



## Хранение данных. Файлы

Последнее обновление: 17.09.2016



При работе с данными мы нередко сталкиваемся с проблемой их сохранения. В Xamarin Forms есть различные способы сохранения данных в зависимости от выбранного типа хранилища. Можно выделить четыре типа хранилища данных:

- Коллекция `Properties` в классе `Application`, которая предназначена для хранения каких-то временных данных
- Настройки (`Preferences`) операционной системы для хранения каких-то атомарных данных типа `int` или `string`
- Файлы в файловой системе
- Базы данных

## Свойство `Properties`

Свойство **`Properties`** класса `Application` представляет словарь, где каждому ключу с типом `string` сопоставляется некоторое значение. Содержимое этого словаря автоматически сохраняется перед тем, как приложение завершается, поэтому при новом запуске приложения мы сможем использовать ранее сохраненные в словаре данные.

При этом мы не ограничены только классом `Application`, в котором собственно определено данное свойство. Мы также можем к нему обращаться в коде страниц с помощью выражения **`App.Current.Properties`**.

Принципы работы с `Properties` те же, что и со стандартными словарями.

Добавление данных в словарь:

```
App.Current.Properties.Add("name", "Tom");  
// или так  
App.Current.Properties["name"] = "Tom";
```

Получение значения:

```
string name = App.Current.Properties["name"];
```

Получение с проверкой на наличие:

```
object name = "";  
if(App.Current.Properties.TryGetValue("name", out name))  
{  
    // выполняем действия, если в словаре есть ключ "name"  
}
```

Удаление:

```
App.Current.Properties.Remove("name");
```

Например, определим в проекте страницу `MainPage`. В коде `xaml` пропишем текстовое поле и кнопку:

```
<?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="StateApp.MainPage">  
    <StackLayout>  
        <Entry x:Name="nameBox" />  
        <Button Text="Сохранить" Clicked="OnClick" />  
    </StackLayout>
```

&lt;/ContentPage&gt;

А в коде C# у страницы определим добавление и извлечение из словаря Properties введенного в текстовое поле текста:

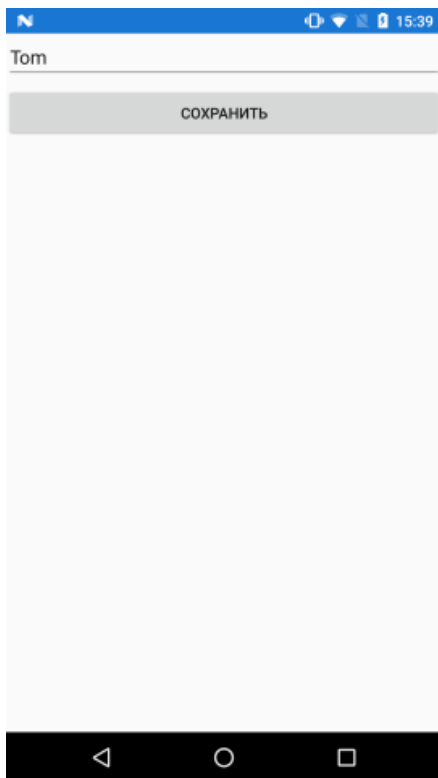
```
using System;
using Xamarin.Forms;

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

        protected override void OnAppearing()
        {
            object name = "";
            if (App.Current.Properties.TryGetValue("name", out name))
            {
                nameBox.Text = (string)name;
            }
            base.OnAppearing();
        }

        private void OnClick(object sender, EventArgs e)
        {
            string value = nameBox.Text;

            App.Current.Properties["name"] = value;
        }
    }
}
```



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



# Buy Bitcoin within 60 seconds - Instant sign up with Via/MC

Easily buy and trade major cryptocurrencies on Canada's safest exchange. [einstein.exchange](https://einstein.exchange)



22 Комментариев metanit.com

Войти

Рекомендовать

Поделиться

Лучшее в начале



Присоединиться к обсуждению...

войти с помощью

или через DISQUS ?

Имя



**Balalamb** • 10 месяцев назад

Добрый день!

А есть возможность сериализации объектов, как для обычного десктопного проекта?

В документации эти функции вроде как описаны, но непонятен механизм их выполнения. Так как атрибут [Serializable] для класса применить не получается.

^ | v • Ответить • Поделиться ›



**Metanit** Модератор → Balalamb • 10 месяцев назад

С помощью библиотеки Newtonsoft.Json можно попробовать сериализовать объект в json или, наоборот, десериализовать

^ | v • Ответить • Поделиться ›



**Balalamb** → Metanit • 10 месяцев назад

а где физически будет лежать сохраненный файл?

