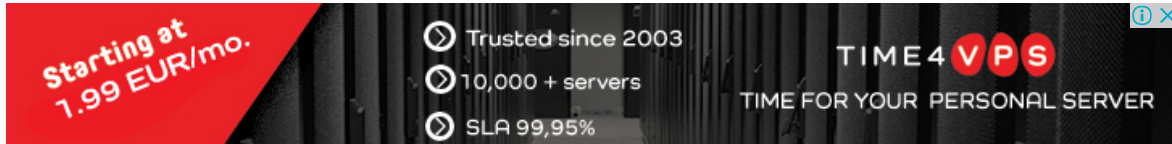




## Навигация

### Основы навигации

Последнее обновление: 25.06.2016



Для поддержки навигации на каждой странице Page в Xamarin Forms определено свойство **Navigation**. Это свойство представляет интерфейс **INavigation**, в котором есть следующие методы:

```
Task PushAsync(Page page)
Task PushModalAsync(Page page)
```

В качестве параметра здесь передается объект Page - страница, на которую надо осуществить переход.

Второй метод имеет в своем названии слово "Modal" и осуществляет переход на модальную страницу. Обычно модальные страницы используются, когда приложению нужно получить некоторую информацию от пользователя. При этом до получения информации нельзя возвращаться на предыдущую страницу.

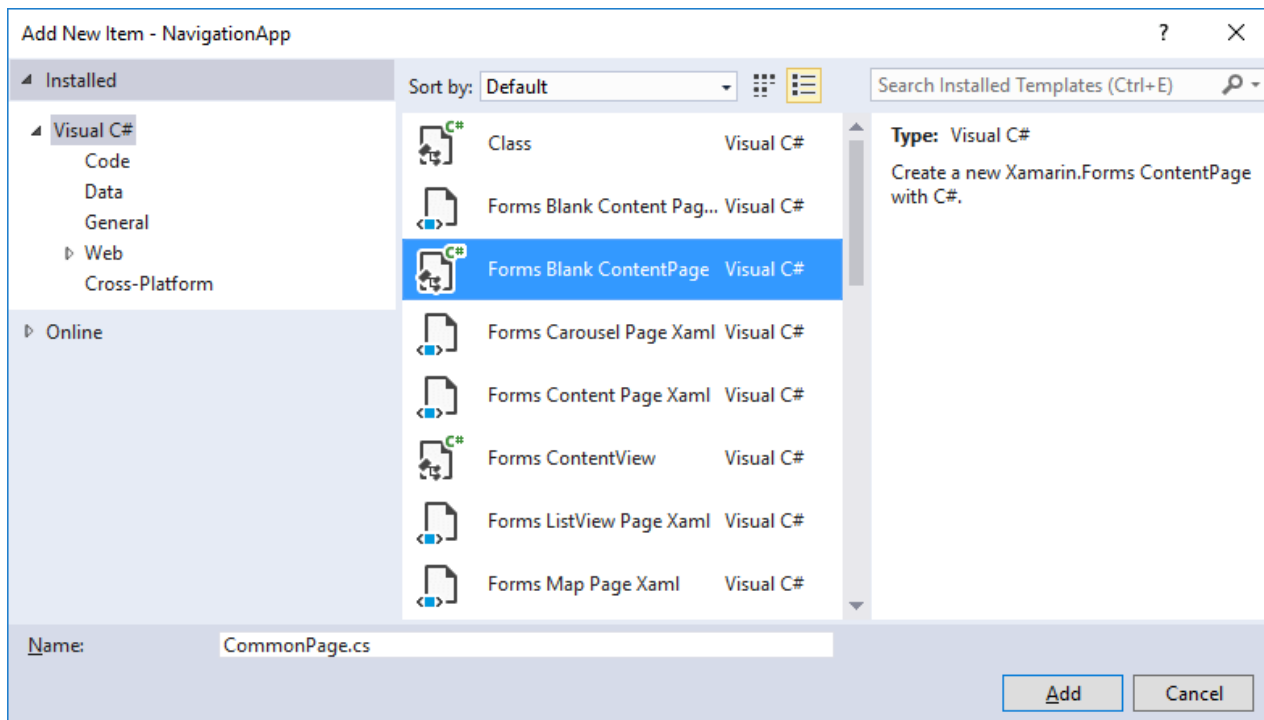
Также имеются методы для возврата на предыдущую страницу:

```
Task<Page> PopAsync()
Task<Page> PopModalAsync()
```

Итак, рассмотрим простейший пример навигации, используя все эти методы. Создадим новый проект по типу Xamarin Forms Portable.

