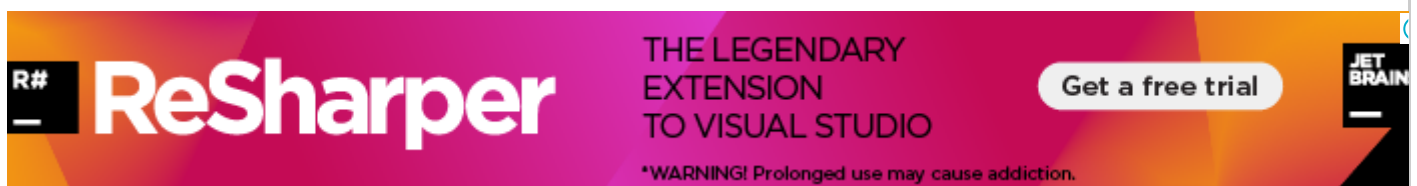




# Визуальные компоненты ContentView

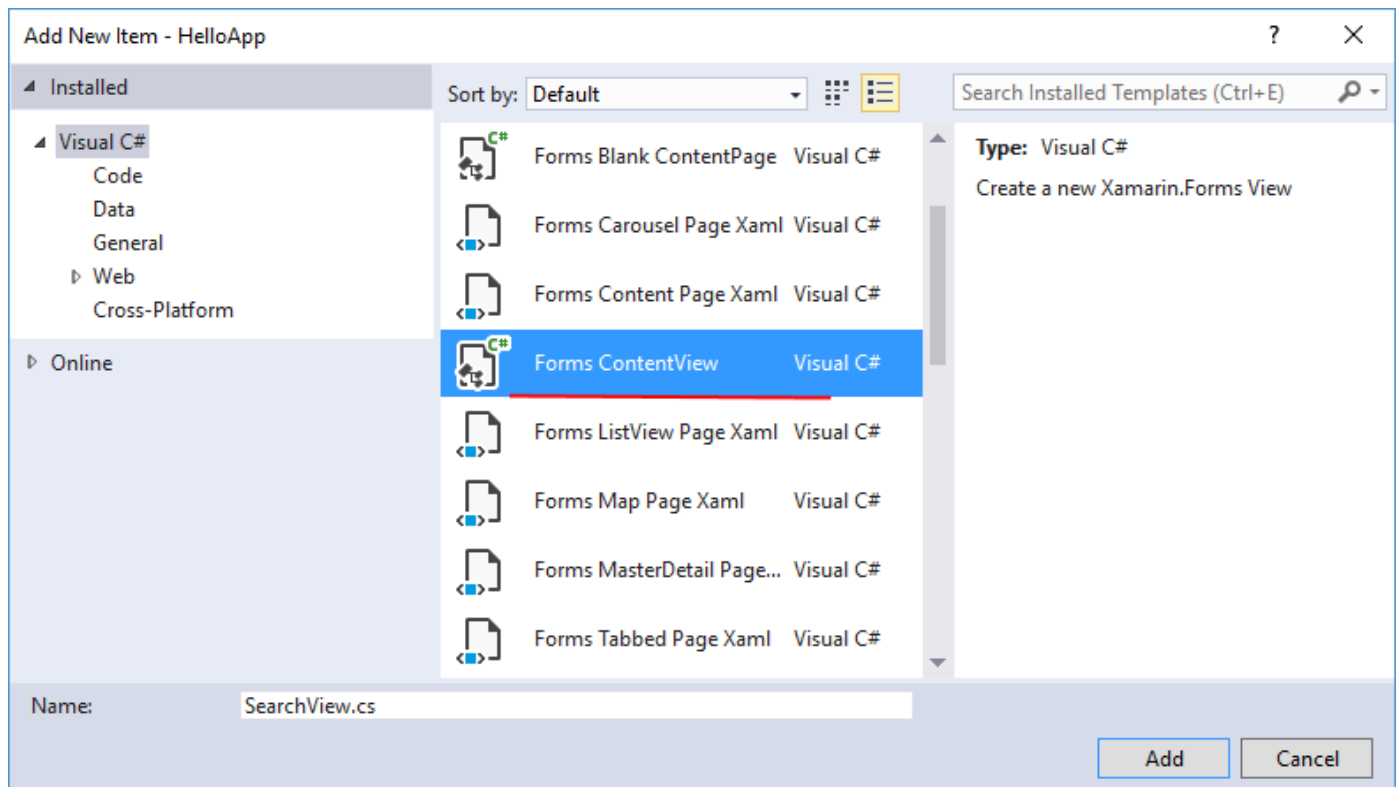
## Создание визуальных компонентов

Последнее обновление: 17.09.2016



Нередко на различных страницах одного приложения и даже в разных приложениях могут использоваться повторяющиеся элементы разметки xaml. Например, форма входа, предусматривающая ввод логина и пароля, или форма поиска. Такая форма входа может использоваться в самых различных приложениях для авторизации пользователя. То же самое относится к форме поиска, так как вне зависимости от страницы или приложения она, как правило, будет содержать текстовое поле и кнопку. И чтобы не повторяться, Xamarin Forms позволяет оформить повторяющиеся элементы в виде отдельного визуального компонента, который можно использовать повторно в разных проектах.

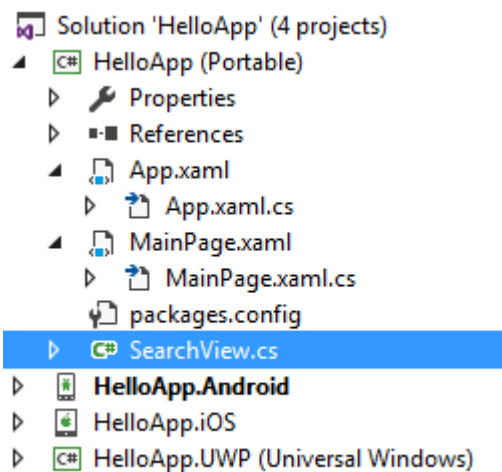
Для создания визуальных компонентов применяется класс **ContentView**. Чтобы создать первый визуальный компонент, который будет представлять окно поиска, добавим в главный проект новый элемент по типу **Forms ContentView**, который назовем **SearchView**:



После этого в проект добавится обычный класс на языке C#, который будет иметь примерно следующий код:

```
using Xamarin.Forms;

namespace HelloApp
{
    public class SearchView : ContentView
    {
        public SearchView()
        {
            Content = new Label { Text = "Hello ContentView" };
        }
    }
}
```



Изменим его код следующим образом:

```
using Xamarin.Forms;
```

```
namespace HelloApp
{
    public delegate void SearchEventHandler(string text);

    public class SearchView : ContentView
    {
