



Создание приложения

Последнее обновление: 18.03.2017



The Most Popular eCommerce Platform
Get Started Today For Free

По умолчанию в созданном в прошлой теме проекте у нас уже есть минимальный код, необходимый для приложения. Само выполнение приложения начинается с файлов App.xaml и App.xaml.cs.

Так, если мы откроем файл **App.xaml.cs**, то увидим там следующий класс:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Xamarin.Forms;

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

            MainPage = new HelloApp.MainPage();
        }

        protected override void OnStart()
        {
            // Handle when your app starts
        }

        protected override void OnSleep()
        {
            // Handle when your app sleeps
        }

        protected override void OnResume()
        {
            // Handle when your app resumes
        }
    }
}
```

Это обычный файл с кодом C#, который даже подключает стандартные для .NET пространства имен за исключением пространства **Xamarin.Forms**. Фактически единственно, что он делает, это устанавливает главную страницу приложения через свойство `MainPage` в конструкторе.

В качестве главной страницы устанавливается объект класса **MainPage**, то есть единственная определенная в проекте страница.

Перейдем к определению этой страницы. Оно разбито на два файла. Файл **MainPage.xaml** представляет визуальный интерфейс страницы в виде кода XAML (как и в WPF/UWP), который аналогичен HTML:

```
<?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"
              x:Class="HelloApp.MainPage">
```

```
<Label Text="Welcome to Xamarin Forms!"
      VerticalOptions="Center"
      HorizontalOptions="Center" />
```

```
</ContentPage>
```

Так как для создания страниц используется класс **ContentPage**, то в качестве корневого элемента в этом файле определен именно элемент ContentPage.

Все его содержимое состоит из одного элемента Label. Класс Label служит для вывода простого текста на экран. Текст задается с помощью атрибута Text. Попутно у метки устанавливаются некоторые свойства позиционирования через атрибуты VerticalOptions и HorizontalOptions.

Таким образом, интерфейс страницы довольно простой - метка с текстом. Но также в проекте есть и файл с кодом логики страницы - файл **MainPage.xaml.cs**:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Xamarin.Forms;

namespace HelloApp
{
    public partial class MainPage : ContentPage
    {
        public MainPage()
        {
            InitializeComponent();
        }
    }
}
```

Однако по факту здесь никакой логики нет, кроме вызова в конструкторе InitializeComponent(). Через этот вызов на странице формируется интерфейс, который определен в файле MainPage.xaml.

Стоит отметить, что чтобы связать оба определения страницы, класс MainPage определяется здесь как частичный (partial), а в файле MainPage.xaml в определении страницы ContentPage указан класс, название которого совпадает с классом из MainPage.xaml.cs:

```
x:Class="HelloApp.MainPage"
```

Главный проект со всеми рассмотренными выше файлами компилируется в библиотеку dll, а остальные три проекта содержат ссылки на него. Так, если мы откроем узел References у каждого проекта, то увидим там ссылку на библиотеку HelloApp - то есть ссылку на главный проект.

Кроме этой библиотеки все три остальных проекта содержат еще ряд важных ссылок, в частности каждый проект содержит ссылку на свою специфическую библиотеку:

- Xamarin.Forms.Platform.Android
- Xamarin.Forms.Platform.iOS
- Universal Windows

Данные библиотеки определяют для своей мобильной платформы статический метод Xamarin.Forms.Init(), который выполняет инициализацию системы Xamarin.Forms.

Например, возьмем проект для Android. В нем по умолчанию имеется файл **MainActivity.cs**, с которого собственно начинается выполнение проекта на Android:

```
public class MainActivity : global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity
{
    protected override void OnCreate(Bundle bundle)
    {
        TabLayoutResource = Resource.Layout.Tabbar;
        ToolbarResource = Resource.Layout.Toolbar;

        base.OnCreate(bundle);

        global::Xamarin.Forms.Forms.Init(this, bundle);
        LoadApplication(new App());
    }
}
```

Вначале происходит вызов метода `Init()`, а затем экземпляр класса `App` передается в метод `LoadApplication()`, который определен у класса `FormsApplicationActivity`.

