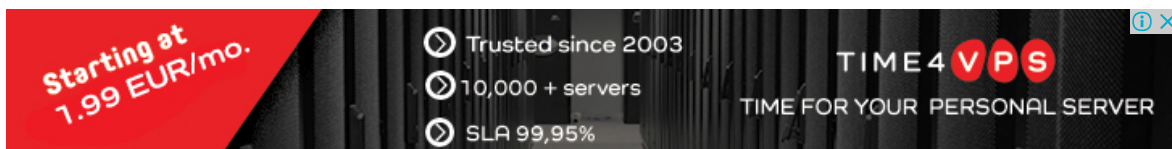




Группировка в ListView

Последнее обновление: 29.03.2017



Элемент ListView в Xamarin поддерживает возможности группировки. Рассмотрим, как мы можем сгруппировать элементы в списке

Для группировки в начале добавим в проект вспомогательный класс, который назовем Grouping:

```
using System.Collections.Generic;
using System.Collections.ObjectModel;

namespace HelloApp
{
    public class Grouping<K, T> : ObservableCollection<T>
    {
        public K Name { get; private set; }
        public Grouping(K name, IEnumerable<T> items)
        {
            Name = name;
            foreach (T item in items)
                Items.Add(item);
        }
    }
}
```

Класс Grouping типизирован двумя параметрами. Параметр K представляет тип ключа группы, который будет храниться в свойстве Name. А параметр T представляет тип объектов, которые будут храниться в коллекции Items. Это свойство-коллекция унаследовано от базового класса ObservableCollection. А в конструкторе мы получаем все необходимые данные.

В качестве объектов возьмем опять же класс Phone:

```
public class Phone
{
    public string Title { get; set; }
    public string Company { get; set; }
    public int Price { get; set; }
}
```

В коде страницы MainPage создадим список групп:

```
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;
using Xamarin.Forms;

namespace HelloApp
{
    public partial class MainPage : ContentPage
    {
        // список групп, к которым идет привязка
        public ObservableCollection<Grouping<string, Phone>> PhoneGroups { get; set; }
        public MainPage()
        {
            InitializeComponent();

            // начальные данные
            var phones = new List<Phone>
            {
                new Phone {Title="Galaxy S8", Company="Samsung", Price=60000 },
            }
        }
    }
}
```

```

new Phone {Title="Galaxy S7 Edge", Company="Samsung", Price=50000 },
new Phone {Title="Huawei P10", Company="Huawei", Price=10000 },
new Phone {Title="Huawei Mate 8", Company="Huawei", Price=29000 },
new Phone {Title="Mi6", Company="Xiaomi", Price=55000 },
new Phone {Title="iPhone 7", Company="Apple", Price=38000 },
new Phone {Title="iPhone 6S", Company="Apple", Price=50000 }
};
// получаем группы
var groups = phones.GroupBy(p => p.Company).Select(g => new Grouping<string, Phone>(g.Key, g));
// передаем группы в PhoneGroups
PhoneGroups = new ObservableCollection<Grouping<string, Phone>>(groups);
this.BindingContext = this;
}
}
}
}

```

В конструкторе переменная `phones` определяет общие данные, по которым создается коллекция групп в виде свойства `PhoneGroups`. Группировка в данном случае идет по свойству `Company` объекта `Phone`.

А в коде `xaml` у `MainPage` пропишем выражения привязки:

```

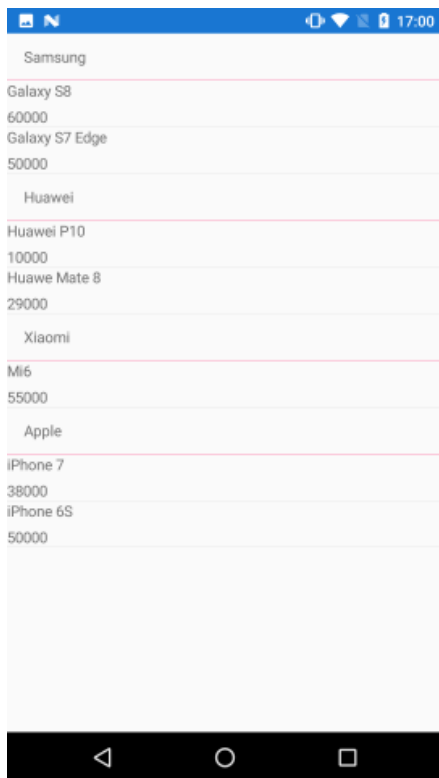
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  xmlns:local="clr-namespace:HelloApp;assembly=HelloApp"
  x:Class="HelloApp.MainPage">
  <StackLayout>
    <ListView x:Name="phonesList"
      HasUnevenRows="True"
      GroupDisplayBinding="{Binding Name}"
      ItemsSource="{Binding PhoneGroups}"
      IsGroupingEnabled="True">
      <ListView.ItemTemplate>
        <DataTemplate>
          <ViewCell>
            <StackLayout>
              <Label Text="{Binding Title}" />
              <Label Text="{Binding Price}" />
            </StackLayout>
          </ViewCell>
        </DataTemplate>
      </ListView.ItemTemplate>
    </ListView>
  </StackLayout>
</ContentPage>

```

Привязка `ListView` здесь идет к свойству `PhoneGroups`, которое содержит группы. Установка свойства `IsGroupingEnabled="True"` добавляет в `ListView` поддержку групп.