        public event SearchEventHandler Search;
        public SearchView()
        {
            Button searchBtn = new Button { Text = "Поиск" };
            Entry searchEntry = new Entry { HorizontalOptions = LayoutOptions.FillAndExpand };

            searchBtn.Clicked += (sender, e) => Search?.Invoke(searchEntry.Text);
            Content = new StackLayout
            {
                Orientation = StackOrientation.Horizontal,
                Spacing = 5,
                Children =
                {
                    searchEntry,
                    searchBtn
                }
            };
        }
    }
}
```

В коде компонента SearchView определены текстовое поле и кнопка, а также событие Search, которое вызывается при нажатии на кнопку. Данное событие будет представлять делегат SearchEventHandler. И через это событие внешний код сможет получить введенное значение и выполнить с ним какие-нибудь действия.

В коде страницы MainPage определим следующее содержимое:

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

namespace HelloApp
{
    public partial class MainPage : ContentPage
    {
        public MainPage()
        {
            List<string> users = new List<string>()
            {
                "Иван Иванов",
                "Олег Кузнецов",
                "Денис Петров",
                "Иван Сидоров",
                "Петр Денисов"
            };

            ListView usersList = new ListView
            {
                ItemsSource = users
            };

            SearchView searchView = new SearchView();
            searchView.Search += (text) =>
            {
                if(!string.IsNullOrEmpty(text))
                {