В остальных проектах инициализация будет происходить подобным образом, немного отличаясь в конкретной реализации. Например, в проекте для iOS то же самое будет происходить в классе **AppDelegate**:

```
public partial class AppDelegate : global::Xamarin.Forms.Platform.iOS.FormsApplicationDelegate
{
    public override bool FinishedLaunching(UIApplication app, NSDictionary options)
    {
        global::Xamarin.Forms.Forms.Init();
        LoadApplication(new App());

        return base.FinishedLaunching(app, options);
    }
}
```

А в проекте для UWP все подобные действия определены в файле **App.xaml.cs**.

Итак, фактически наш проект по умолчанию уже представляет простейшее приложение, которое просто выводит на экран некоторый текст. И теперь рассмотрим, как мы можем запускать приложение на разных ОС.



Mad King

Новый с

rbkgames.com

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



Ищешь Python курсы? ▾

shultais.education

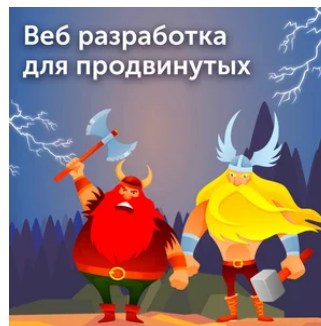


18+

Яндекс.Директ

Школа программирования Loftschool ▾

loftschool.com



Яндекс.Директ

13 Комментариев [metanit.com](#)

Войти ▾

Рекомендовать 2 Поделиться

Лучшее в начале ▾



Присоединиться к обсуждению...

войти с помощью

или через DISQUS

Имя



Владимир • год назад

Добрых суток, Админ, такое дело - как только создаю новый проект (так же, как и в Вашем примере), у меня появляются сразу 9 ошибок - студия ругается на xml-файл, дескать нет каких то ресурсов. В сети решение не нашёл((

^ | ▾ • Ответить • Поделиться ▸



Владимир → Владимир • год назад

Да еще - построение решения не помогает, пробовал отладку на Андроиде (и с телефона и с эмулятора) - ругается точно также

^ | v • Ответить • Поделиться ›



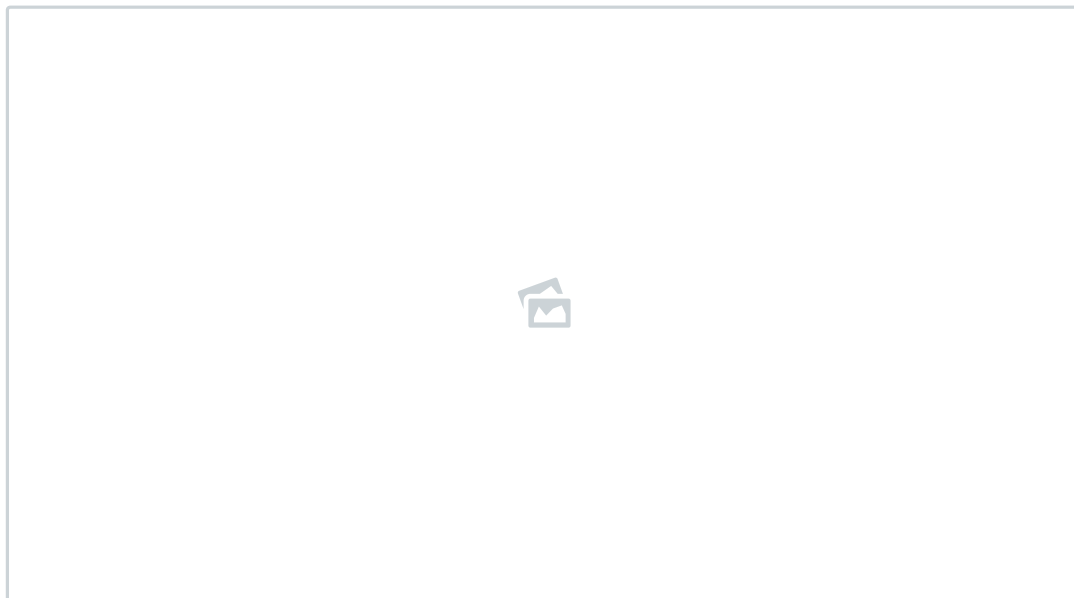
Metanit Модератор → Владимир • год назад

покажите скриншот ошибок.