С помощью свойства **`GroupDisplayBinding`** можно задать то значение, которое будет отображаться для каждой группы. В нашем случае идет привязка к имени группы, которое представляет критерий группировки.

И после запуска приложения все данные в списке будут сгруппированы по компаниям:



Однако по умолчанию заголовки групп выглядят не очень хорошо, мало отличимы от основного содержимого элементов. И в этом случае мы можем настроить шаблон отображения заголовков групп. Для этого изменим разметку 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:HelloApp;assembly=HelloApp"
  x:Class="HelloApp.MainPage">
  <StackLayout>
    <ListView x:Name="phonesList"
      HasUnevenRows="True"
      ItemsSource="{Binding PhoneGroups}"
      IsGroupingEnabled="True">
      <ListView.GroupHeaderTemplate>
        <DataTemplate>
          <ViewCell Height="30">
            <StackLayout>
              <Label Text="{Binding Name}" FontSize="Large" />
            </StackLayout>
          </ViewCell>
        </DataTemplate>
      </ListView.GroupHeaderTemplate>
      <ListView.ItemTemplate>
        <DataTemplate>
          <ViewCell>
            <StackLayout>
              <Label Text="{Binding Title}" />
              <Label Text="{Binding Price}" />
            </StackLayout>
          </ViewCell>
        </DataTemplate>
      </ListView.ItemTemplate>
    </ListView>
  </StackLayout>
</ContentPage>
```

Свойство **GroupHeaderTemplate** позволяет в корне изменить отображение заголовка, определив ему свой шаблон DataTemplate:



Яндекс.Директ

[Назад](#) [Содержание](#) [Вперед](#)


G+



9 Комментариев metanit.com

1 Войти ▾

[❤ Рекомендовать](#) [🔗 Поделиться](#)

Лучшее в начале ▾



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS (?)

Имя

**Андрей Лысенко** • месяц назад

Вопрос конечно не по рассматриваемой теме, а по коллекциям

// получаем группы

```
var groups = phones.GroupBy(p => p.Company).Select(g => new Grouping<string, phone>{Key = g.Key, g});
```

// передаем группы в PhoneGroups

```
PhoneGroups = new ObservableCollection<grouping<string, phone>>(groups);
```

здесь в качестве второго аргумента когда создается new Grouping<string, phone>{Key = g.Key, g}, почему вторым аргументом конструктор допускает передачу не только IEnumerable<phone>, но и как в нашем случае вместо g подставляется IGrouping<string, phone>? ведь наш конструктор Grouping явно дает понять что принимает только IEnumerable<phone>, я понимаю что здесь явно пробел моих знаний, но никак не могу понять где этот пробел

^ | v • Ответить • Поделиться ›

**Andrey Lysenko** → Андрей Лысенко • 17 дней назад

Отвечать не надо. Догнал с повторного прочтения статьи, не вдумался тогда в метод GroupBy(....)

^ | v • Ответить • Поделиться ›

**Anatoliy Tkachenko** • 4 месяца назад

Подскажите, а при добавлении нового элемента в ListView необходимо заново получать группы, передавать группы в PhoneGroups, а затем передавать в BindingContext?

т.е. следующий код дублируется?

// получаем группы

```
var groups = phones.GroupBy(p => p.Company).Select(g => new Grouping<string, phone>{Key = g.Key, g});
```

// передаем группы в PhoneGroups