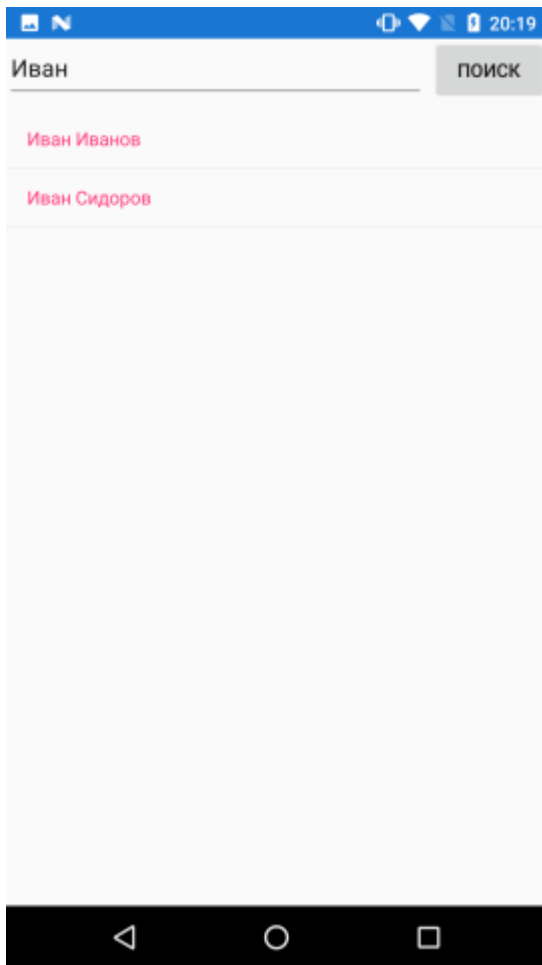
```

```

        usersList.ItemsSource = users.Where(u=>u.Contains(text));
    }
    else
    {
        usersList.ItemsSource = users;
    }
};
Content = new StackLayout { Children = { searchView, usersList } };
}
}
}

```

Для вывода списка пользователей определен элемент ListView. В обработчике события Search происходит фактически фильтрация списка.



Аналогично можно использовать ContentView и в коде xaml. Допустим, у нас есть страница MainPage.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>
        <local:SearchView Search="SearchUsers"></local:SearchView>
        <ListView x:Name="usersList" />
    </StackLayout>
</ContentPage>

```

А в коде страницы пропишем установку списка и обработчик события Search:

```

using System.Collections.Generic;

```

```
using System.Linq;

using Xamarin.Forms;

namespace HelloApp
{
    public partial class MainPage : ContentPage
    {
        List<string> users;

        public MainPage()
        {
            InitializeComponent();

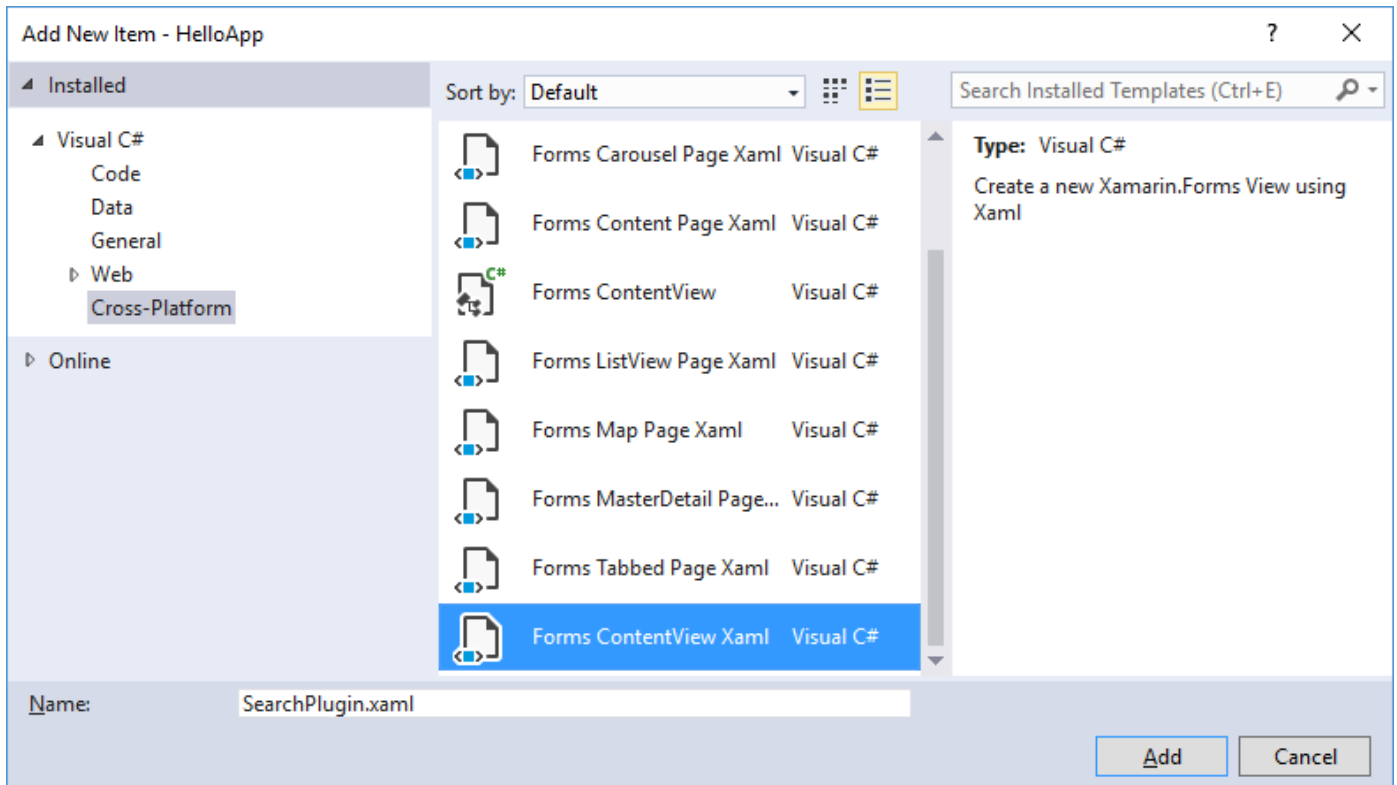
            users = new List<string>()
            {
                "Иван Иванов",
                "Олег Кузнецов",
                "Денис Петров",
                "Иван Сидоров",
                "Петр Денисов"
            };
            userList.ItemsSource = users;
        }

        private void SearchUsers(string text)
        {
            if (!string.IsNullOrEmpty(text))
            {
                userList.ItemsSource = users.Where(u => u.Contains(text));
            }
            else
            {
                userList.ItemsSource = users;
            }
        }
    }
}
```