^ | v • Ответить • Поделиться ›



Константин Крылович → Metanit • год назад



^ | v • Ответить • Поделиться ›



Metanit Модератор → Константин Крылович • год назад

это проект с нуля? ничего не добавлялось?

^ | v • Ответить • Поделиться ›



Константин Крылович → Metanit • год назад

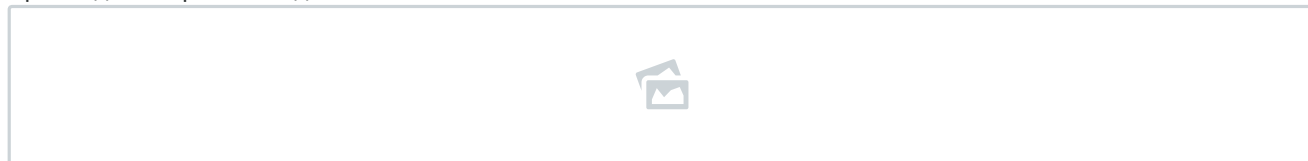
да, может что-то не верно установилось

^ | v • Ответить • Поделиться ›



Mike Zh • год назад

При создании проекта выдается вот такая вот ошибка



^ | v • Ответить • Поделиться ›



Metanit Модератор → Mike Zh • год назад

попробуйте добавить в проект для андроида через нугет Xamarin.Forms.Platform.Android и Xamarin.Forms

^ | v • Ответить • Поделиться ›



Алексей → Metanit • год назад

Ошибка решилась путем установки обновлений VS

^ | v • Ответить • Поделиться ›



Алексей → Metanit • год назад

В ссылке они уже указаны, проблема не решилась

^ | v • Ответить • Поделиться ›

**Карен** • 2 года назад

Metanit телефон не отображается в visual studio

^ | v • Ответить • Поделиться ›

**Евгений Комиссаров** → Карен • год назад

У меня была такая же проблема на ноутбуке, когда подключал телефон на Android, перепробовал многое: драйвера, кабель, обновления. Но оказалось, что телефон просто нужно воткнуть в другой порт и всё заработало, хотя файлы через оба порта передаются корректно.

^ | v • Ответить • Поделиться ›

**Metanit** Модератор → Карен • 2 года назад

проблема с драйверами, проблема с кабелем, проблема с телефоном - включен режим разработчика, какой-то другой глюк визуал студии, больше предположений нет

^ | v • Ответить • Поделиться ›

ТАКЖЕ НА METANIT.COM

Руководство по Angular в ASP.NET Core 2.0

2 комментария • месяц назад

Metanit — да, еще будут статьи**C# и .NET | Соккрытие методов**

3 комментария • 2 месяца назад

Везнич — пример, чтоб видно разницу между new и override

```
class Water { public virtual void Drink() { Console.WriteLine("water life"); } } class Tea : Water { public
```

Vue.js | Props

2 комментария • 3 месяца назад

Metanit — нет, и так должно работать, проверил в Google Chrome, MS Edge, Firefox

Angular в ASP.NET Core | Маршрутизация

2 комментария • месяц назад

Zaur YakubOff — Если создал проект на основе Asc.NetCore MVC куда нужно вставить <base href="/" /> ???

✉ Подписаться

D Добавить Disqus на свой сайтДобавить DisqusДобавить

🔒 Конфиденциальность


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

Контакты для связи: metanit22@mail.ru

<https://metanit.com/sharp/xamarin/2.2.php>

