



Всплывающие окна

Последнее обновление: 01.07.2016



	40,85 руб.	59,98 руб.	63 руб.	85 руб.	88 руб.	60,14 руб.	51,20 руб.

В Xamarin Forms для создания всплывающих окон используются специальные методы, которые определены у объекта Page, а поэтому есть у любой страницы:

```
Task DisplayAlert (string title, string message, string cancel)
Task<bool> DisplayAlert (string title, string message, string accept, string cancel)
Task<string> DisplayActionSheet (string title, string cancel,
                                string destruction, params string[] buttons)
```

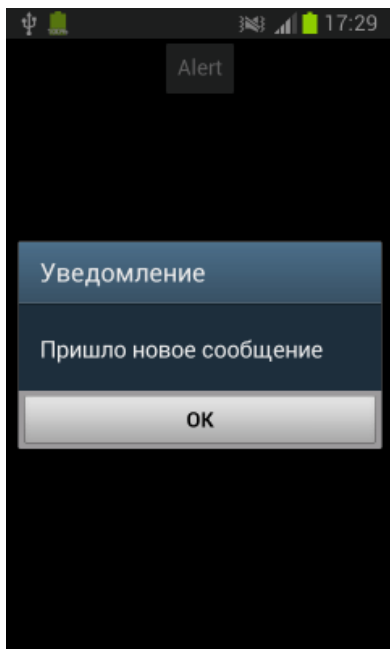
Используем первую версию:

```
class MainPage : ContentPage
{
    public MainPage()
    {
        Button alertButton = new Button
        {
            Text = "Alert",
            VerticalOptions = LayoutOptions.Start,
            HorizontalOptions = LayoutOptions.Center
        };
        alertButton.Clicked += AlertButton_Clicked;

        Content = alertButton;
    }

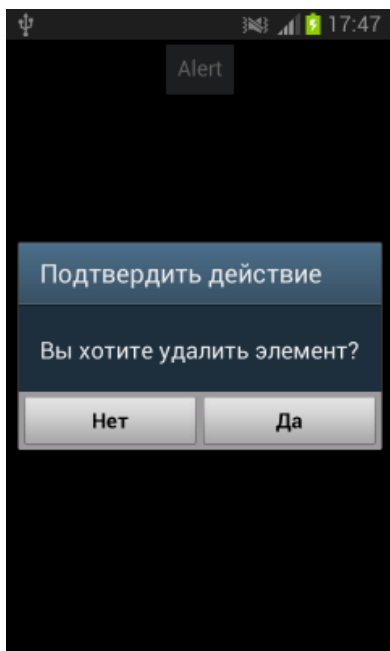
    private void AlertButton_Clicked(object sender, EventArgs e)
    {
        DisplayAlert("Уведомление", "Пришло новое сообщение", "OK");
    }
}
```

Первый параметр передает заголовок окна, второй - текст окна, третий - нажпись на кнопки отмены.



Эта версия метода просто выводит некоторое диагностическое сообщение. Но если нам необходимо, чтобы всплывающее окно предоставляло пользователю некоторый выбор, который надо обработать, то можно использовать вторую версию. Для этого изменим обработчик кнопки:

```
private async void AlertButton_Clicked(object sender, EventArgs e)
{
    bool result = await DisplayAlert("Подтвердить действие", "Вы хотите удалить элемент?", "Да", "Нет");
    await DisplayAlert("Уведомление", "Вы выбрали: " + (result ? "Удалить" : "Отменить"), "ОК");
}
```



DisplayActionSheet

Всплывающее окно, которое создается методом `DisplayActionSheet()`, предоставляет возможность выбрать из ряда действий. Получив выбранное значение, затем мы можем произвести какие-то действия на основании этого значения.

