



Ресурсы и стили

Концепция ресурсов в Xamarin Forms

Последнее обновление: 12.09.2016



Помогите им найти вашу
компанию в Google!

[НАЧНИТЕ СЕЙЧАС](#)

Компенсирuem
до \$60

Google AdWords

Для совместного использования одних и тех же компонентов различными элементами Xamarin Forms применяется концепция ресурсов. В данном случае под ресурсами понимаются не вспомогательные файлы - изображений и т.д., которые используются в приложении, а логические ресурсы, которые определяются в коде C# или XAML.

Определение ресурсов

В качестве ресурса можно определить любой объект. Все ресурсы помещаются в объект **ResourceDictionary**. У каждого визуального объекта, например, **ContentPage** или **Button**, имеется свойство **Resources**, которое как раз хранит объект **ResourceDictionary**. Например, определим несколько ресурсов:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="StylesApp.MainPage">
  <ContentPage.Resources>
    <ResourceDictionary>
      <Color x:Key="textColor">#004D40</Color>
      <Color x:Key="backColor">#80CBC4</Color>
      <x:String x:Key="fontSize">Large</x:String>
    </ResourceDictionary>
  </ContentPage.Resources>
  <StackLayout>
    <Button Text="iOS" TextColor="{StaticResource Key=textColor}"
      BackgroundColor="{StaticResource Key=backColor}" FontSize="{StaticResource Key=fontSize}" />
    <Button Text="Android" TextColor="{StaticResource Key=textColor}"
      BackgroundColor="{StaticResource Key=backColor}" FontSize="{StaticResource Key=fontSize}" />
  </StackLayout>
</ContentPage>
```

Каждый ресурс должен иметь ключ, задаваемый с помощью атрибута **x:Key**. Это своего рода уникальный идентификатор ресурса. Например:

```
<Color x:Key="textColor">#004D40</Color>
```

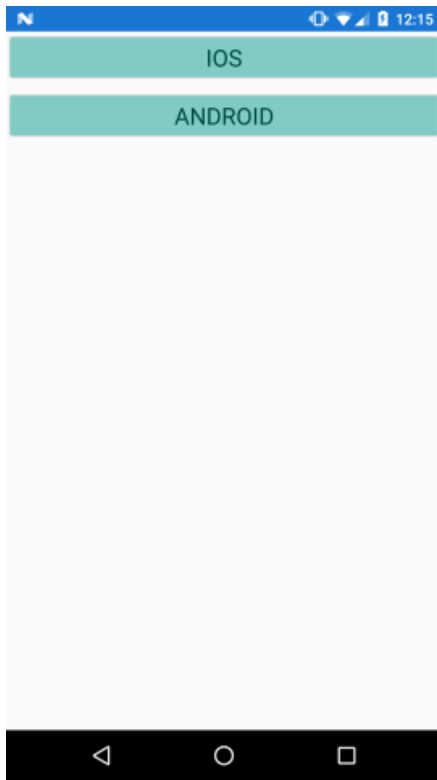
Здесь в качестве ресурса определяется объект **Color**. Этот объект имеет ключ **textColor** и значение **#004D40**.

Чтобы обратиться к этому ресурсу в коде, надо использовать расширение **StaticResource**:

```
TextColor="{StaticResource Key=textColor}"
```

Свойство **Key** через ключ ресурса будет ссылаться на данный ресурс.

То есть в итоге получится следующая страница:



При этом важно, что на данный ресурс могут ссылаться сразу несколько элементов. Например, мы захотели создать общий для всех кнопок цвет фона. И в этом случае проще определить цветовой ресурс, чем присваивать свойству `BackgroundColor` у каждой кнопки конкретный цвет. А в случае, если мы захотим изменить цвет кнопок на другой, то не надо будет менять свойство `BackgroundColor` у всех кнопок. Достаточно будет поменять значение ресурса.

Уровни ресурсов

Ресурсы могут определяться на трех уровнях:

- На уровне отдельного элемента управления. Такие ресурсы могут применяться ко всем вложенным элементам, которые определены внутри этого элемента
- На уровне всей страницы. Такие ресурсы могут применяться ко всем элементам на странице
- На уровне всего приложения. Эти ресурсы доступны из любого места и из любой страницы приложения.

Выше в примере ресурсы определялись а уровне страницы. Теперь опустимся на уровень ниже и определим их на уровне элемента `StackLayout`:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="StylesApp.MainPage">
  <StackLayout>
    <StackLayout.Resources>
      <ResourceDictionary>
        <Color x:Key="textColor">#004D40</Color>
        <Color x:Key="backColor">#80CBC4</Color>
        <x:String x:Key="fontSize">Large</x:String>
      </ResourceDictionary>
    </StackLayout.Resources>
    <Button Text="ios" TextColor="{StaticResource Key=textColor}"
      BackgroundColor="{StaticResource Key=backColor}" FontSize="{StaticResource Key=fontSize}" />
    <Button Text="Android" TextColor="{StaticResource Key=textColor}"
      BackgroundColor="{StaticResource Key=backColor}" FontSize="{StaticResource Key=fontSize}" />
  </StackLayout>
</ContentPage>
```

По сути результат в данном случае будет от же самый, так как все элементы все равно определены внутри `StackLayout`.

Ресурсы приложения

Для определения общих для всего приложения ресурсов в VS 2017 в проекте присутствует файл `App.xaml`, который связан с основным файлом приложения `App.xaml.cs`. Изменим файл `App.xaml` следующим образом:

```
<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="StylesApp.App">
  <Application.Resources>

    <ResourceDictionary>
      <Color x:Key="textColor">#004D40</Color>
      <Color x:Key="backColor">#80CBC4</Color>
      <x:String x:Key="fontSize">Large</x:String>
    </ResourceDictionary>

  </Application.Resources>
</Application>
```

То есть здесь определены все те же ресурсы, что и ранее, только теперь они будут доступны для любого элемента на любой странице внутри приложения.

