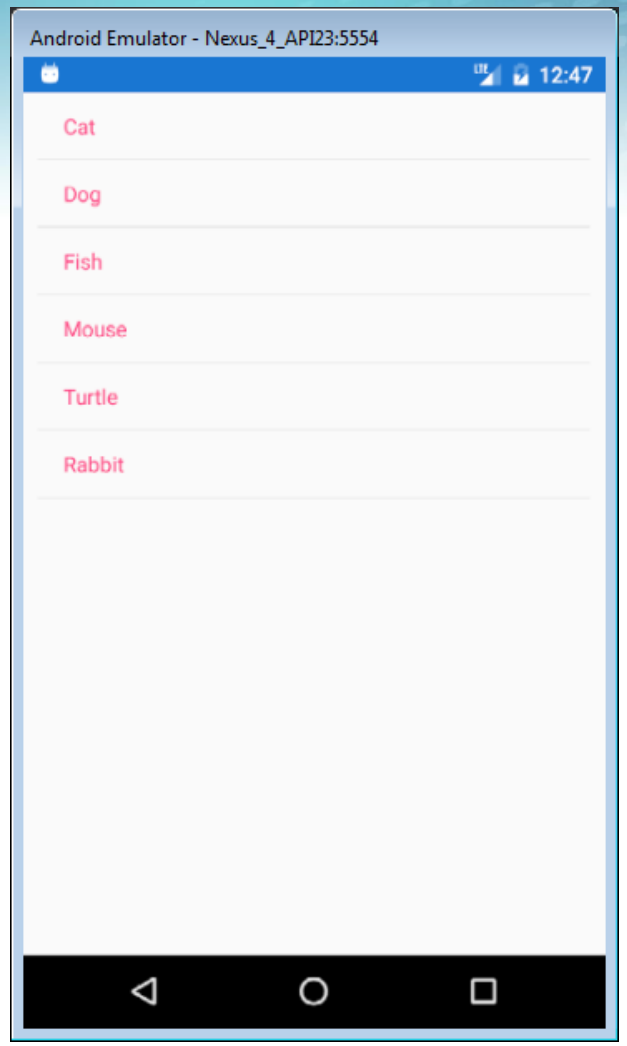
Populacja listy danymi

- **Metoda manualna:**

  wykorzystanie pola ItemsSource

XAML

```
<ListView x:Name="listView">
    <ListView.ItemsSource>
        <x:Array Type="{x:Type x:String}">
            <x:String>Cat</x:String>
            <x:String>Dog</x:String>
            <x:String>Fish</x:String>
            <x:String>Mouse</x:String>
            <x:String>Turtle</x:String>
            <x:String>Rabbit</x:String>
        </x:Array>
    </ListView.ItemsSource>
</ListView>
```



Wiązanie danych – przykład

XAML

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:XamLV"
             x:Class="XamLV.MainPage">

    <ContentPage.Padding>
        <OnPlatform x:TypeArguments="Thickness"
                    iOS="10, 20, 10, 0"
                    Android="10, 0"
                    WinPhone="10, 0" />
    </ContentPage.Padding>