Так как наше приложение будет производить навигацию, то есть переходы между страницами, то вначале добавим эти вспомогательные страницы.

Итак, добавим в проект новый элемент **Forms Blank ContentPage**, который назовем **CommonPage.cs**:



Определим в этом классе следующий код:

```
public class CommonPage : ContentPage
{
    public CommonPage()
    {
        Title = "Common Page";
        Button backButton = new Button
        {
            Text = "Назад",
            HorizontalOptions = LayoutOptions.Center,
            VerticalOptions = LayoutOptions.Center
        };
        backButton.Clicked += BackButton_Click;
        Content = backButton;
    }
    private async void BackButton_Click(object sender, EventArgs e)
    {
        await Navigation.PopAsync();
    }
}
```

На этой странице определяется единственная кнопка, по нажатию на которую осуществляется переход назад с помощью метода `Navigation.PopAsync()`. Поскольку метод является асинхронным, то перед ним ставится ключевое слово `await`, а сам обработчик кнопки помечается с помощью ключевого слова `async`.

Также стоит отметить, что у страницы устанавливается заголовок через свойство `Title`.

Подобным образом добавим вторую страницу, которую назовем **ModalPage**. Определим в ней следующий код:

```
public class ModalPage : ContentPage
{
    public ModalPage()
    {
        Title = "Modal Page";
        Button backButton = new Button
        {
            Text = "Назад",
            HorizontalOptions = LayoutOptions.Center,
            VerticalOptions = LayoutOptions.Center
        };
        backButton.Clicked += BackButton_Click;

        Content = backButton;
    }
    private async void BackButton_Click(object sender, EventArgs e)
    {
        await Navigation.PopModalAsync();
    }
}
```

Данная страница выглядит во многом аналогично предыдущий за тем исключением, что переход назад здесь производится с помощью метода `Navigation.PopModalAsync()`. То есть эта страница будет модальной.

И изменим код главной страницы MainPage:

```
using System;
using Xamarin.Forms;

namespace NavigationApp
{
    public partial class MainPage : ContentPage
    {
        public MainPage()
        {
            Title = "Main Page";
            Button toCommonPageBtn = new Button
            {
                Text = "На обычную страницу",
                HorizontalOptions = LayoutOptions.Center,
                VerticalOptions = LayoutOptions.CenterAndExpand
            };
            toCommonPageBtn.Clicked += ToCommonPage;

            Button toModalPageBtn = new Button
            {
                Text = "На модальную страницу",
                HorizontalOptions = LayoutOptions.Center,
                VerticalOptions = LayoutOptions.CenterAndExpand
            };
            toModalPageBtn.Clicked += ToModalPage;

            Content = new StackLayout { Children = { toCommonPageBtn, toModalPageBtn } };

            private async void ToModalPage(object sender, EventArgs e)
            {
                await Navigation.PushModalAsync(new ModalPage());
            }
            private async void ToCommonPage(object sender, EventArgs e)
            {
                await Navigation.PushAsync(new CommonPage());
            }
        }
    }
}
```

Итак, здесь две кнопки. Одна производит переход на `CommonPage` с помощью метода `Navigation.PushAsync()`, а другая осуществляет переход на `ModalPage` посредством вызова `Navigation.PushModalAsync()`

Но чтобы с `MainPage` можно было бы переходить к `CommonPage` или `ModalPage`, нам надо в качестве главной страницы использовать объект `NavigationPage`, а не `ContentPage`, как установлено по умолчанию. Поэтому перейдем к классу `App` и изменим код установки главной страницы:

```
using Xamarin.Forms;

namespace NavigationApp
{
    public partial class App : Application
    {
        public App()
        {
            InitializeComponent();

            MainPage = new NavigationPage(new MainPage());
        }

        protected override void OnStart() { }

        protected override void OnSleep() { }

        protected override void OnResume() { }
    }
}
```