```
PhoneGroups = new ObservableCollection<grouping<string, phone>>(groups);
```

```
this.BindingContext = this;
```

^ | v • Ответить • Поделиться ›

**Anatoliy Tkachenko** → Anatoliy Tkachenko • 4 месяца назад

Подскажите, пожалуйста, есть какие либо еще варианты?

^ | v • Ответить • Поделиться ›

**Monte Carlo** • 7 месяцев назад

привет, нужна помощь. У меня больше 30 итемов в листвю не отображается. как это исправить? Буду очень благодарен за помощь!

^ | v • Ответить • Поделиться ›

**Metanit** Модератор → Monte Carlo • 7 месяцев назад

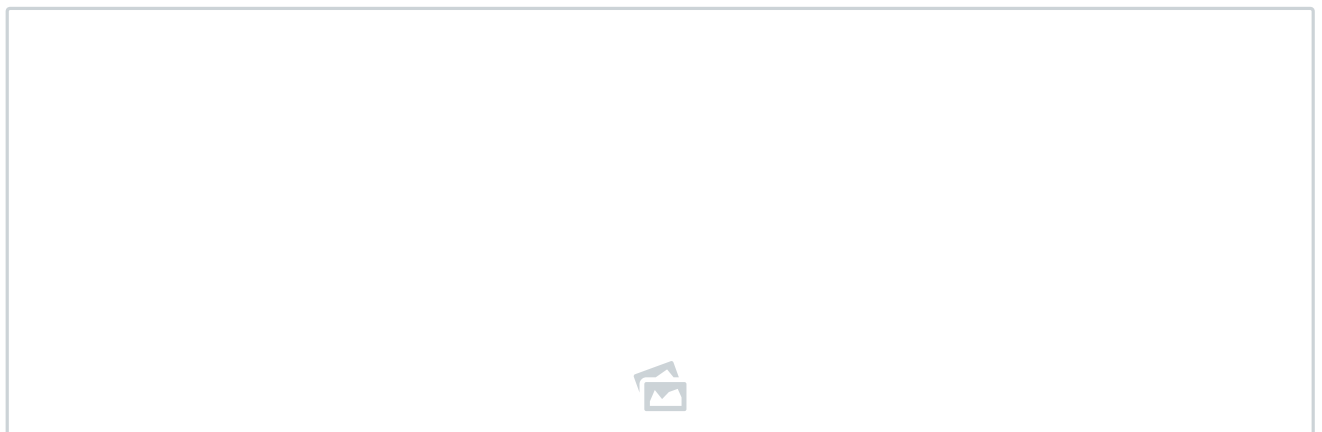
не отображаются вообще даже при прокрутке?

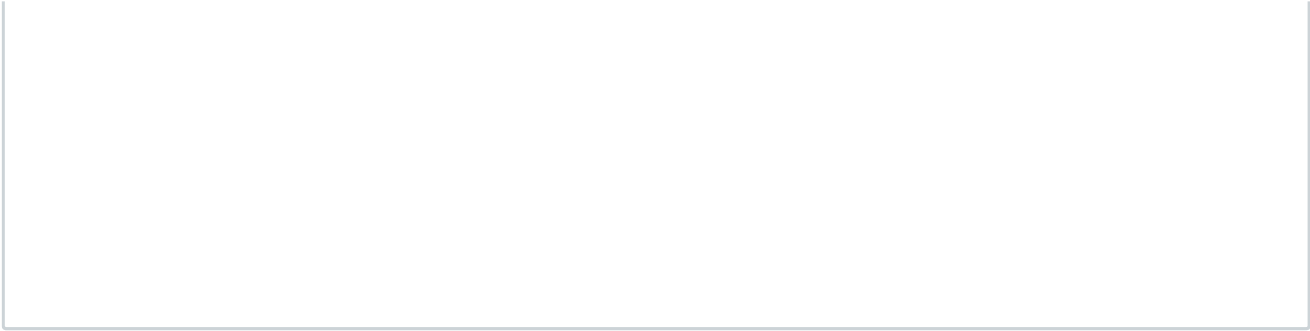
^ | v • Ответить • Поделиться ›

**Monte Carlo** → Metanit • 7 месяцев назад

есть скролвю, а в нем лист вю. И скролится только 30 элементов, когда в коллекции есть намного больше(50-80). Остальные элементы не помещаются в этот лист, отображаются только первых 30

^ | v • Ответить • Поделиться ›

**Александр Рыбак** • 7 месяцев назад



На устройствах Windows заголовки группы отображаются большим шрифтом без дополнительной настройки заголовка.

^ | v • Ответить • Поделиться ›



Aleksei • 8 месяцев назад


А как сворачивать группы?

^ | v • Ответить • Поделиться ›

ТАКЖЕ НА METANIT.COM


C# и .NET | Раннее и позднее связывание

4 комментариев • 2 месяца назад

 **dev loop** — спасибо, ответили и на мой вопрос тоже)
Аватар


Vue.js | Локальные и глобальные фильтры

1 комментарий • 3 месяца назад

 **eugene81** — Можно ли создать фильтр для фильтрации массивов?
Аватар


Go | Соответствие интерфейсу

1 комментарий • месяц назад

 **Пу** — Еще давай. Не останавливайся, пожалуйста. Это гениальный язык. Он, конечно, не божественен, как шарп, но гениальный.

C++ | Динамические массивы

4 комментариев • 3 месяца назад

 **Metanit** — в данном случае вы делаете ошибку, что переносите опыт работы с одним языком на другой. В разных языках может быть разная терминология. В C++