Результат в данном случае будет аналогичен работе страницы MainPage.

## Определение ContentView в XAML

Не всем удобно определять интерфейс в коде C#, кому-то будет комфортнее работать с XAML. Для этого добавим в проект новый элемент по типу **Forms ContentView Xaml**, который назовем SearchPlugin:



Добавленный файл xaml во многом напоминает код xaml обычной страницы, только в данном случае определяется не `ContentPage`, а элемент `ContentView`. И далее изменим код xaml добавленного файла следующим образом:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentView xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="HelloApp.SearchPlugin">
  <StackLayout Orientation="Horizontal" Spacing="5">
    <Entry x:Name="searchEntry" HorizontalOptions="FillAndExpand" />
    <Button Clicked="OnClicked" Text="Поиск" />
  </StackLayout>
</ContentView>
```

И также изменим файл связанного кода:

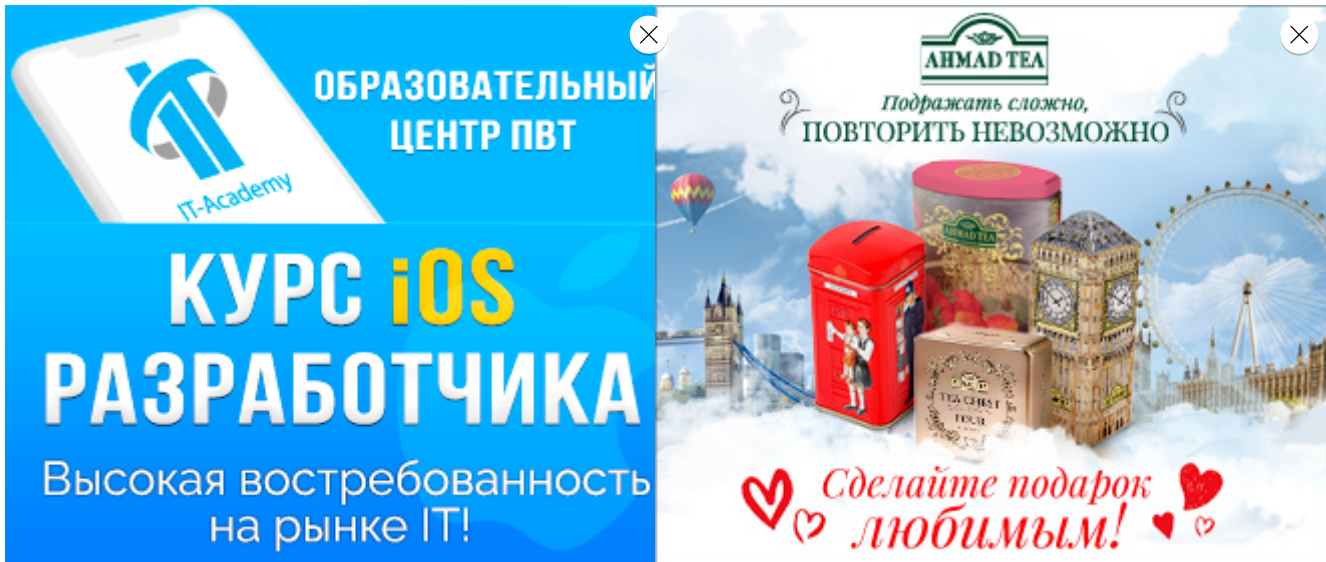
```
using System;
using Xamarin.Forms;