И в этом случае мы просто можем их использовать:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="StylesApp.MainPage">
  <StackLayout>
    <Button Text="iOS" TextColor="{StaticResource Key=textColor}"
      BackgroundColor="{StaticResource Key=backColor}" FontSize="{StaticResource Key=fontSize}" />
    <Button Text="Android" TextColor="{StaticResource Key=textColor}"
      BackgroundColor="{StaticResource Key=backColor}" FontSize="{StaticResource Key=fontSize}" />
  </StackLayout>
</ContentPage>
```

Управление ресурсами в коде C#

Аналогичный пример использования ресурсов выглядел бы следующим образом:

```
public partial class MainPage : ContentPage
{
    public MainPage()
    {
        Color textColor = Color.FromRgb(0, 77, 64);
        Color backColor = Color.FromRgb(128, 203, 196);
        Double fontSize = 21;
        ResourceDictionary resDict = new ResourceDictionary();
        // добавляем ресурсы в словарь
        resDict.Add("textColor", textColor);
        resDict.Add("backColor", backColor);
        resDict.Add("fontSize", fontSize);
        // устанавливаем словарь ресурсов
        this.Resources = resDict;

        Button iosButton = new Button { Text = "iOS" };
        // получаем ресурс из словаря
        iosButton.TextColor = (Color)Resources["textColor"];
        iosButton.BackgroundColor = (Color)Resources["backColor"];
        iosButton.FontSize = (double)Resources["fontSize"];

        Button androidButton = new Button { Text = "Android" };
        androidButton.TextColor = (Color)Resources["textColor"];
        androidButton.BackgroundColor = (Color)Resources["backColor"];
        androidButton.FontSize = (double)Resources["fontSize"];

        Content = new StackLayout
        {
            Children = { iosButton, androidButton }
        };
    }
}
```

Для управления ресурсами применяются методы и свойства ResourceDictionary:

- Add(string key, object resource): добавляет объект с ключом key в словарь, причем в словарь можно добавить любой объект, главное ему сопоставить ключ
- Remove(string key): удаляет из словаря ресурс с ключом key

Чтобы найти ресурс в словаре, достаточно обратиться по ключу:

```
androidButton.TextColor = (Color)Resources["textColor"];
```

Так как словарь ресурсов хранит объекты типа object, то при получении ресурса его надо привести к нужному типу.

Динамические ресурсы

Ресурсы могут быть статическими и динамическими. Рассмотрим разницу на примере:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="StylesApp.MainPage">
  <ContentPage.Resources>
    <ResourceDictionary>
      <Color x:Key="textColor">Red</Color>
    </ResourceDictionary>
  </ContentPage.Resources>
  <StackLayout>
    <Button Text="Изменить" Clicked="ColorChange" TextColor="{StaticResource Key=textColor}" />
  </StackLayout>
</ContentPage>
```

В файле кода C# добавим обработчик для кнопки:

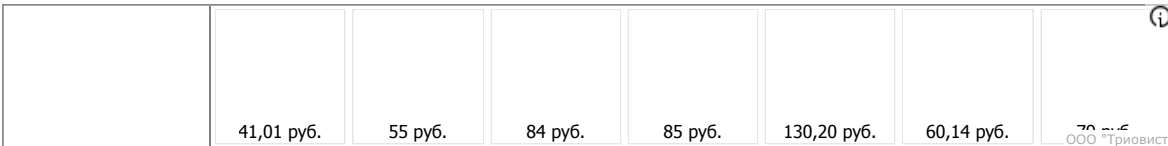
```
private void ColorChange(object sender, EventArgs e)
{
    Color textColor = (Color)Resources["textColor"];
    Resources["textColor"] = textColor == Color.Red ? Color.Green : Color.Red;
}
```

То есть здесь у кнопки цвет текста привязан к ресурсу "textColor". В обработчике кнопки мы пытаемся поменять значение ресурса - если цвет красный, то меняем на зеленый и наоборот.

Однако после запуска приложения и нажатия на кнопку кнопка не меняет цвет текста, так как ресурс "textColor" установлен как статический. То есть после установки он уже не изменяется. И чтобы изменение срабатывало, данный ресурс надо установить как динамический. Для этого вместо StaticResource используется расширение DynamicResource:

```
<Button Text="Изменить" Clicked="ColorChange" TextColor="{DynamicResource Key=textColor}" />
```

Весь остальной код остается без изменений.



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



Нужно программировать контроллеры? ▾

festo.com



Яндекс.Директ

1 Комментарий metanit.com

1 Войти ▾

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

Лучшее в начале ▾



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS (?)

Имя



Комментарий был удален.

**Metanit** Модератор ➔ Guest • 4 месяца назад

нет, в статье же приведен код

^ | ▾ • Ответить • Поделиться ›

ТАКЖЕ НА METANIT.COM

Руководство по Angular в ASP.NET Core 2.0

2 комментариев • 2 месяца назад

Metanit — да, еще будут статьи**Go | Каналы**

2 комментариев • 4 дня назад

Metanit — поправил**React | Сервер. Node.js**

9 комментариев • 3 месяца назад

Stanimir Stankov — Можно с WebStorm - Built-in Server.**C# и .NET | Раннее и позднее связывание**

3 комментариев • 2 месяца назад

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

✉ Подписаться

🔗 Добавить Disqus на свой сайтДобавить DisqusДобавить

🔒 Конфиденциальность



Tribal Print Hoodies



Shop Now >>

DressLily

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

Контакты для связи: metanit22@mail.ru

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