    <StackLayout>

    </StackLayout>

</ContentPage>
```

Wypełnienie StackLayout na następnym slajdzie

Wiązanie danych – przykład

XAML

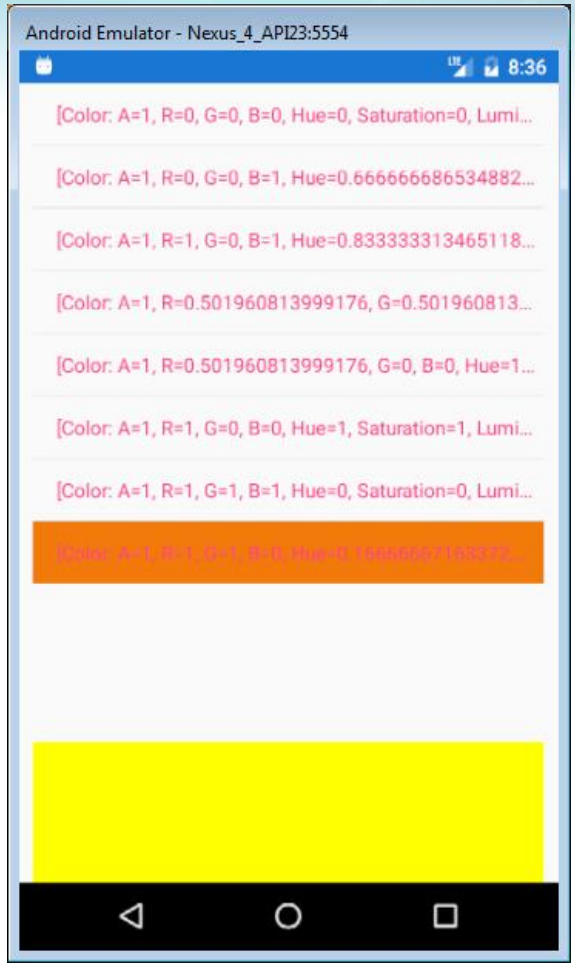
```
<StackLayout>
    <ListView x:Name="listView"
              SelectedItem="{Binding Source={x:Reference boxView},
                                     Path=Color,
                                     Mode=TwoWay}">
        <ListView.ItemsSource>
            <x:Array Type="{x:Type Color}">
                <Color>Black</Color>
                <Color>Blue</Color>
                <Color>Fuchsia</Color>
                <Color>Gray</Color>
                <Color>Maroon</Color>
                <Color>Red</Color>
                <Color>White</Color>
                <Color>Yellow</Color>
            </x:Array>
        </ListView.ItemsSource>
    </ListView>

    <BoxView x:Name="boxView"
             Color="Yellow"
             HeightRequest="100" />
</StackLayout>
```

Wiązanie dwukierunkowe (ang. *two-way data binding*)

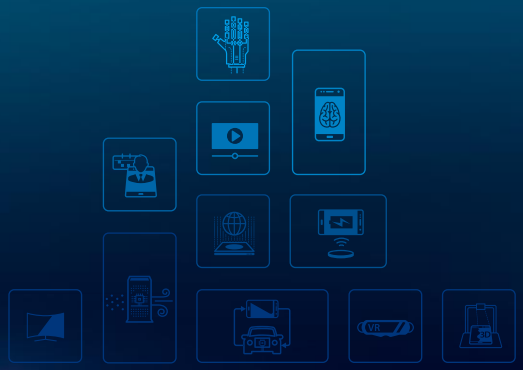
Uwaga: wartość początkowa powinna występować w tablicy powyżej!

Wiązanie danych – przykład

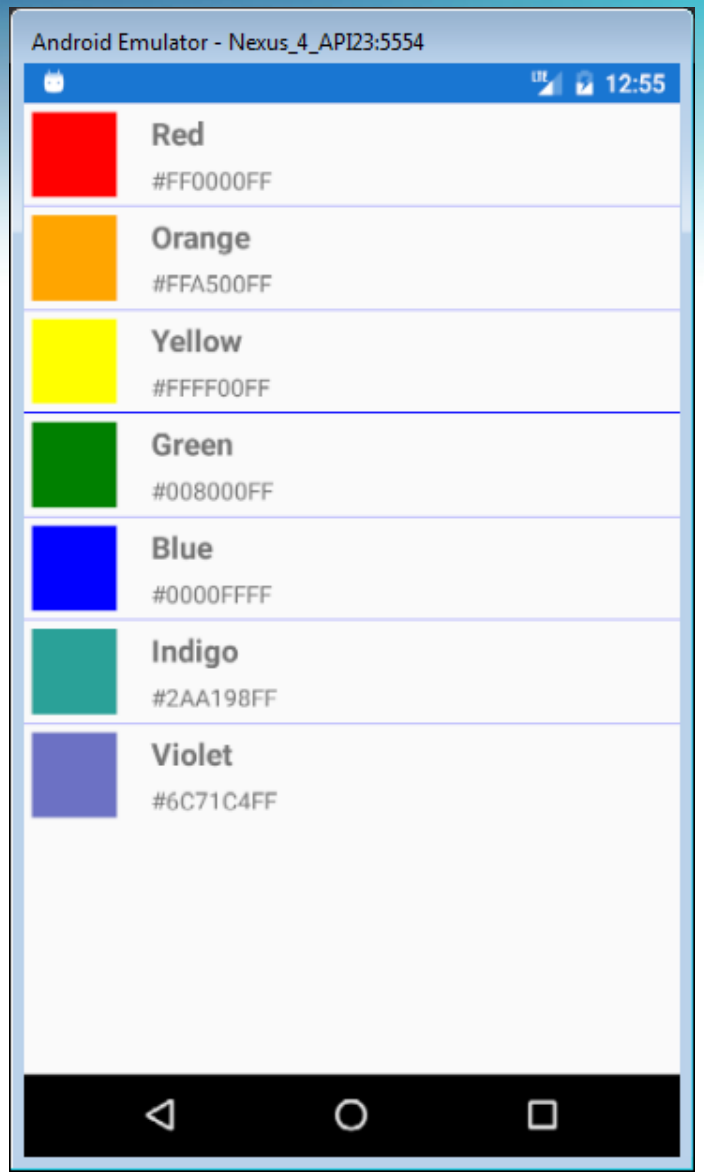
Rezultat:



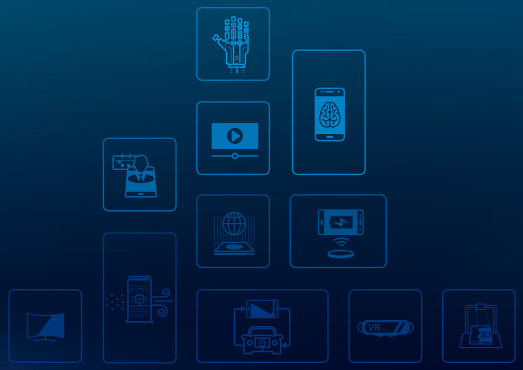
Sesja na żywo



Aplikacja do stworzenia



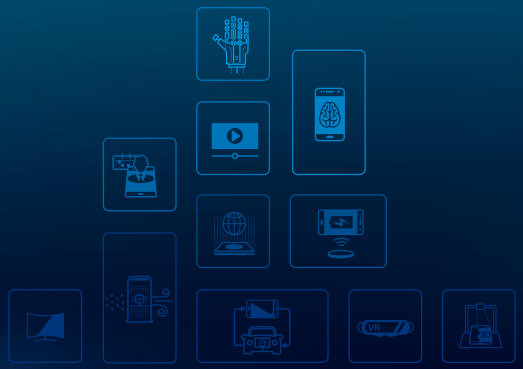
ListView
Interaktywność



Interaktywność ListView

- **Charles Petzold, "Creating Mobile Apps With Xamarin.Forms"**
  (Rozdział 19, "Collection views", punkt "ListView and Interactivity", str. 570-574)

ListView
Typowe pułapki



Modyfikacja źródła danych

C#