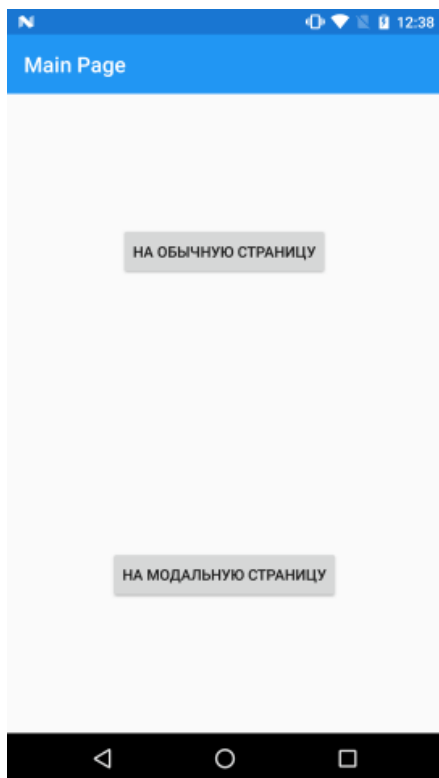
И чтобы использовать механизм навигации в приложении, необходимо обернуть объект `MainPage` в `NavigationPage`:

```
MainPage = new NavigationPage(new MainPage());
```

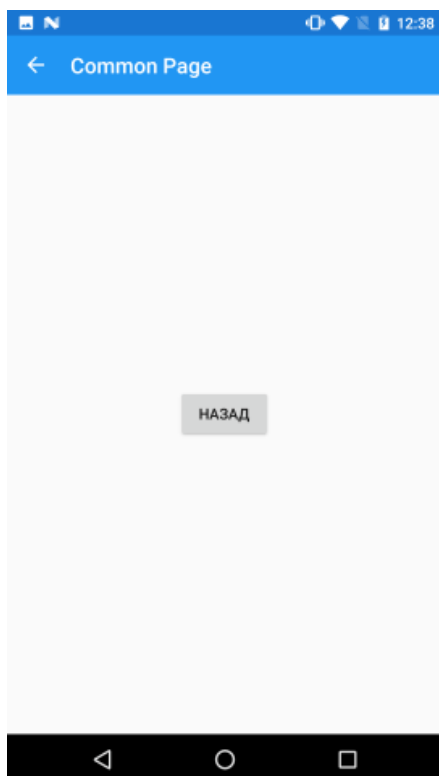
Иначе мы не сможем использовать навигацию в приложении. Сам же класс `NavigationPage` визуально только добавляет заголовок страницы, который устанавливается с помощью свойства `Title` и который отображается рядом с иконкой

приложения. Хотя в зависимости от операционной системы отображение заголовка может отличаться.

Итак, запустим приложение:



Попробуем перейти на обычную страницу, нажав на первую кнопку:



Как из скриншота можно увидеть, что в данном случае даже не нужна кнопка для возвращения назад, так как мы можем вернуться через стрелку рядом с заголовком.

Вернемся назад и нажмем на вторую кнопку для перехода к ModalPage:



Так как для перехода к этой странице использовался метод `Navigation.PushModalAsync()`, то данная страница будет модальной. Основное отличие модальной страницы в визуальном плане состоит в отсутствии заголовка, несмотря на то, что в коде мы устанавливаем свойство `Title` у `ModalPage`.

Ну и для возврата с этой страницы назад нам надо использовать метод `Navigation.PopModalAsync()`

При в реальности для создания модальной страницы необязательно создавать специальный класс страницы, как здесь класс `ModalPage`. Так, мы могли бы использовать в качестве модальной и `CommonPage`.

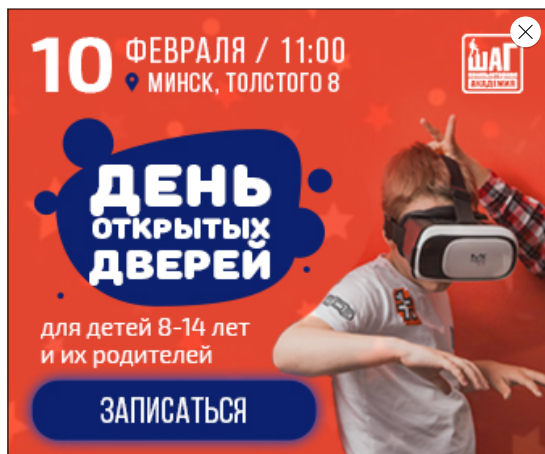
### Анимация в постраничных переходах

Все ранее рассмотренные методы навигации в качестве второго параметра принимают логическое значение:

```
Task PushAsync(Page page, bool animated)
Task PushModalAsync(Page page, bool animated)
Task<Page> PopAsync(bool animated)
Task<Page> PopModalAsync(bool animated)
```

И если оно равно `true`, то при переходе на другую страницу осуществляется анимация, если, конечно, целевая платформа поддерживает подобную анимацию.