Метод `DisplayActionSheet()` принимает следующие параметры:

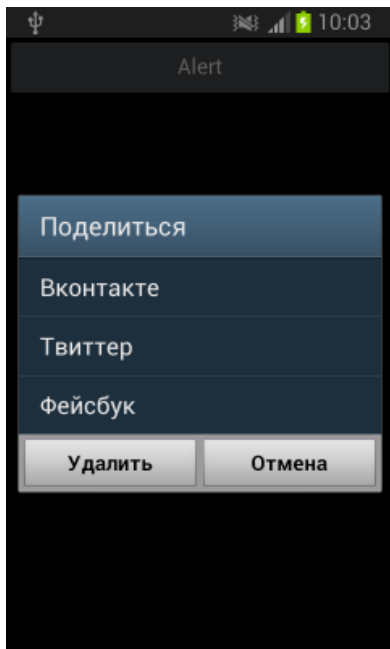
```
Task<string> DisplayActionSheet (String title, String cancel, String destruction, params String[] buttons)
```

- `title`: заголовок окна
- `cancel`: текст для кнопки отмены. Может быть равен `null`, если мы хотим скрыть эту кнопку
- `destruction`: текст для кнопки удаления. Также для скрытия этой кнопки можно передать значение `null`

- buttons: текстовые метки для дополнительных кнопок

```
class MainPage : ContentPage
{
    Label actionLabel;
    public MainPage()
    {
        Button alertButton = new Button { Text = "Alert" };
        alertButton.Clicked += AlertButton_Clicked;
        actionLabel = new Label();
        Content = new StackLayout { Children = { alertButton, actionLabel } };
    }

    private async void AlertButton_Clicked(object sender, EventArgs e)
    {
        var action = await DisplayActionSheet("Поделиться", "Отмена", "Удалить", "Вконтакте", "Твиттер", "Фейсбук");
        actionLabel.Text = action;
    }
}
```



Контекстная реклама

Настройка, ведение контекстной рекламы. Запуск 3 дня. От 860 рублей. Заказывайте сейчас! gusarov-group.by



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



Выбрать!

Рождественская распродажа!
Только до 15 января!

Скидки до 15% на технику Karcher!

Гибкий кабель-канал

promtechnolog.ru



Яндекс.Директ

5 Комментариев metanit.com

1 Войти ▾

♥ Рекомендовать  Поделиться

Лучшее в начале ▾



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS **Makarkin & Partners / Макаркин** • год назад

а для XAML?

^ | ▾ • Ответить • Поделиться ›

**Кирилл** → Makarkin & Partners / Макаркин • год назад

Можешь создать любые всплывающие окна с помощью этого плагина, даже с xaml.

<https://github.com/rotorgam...>

1 ^ | ▾ • Ответить • Поделиться ›

**Metanit** Модератор → Makarkin & Partners / Макаркин • год назад

для XAML нет аналога

^ | ▾ • Ответить • Поделиться ›

**Makarkin & Partners / Макаркин** → Metanit • год назад

прикольно, чтобы делать что-то полноценное про XAML впринципе можно забыть?

^ | ▾ • Ответить • Поделиться ›

**Metanit** Модератор → Makarkin & Partners / Макаркин • год назад

нет, почему же. вам никто не мешает определить свою логику работы окон, создав свои элементы в xaml и управляя их видимостью и обработкой их событий

^ | ▾ • Ответить • Поделиться ›

ТАКЖЕ НА METANIT.COM

Kotlin | Руководство

2 комментария • месяц назад

**Артур Голояд** — Просто шикарно, может подкинете материалов по работе с сетями на Kotlin?

Go | Соответствие интерфейсу

1 комментарий • 16 дней назад

**Пу** — Еще давай. Не останавливайся, пожалуйста. Это гениальный язык. Он, конечно, не божественен, как шарп, но гениальный.

Angular в ASP.NET Core | Маршрутизация

2 комментария • 2 месяца назад

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

Angular в ASP.NET Core | Совмещение маршрутизации

2 комментария • 2 месяца назад

**Metanit** — это больше вопрос предпочтений Подписаться  Добавить 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.