



## Подключение к существующей базе данных

Последнее обновление: 16.09.2016

Department of  
Management**Global Master's in Management**[Click here for details](#)

В прошлой теме мы подключались к базе данных SQLite, которая автоматически создавалась (при ее отсутствии) при запуске приложения. Однако это не всегда бывает удобно. Иногда необходимо, чтобы к первому запуску приложения база данных уже имела некоторые начальные данные, которые мы тут же можем использовать. Поэтому рассмотрим, как мы можем добавить в проект уже имеющуюся базу данных SQLite и использовать ее.

Вначале создадим саму базу данных. Для работы с SQLite установим специальный инструмент - [DB Browser for SQLite](#).

Откроем DB Browser for SQLite и для создания новой базы данных нажмем на кнопку **New Database**. В открывшемся окне выберем место расположения файла базы данных и укажем для нее имя, например, **friends2**.

Далее нам будет предложено добавить в базу данных таблицы. Возьмем проект из прошлой темы. В нем была модель Friend:

```
using SQLite;

namespace SQLiteApp
{
    [Table("Friends")]
    public class Friend
    {
        [PrimaryKey, AutoIncrement, Column("_id")]
        public int Id { get; set; }

        public string Name { get; set; }
        public string Email { get; set; }
        public string Phone { get; set; }
    }
}
```

Для хранения объектов данной модели определим в базе данных новую таблицу Users и добавим четыре столбца \_id, Name, Email, Phone, как на скриншоте:

**Edit table definition**

Table:

▼ Advanced

Fields

Add field
 Remove field
 ▲ Move field up
 ▼ Move field down

Name	Type	Not	PK	AI	U	Default
_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Name	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Email	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Phone	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```

1 CREATE TABLE `Friends` (
2   ` _id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
3   `Name` TEXT,
4   `Email` TEXT,
5   `Phone` TEXT
6 );
  
```

OK Cancel

Эта таблица имеет столбцы, которые соответствуют свойствам класса Friend.

Можно тут же добавить некоторые начальные данные в таблицу.

После определения бд установим ее название в качестве используемого в классе App в главном проекте приложения:

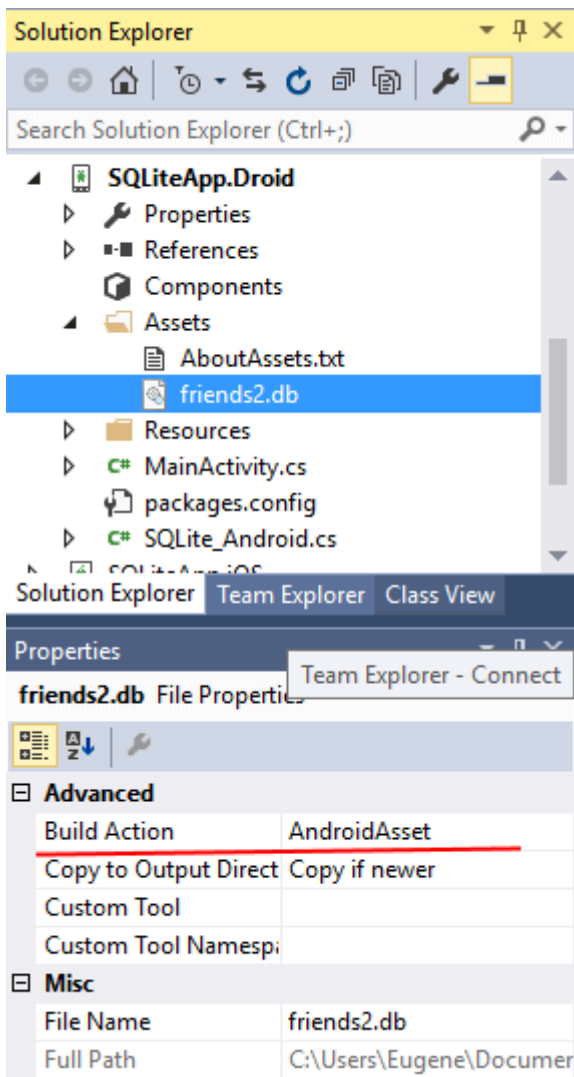
```

public class App : Application
{
    public const string DATABASE_NAME= "friends2.db";
    // остальное содержимое класса
}
  
```

После создания базы данных ее надо добавить в проекты для каждой ОС, которые мы будем использовать.

## Добавление в проект для Android

Добавим базу данных в проект для Android в папку **Assets**:



При добавлении у файла базы данных в панели свойств установим для ее опции **Build Action** значение **AndroidAsset**.

Далее изменим файл **SQLite\_Android.cs**:

```
using System;
using SQLiteApp.Droid;
using System.IO;
using Xamarin.Forms;

[assembly: Dependency(typeof(SQLite_Android))]
namespace SQLiteApp.Droid
{
    public class SQLite_Android : ISQLite
    {
        public SQLite_Android() { }
        public string GetDatabasePath(string sqliteFilename)
        {
            string documentsPath =
                Environment.GetFolderPath(Environment.SpecialFolder.Personal);
            var path = Path.Combine(documentsPath, sqliteFilename);
            // копирование файла из папки Assets по пути path
            if (!File.Exists(path))
            {
                var dbAssetStream = Forms.Context.Assets.Open(sqliteFilename);

                var dbFileStream = new System.IO.FileStream(path,
                    System.IO.FileMode.OpenOrCreate);
                var buffer = new byte[1024];
```

```
        int b = buffer.Length;
        int length;

        while ((length = dbAssetStream.Read(buffer, 0, b)) > 0)
        {
            dbFileStream.Write(buffer, 0, length);
        }

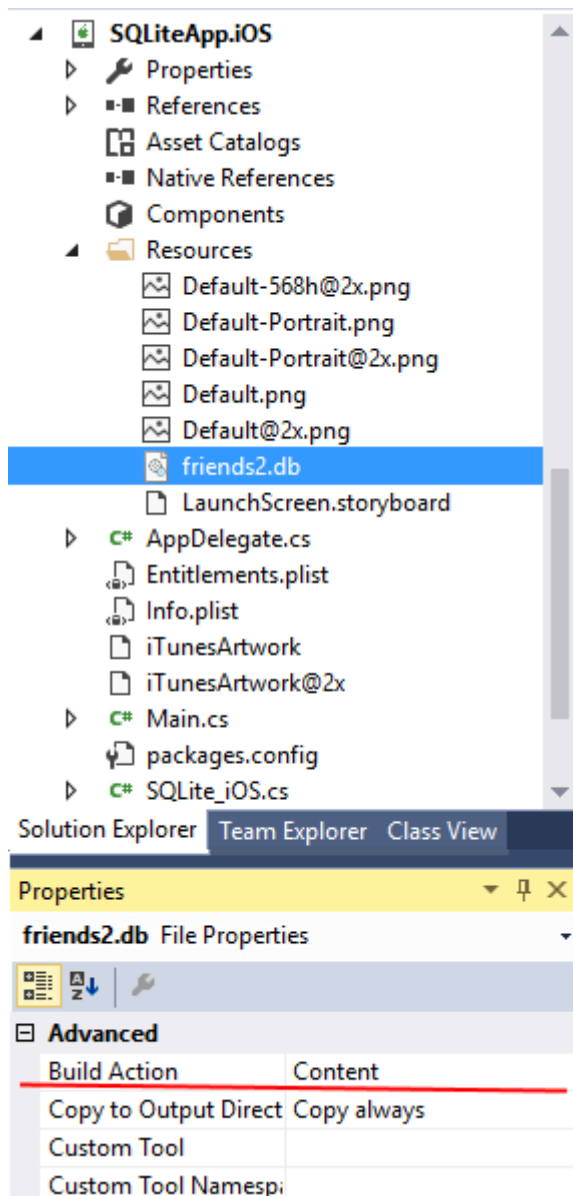
        dbFileStream.Flush();
        dbFileStream.Close();
        dbAssetStream.Close();
    }

    return path;
}
}
```

Этот класс по прежнему возвращает объект подключения SQLiteConnection, только перед созданием подключения происходит копирование файла базы данных из папки Assets по указанному пути в переменной path.

## Добавление в проект для iOS

Добавим базу данных в проект для iOS в папку **Resources**:



При добавлении у файла базы данных в панели свойств установим для ее опции **Build Action** значение **Content**.

И также изменим файл **SQLite\_iOS.cs**:

```
using System;
using Xamarin.Forms;
using System.IO;
using SQLiteApp.iOS;

[assembly: Dependency(typeof(SQLite_iOS))]
namespace SQLiteApp.iOS
{
    public class SQLite_iOS : ISQLite
    {
        public SQLite_iOS() { }
        public string GetDatabasePath(string sqliteFilename)
        {
            // определяем путь к бд
            string documentsPath =
Environment.GetFolderPath(Environment.SpecialFolder.Personal);
            string libraryPath = Path.Combine(documentsPath, "..", "Library"); // папка
библиотеки
            var path = Path.Combine(libraryPath, sqliteFilename);
        }
    }
}
```