По умолчанию анимация включена, поэтому, чтобы ее отключить, нам надо использовать одну из этих версий, передавая значение `false`.



Яндекс.Директ

Что происходит  
в Чернобыле?

yandex.by



If you own or owned a home or other structure with Windsor Pinnacle or Legend Series windows manufactured between January 1, 2000 and January 5, 2018,

Legal Notice

www.windowsettlements.com

Legal Notice

15 Комментариев metanit.com

1 Войти ▾

Рекомендовать 1
 Поделиться

Лучшее в начале ▾



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS ?

Имя

**Andrew** • 3 года назад

При нажатии на кнопку "Добавить", возникает ошибка System.InvalidOperationException: Push Async is not supported globally on Android, please use a NavigationPage. Решением было добавить в конструктор класса "App : Application" строку MainPage = new NavigationPage(new MainPage());

2 ^ | ▾ • Ответить • Поделиться ›

**Наталья Королева** • 3 месяца назад

Добрый день!

Скажите, пожалуйста, что делают данные конструкции?

- 1) toCommonPageBtn.Clicked += ToCommonPage;
- 2) Content = backButton;
- 3) Content = new StackLayout { Children = { toCommonPageBtn, toModalPageBtn } };

^ | ▾ • Ответить • Поделиться ›

**Metanit** Модератор → Наталья Королева • 3 месяца назад

1) в качестве обработчика события нажатия кнопки toCommonPageBtn устанавливается метод ToCommonPage - подробнее тут - <https://metanit.com/sharp/t...>

2) установка содержимого страницы

3) установка содержимого другой страницы

^ | ▾ • Ответить • Поделиться ›

**Сергей** • 4 месяца назад

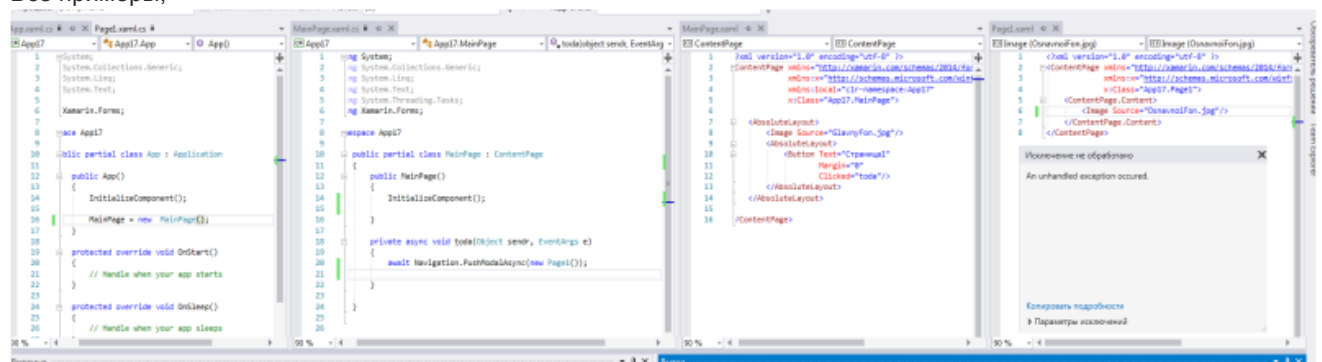
Как сделать что бы он не делал этот стак а просто перешёл на страницу?

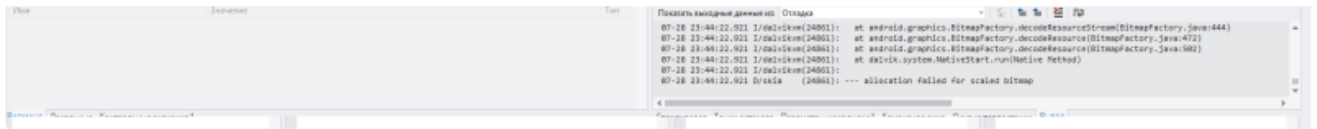
^ | ▾ • Ответить • Поделиться ›

**Erazer** • 6 месяцев назад

Добрый день, подскажите, создаю две страницы и переход между ними, если во второй оставить все как есть при создании то все хорошо и переход работает. но если я пытаюсь изменить XAML файл то появляется исключение. Где, что делаю не так?