namespace HelloApp
{
    public delegate void SearchPluginEventHandler(string text);
    public partial class SearchPlugin : ContentView
    {
        public event SearchPluginEventHandler Search;

        public SearchPlugin()
        {
            InitializeComponent();
        }
        private void OnClicked(object sender, EventArgs e)
        {
            Search?.Invoke(searchEntry.Text);
        }
    }
}
```

И далее данный класс мы сможем использовать аналогичным образом:

```
<?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>
    <local:SearchPlugin Search="SearchUsers"></local:SearchPlugin>
    <ListView x:Name="usersList" />
  </StackLayout>
</ContentPage>
```



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



G+

### Платье Rosanna

Нарядное платье для праздников, выпускных вечеров и особых случаев, при которых вы будете выглядеть безупр...

Wildberries

~~426 р.~~

107 р.

-15%

0 Комментариев metanit.com

1 Войти ▾

 Рекомендовать Поделиться

Лучшее в начале ▾



Начать обсуждение...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS 

Прокомментируйте первым.

## ТАКЖЕ НА METANIT.COM

**Go | Синхронизация**

1 комментарий • 2 месяца назад

**Roman** — Получается, что каналы и мапы передаются в функцию по ссылке, верно?**Angular в ASP.NET Core | CRUD и маршрутизация. Часть 2**

7 комментариев • 3 месяца назад

**Igor Teterkin** — Поправка - это в IE11. В google chrome всё ок**Go | Visual Studio Code**

1 комментарий • 3 месяца назад

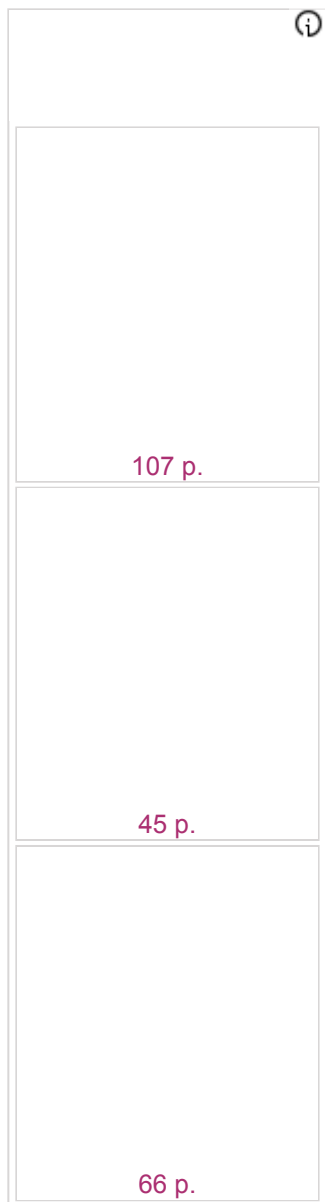
**Алексей Сентюрин** — Для полноты не добавьте пару строк про дебаггинг?)**Django | Введение**

3 комментариев • 23 дня назад

**Andriy Palchik** — Ну да, вы тоже ознакомьтесь... <https://w3techs.com/technol...>  
<https://www.similartech.com...> Подписаться Добавь Disqus на свой сайт [Добавить Disqus](#) [Добавить](#)

Конфиденциальность





[Вконтакте](#) | [Twitter](#) | [Google+](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2012-2017. Все права защищены.