```
var listView = new ListView();
var data = new List<string> {
    "Cat",
    "Dog",
    "Fish",
    "Mouse",
    "Turtle",
    "Rabbit"
};

listView.ItemsSource = data;

Device.StartTimer(TimeSpan.FromSeconds(1), () =>
{
    data.Add("Spider"); // Czy to zadziała?
    return true;
});
```

Kontrolka ListView nie jest synchronizowana ze źródłem danych

Modyfikacja źródła danych

Modyfikacje dokonane na źródle po populacji listy są ignorowane.

Rozwiązanie problemu: użyj ObservableCollection

ObservableCollection przypomina standardową listę,
ma jednak możliwość poinformowania ListView o swoich zmianach

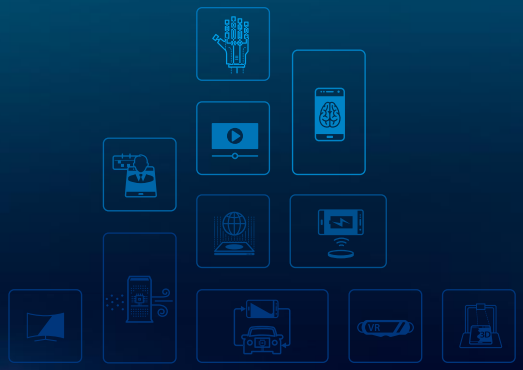
C#

```
var listView = new ListView();
ObservableCollection<string> animalsList = new ObservableCollection<string>();
listView.ItemsSource = animalsList;