Все примеры,





что есть на этой странице работают без проблем!

^ | v • Ответить • Поделиться ›



**Metanit** Модератор → Erazer • 6 месяцев назад

какая-то проблема с изображениями, которые вы пытаетесь использовать

^ | v • Ответить • Поделиться ›



**Erazer** → Metanit • 6 месяцев назад

А по отдельности все работает! Как ее можно исправить или где, что почитать?

^ | v • Ответить • Поделиться ›



**Metanit** Модератор → Erazer • 6 месяцев назад

к сожалению, ничего не могу подсказать поэтому поводу

^ | v • Ответить • Поделиться ›



**Erazer** → Metanit • 6 месяцев назад

Судя по всему слишком большие картинки и переполнение на хипе как я понял. Размер приложения и ресурсов как то можно задать или увеличить?

^ | v • Ответить • Поделиться ›



**Erazer** → Metanit • 6 месяцев назад

Ну по сути если он выдает исключение, значит нужен обработчик исключений который подскажет программе что делать в том или ином случае! А вообще я все тут разбираю путем проб и экспериментов, может я в принципе не правильный подход выбрал? Мне нужно куча страниц и чтобы эти страницы открывали другие страницы. Как это можно реализовать?

^ | v • Ответить • Поделиться ›



**DarkByte** • год назад

Пожалуйста расскажите как увязать навигацию с паттерном MVVM? Очень надо! Потому что это все что тут описано никак с ним не увязывается. :(

^ | v • Ответить • Поделиться ›



**Vitaliyzayats** • год назад

Добрый вечер.

Добавил навигацию подобным образом и сборка с развёртыванием на Android проходят успешно, но при отладке выскакивает окно о необработанном исключении: System.MissingMethodException: Method 'Android.Support.V4.Widget.DrawerLayout.AddDrawerListener' not found.

Как решить данную проблему? UWP запускается без проблем.

Xamarin 4.2.1.64

NuGet Xamarin.forms 2.2.0.45

Запускаю на эмуляторе Visual Studio Emulator for Android, реального устройства не имеется.

^ | v • Ответить • Поделиться ›



**Metanit** Модератор → Vitaliyzayats • год назад

обновите в проекте для Android и в Portable-проекте через Nuget пакет Xamarin.Forms

Плюс еще может быть проблема с эмулятором

^ | v • Ответить • Поделиться ›



**IgorT** • год назад

Добрый день, у меня при нажатии на кнопку просит указать synccontext.cs, расскажите пожалуйста что это и где найти

^ | v • Ответить • Поделиться ›



**Metanit** Модератор → IgorT • год назад


значит, в коде где-то ошибка. Никакой synccontext.cs не надо указывать

^ | v • Ответить • Поделиться ›

ТАКЖЕ НА METANIT.COM


## Vue.js | Взаимодействие между сестринскими компонентами. Шина событий

3 комментария • 4 месяца назад

 **Metanit** — В первом примере и должно быть предупреждение. Оно не говорит о невозможности использовать данный код или что там какая-то ошибка,


## Паттерны в C# и .NET | Fluent Builder

10 комментариев • 3 месяца назад

 **LamaK** — При том, что, если забыть какой-то параметр, то что-то может обвалиться. По-хорошему, в метод Build() можно добавить проверку всех необходимых ...


## Go | Отправка запросов

2 комментария • 17 дней назад

 **Metanit** — потому что после получения ответа клиент и сервер продолжают поддерживать соединение в течение некоторого периода. По истечении этого периода ...

## C++ | Конструкторы и инициализация объектов

1 комментарий • 4 месяца назад

 **Aleksander Hell (EastHell)** — Можно ли сократить код чтобы один конструктор вызывал другой, и при этом использовать списки инициализации?

 Подписаться Добавить Disqus на свой сайт [Добавить Disqus](#) [Добавить](#) Конфиденциальность[Вконтакте](#) | [Twitter](#) | [Google+](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

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