^ | v • Ответить • Поделиться ›



**Metanit** Модератор → Balalamb • 10 месяцев назад

в папке приложения

^ | v • Ответить • Поделиться ›



**Balalamb** → Metanit • 10 месяцев назад

Весь день с этой Newtonsoft.Json бодался, вроде поток сформировать получается, а как записать его в файл не соображу. FileStream не работает.

^ | v • Ответить • Поделиться ›



**Metanit** Модератор → Balalamb • 10 месяцев назад

смотрите дальше через одну статью - там про работу с файлами

^ | v • Ответить • Поделиться ›



**Balalamb** → Metanit • 10 месяцев назад

Там описывается, как в файл писать текст, а не поток, или я что-то не правильно понимаю?

^ | v • Ответить • Поделиться ›



**Metanit** Модератор → Balalamb • 10 месяцев назад

с помощью библиотеки Newtonsoft.Json сериализуете объект в строку и сохраняете в файл

^ | v • Ответить • Поделиться ›



**Balalamb** → Metanit • 10 месяцев назад

спасибо, за быстрые ответы, но вопросы все появляются и появляются, Newtonsoft.Json сериализирует только открытые параметры. Есть ли возможность сериализировать класс целиком на подобии сверх-удобной бинарной сериализации?

^ | v • Ответить • Поделиться ›

**Fot Hjk** → Balalamb • 2 месяца назад

Для сериализации приватных полей используйте атрибуты DataContext. Можно сделать и по-другому: с помощью Json сериализовать объект в строку, затем записать его в словарь App.Current.Properties и брать из словаря сериализованный объект, когда это необходимо.

2 ^ | ▾ • Ответить • Поделиться ›

**Balalamb** → Fot Hjk • 2 месяца назад

Спасибо!

^ | ▾ • Ответить • Поделиться ›

**Ярослав Орлов** • год назад

Добрый вечер, как и где можно установить статические значения в Properties, что бы при старте приложения уже имелся какой-то стартовый набор свойств, которые можно было считать?

^ | ▾ • Ответить • Поделиться ›

**Phenman** → Ярослав Орлов • месяц назад

для управления настройками вашего приложения используйте плагин <https://github.com/jamesmon...>

^ | ▾ • Ответить • Поделиться ›

**Metanit** Модератор → Ярослав Орлов • год назад

Можно попробовать использовать метод OnStart класса App. Установка так же, как и в примере выше

^ | ▾ • Ответить • Поделиться ›

**Ярослав Орлов** → Metanit • год назад

Это я понимаю, таким способом получается что при старте приложения всегда будет обновляться моя переменная, а мне надо что бы как на подобие WinForms можно было зайти в свойства проекта и хранить словарь ключ-значение, что бы потом в ходе работы приложения можно было сохранить новое значение в эти свойства и при новом старте приложения обращаться к уже пересохраненному значению.

^ | ▾ • Ответить • Поделиться ›

**Metanit** Модератор → Ярослав Орлов • год назад

А можно проверить - если есть значение, то извлекать имеющееся, если нет - то сохранять новое

^ | ▾ • Ответить • Поделиться ›

**Ярослав Орлов** → Metanit • год назад

То есть механизма описанного выше как в WinForms нету?

^ | ▾ • Ответить • Поделиться ›

**Metanit** Модератор → Ярослав Орлов • год назад

ну если только через ресурсы пойти

^ | ▾ • Ответить • Поделиться ›

**Ярослав Орлов** → Metanit • год назад

Можно подробнее, в каком смысле через ресурсы? В данном случае я поставил себе задачу что в зависимости от "названия стиля", который хранится в определенной переменной Properties я присваиваю определенные стили приложению, так сказать делаю темы приложения. Если есть еще как то ресурсы использовать, подскажите что это или где ознакомиться.

^ | ▾ • Ответить • Поделиться ›

**Metanit** Модератор → Ярослав Орлов • год назад

использование ресурсов было описано в главе по локализации - <http://metanit.com/sharp/xa...>

^ | ▾ • Ответить • Поделиться ›

**Ярослав Орлов** → Metanit • год назад

Мы не много друг друга не поняли, я вот об этом в формах



Вот о чем, и получается что еще до самого первого запуска приложения я могу записать туда значение какого то типа и юзать или сохранять туда что либо в runtime.

^ | v • Ответить • Поделиться ›



**Metanit** Модератор ➔ Ярослав Орлов • год назад


в xamarin, по-моему, нету полного аналога подобным настройкам

^ | v • Ответить • Поделиться ›

ТАКЖЕ НА METANIT.COM

## Vue.js | Локальные и глобальные фильтры

1 комментарий • 4 месяца назад


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

## Go | Каналы

месяц назад


## 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?

## React | React-router и webpack

8 комментариев • 3 месяца назад

 **Юрий Демин** — да, установлен. решил проблему обновив OS, но конкретной причины и связи с обновлением системы так и не нашел!) Спасибо за помощь и за ...