Device.StartTimer(TimeSpan.FromSeconds(1), () =>
{
    oc.Add("Spider"); // Czy to zadziała?
    return true;
});
```

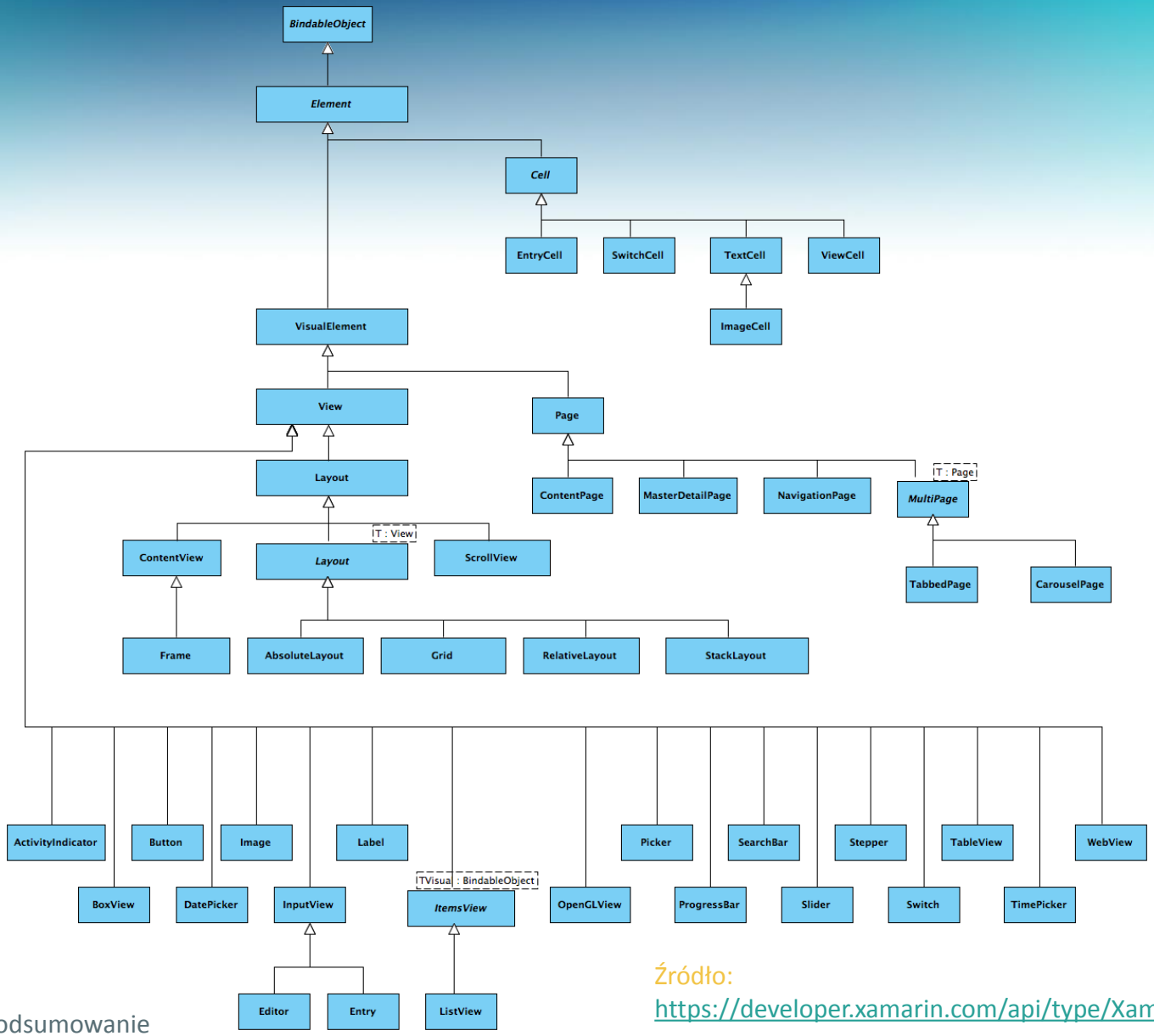
```
// TAK, Spider zostanie dodany do listy.
// ObservableCollection poinformuje listView o zmianach
// za pomocą zdarzenia CollectionChanged
// z interfejsu INotifyCollectionChanged.
```

Odwiedź:
http://www.codeproject.com/Articles/42536/List-vs-ObservableCollection-vs-INotifyPropertyCha
aby przeczytać krótki samouczek dotyczący ObservableCollection.

Podsumowanie



Pełna hierarchia klas Xamarin.Forms



Źródło:
https://developer.xamarin.com/api/type/Xamarin.Forms.Element/

SAMSUNG

Dziękujemy!