```

        if (!File.Exists(path))
        {
            File.Copy(sqliteFilename, path);
        }

        return path;
    }
}

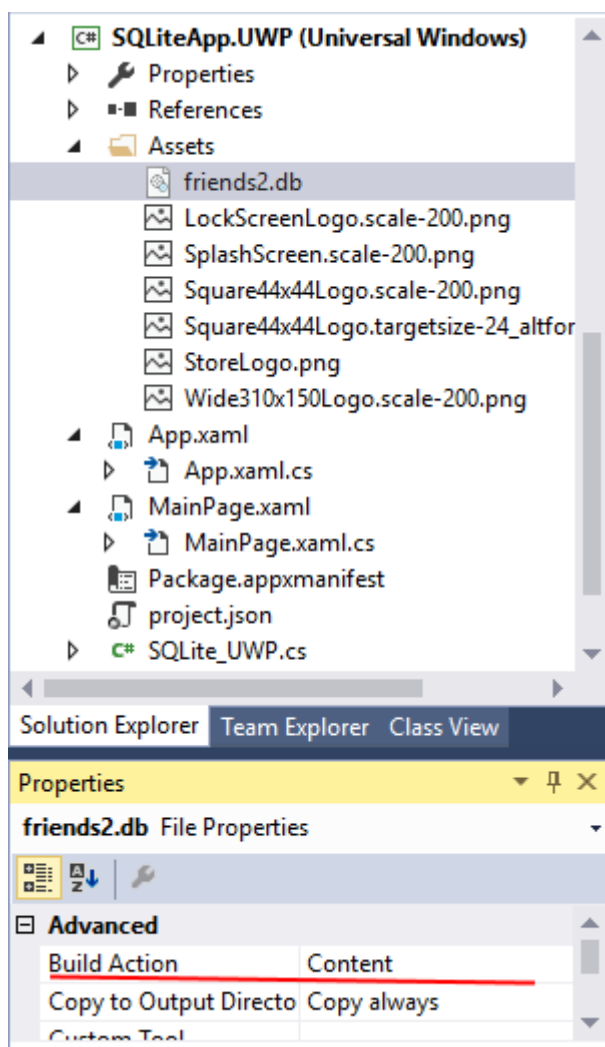
```

Здесь опять же перед созданием подключения происходит копирование файла базы данных по указанному пути.

## Добавление в проект для UWP

Настройка использования бд в проекте для UWP имеет ряд особенностей в силу применяемого асинхронного API в UWP для работы с файлами.

Также добавим базу данных в проект для UWP в папку **Assets**:



При добавлении у файла базы данных в панели свойств установим для ее опции **Build Action** значение **Content**.

Для копирования файла базы данных в проекте для UWP используем класс App, который срабатывает при старте приложения. Итак, в проекте для UWP в файле **App.xaml.cs** добавим в начало метода OnLaunched копирование базы данных и отметим метод как асинхронный с помощью ключевого слова **async**:

```
protected override async void OnLaunched(LaunchActivatedEventArgs e)
{
    // получаем имя базы данных
    string databaseName = SQLiteApp.App.DATABASE_NAME;
    if (await ApplicationData.Current.LocalFolder.TryGetItemAsync(databaseName) == null)
    {
        // копируем бд из папки Assets
        StorageFile databaseFile =
            await Package.Current.InstalledLocation.GetFileAsync($"Assets\\{databaseName}");
        await databaseFile.CopyAsync(ApplicationData.Current.LocalFolder);
    }

    // .....остальное содержимое метода
}
```

Здесь мы вынуждены обращаться к классу App из главного проекта приложения для получения имени базы данных.

В итоге вне зависимости от того, какой мы проект запустим, в каждом проекте будет происходить копирование и обращение к уже существующей базе данных.



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

[G+](#)

8 Комментариев [metanit.com](#)

[Войти](#)

[Рекомендовать](#) [Поделиться](#)

[Лучшее в начале](#)



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

или через DISQUS

**Виктор Викторов** • 5 месяцев назад

Возможно ли как-нибудь организовать внешние ключи? При создании таблицы в DB Browser for SQLite это возможно, но как тогда работать с этим в приложении?

^ | ▾ • Ответить • Поделиться ›

**Богдан Нестеренко** • 6 месяцев назад

Спасибо за материал. Не могли бы вы дополнить или добавить информацию как создать отношение один ко многим. И как работать с двумя или тремя связанными таблицами.

^ | ▾ • Ответить • Поделиться ›

**Ilya egorov** • год назад

Спасибо за Вашу работу.

Метод TryGetItemAsync - будет работать только для Windows 8.1. Для проектов Windows 8 или WP8.1 XAML данного метода нет. Так же для проектов WP8.1 XAML нет метода File.Exists. Поэтому чтоб проверить существование БД надо "вручную" отлавливать исключение. Например:

```
string databaseName = SQLiteApp.App.DATABASE_NAME;
```

```
try
```

```
{
```

```
StorageFile file = await ApplicationData.Current.LocalFolder.GetFilesAsync(databaseName);
```

```
Stream fileStream = await file.OpenStreamForReadAsync();
```

```
fileStream.Dispose();
```

```
}
```

```
catch (FileNotFoundException)
```

```
{
```

```
StorageFile databaseFile =
```

```
await Package.Current.InstalledLocation.GetFilesAsync($"@\"Assets\\{databaseName}\"");
```

```
await databaseFile.CopyAsync(ApplicationData.Current.LocalFolder);
```

```
}
```

^ | ▾ • Ответить • Поделиться ›

**Metanit** Модератор ➔ Ilya egorov • год назад

Весь приведенный код для Windows - справедлив только для UWP. Для Win8/8.1 возможно что-то будет отличаться

^ | ▾ • Ответить • Поделиться ›

**Vadim Cherny** • год назад

Здравствуйте. Разрабатываю приложение под Android. Вот проблема. Как можно получить БД с Андроида на папку от Windows.?

^ | ▾ • Ответить • Поделиться ›

**Metanit** Модератор ➔ Vadim Cherny • год назад

только скопировать из проекта для андроида в проект для windows, больше никак

^ | ▾ • Ответить • Поделиться ›





**Vadim Cherny** → Metanit • год назад

Вы меня не совсем поняли. Моя бд находится по пути /data/data ....test.db в Android проекте. Мне нужно вытащить ее оттуда на основную Windows. Не на проект для винфона, а на основную ОС

^ | v • Ответить • Поделиться ›



**Metanit** Модератор → Vadim Cherny • год назад

ну откройте проект в проводнике, скопируйте базу данных и делайте с ней все что хотите

^ | v • Ответить • Поделиться ›

#### ТАКЖЕ НА METANIT.COM

### Kotlin | Руководство

3 комментариев • 3 месяца назад



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

### Angular в ASP.NET Core | CRUD и маршрутизация. Часть 1

1 комментарий • 3 месяца назад



**Vadim Prokopchuk** — Добрый вечер.Попытался в текущий пример добавить стилизацию, но столкнулся с

### Django | Статические файлы

1 комментарий • 7 дней назад



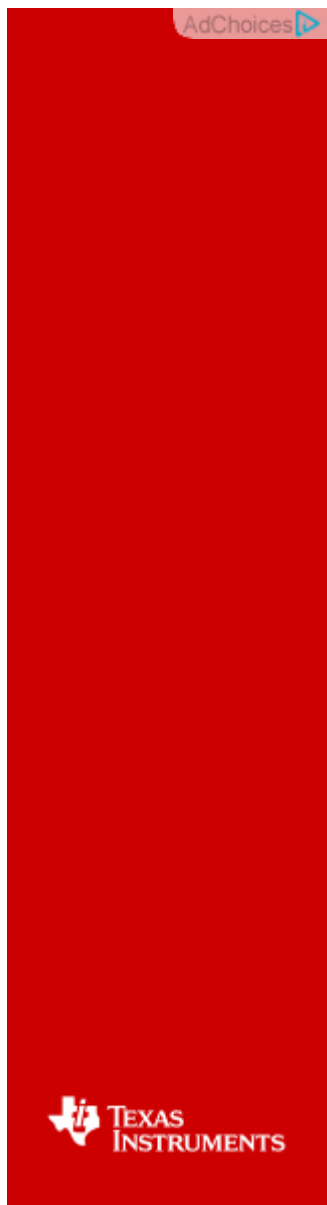
**Pavel Korolecky** — Отличный, а самое главное — очень актуальный материал! По свежей Django колоссально мало

### Angular в ASP.NET Core | Маршрутизация

6 комментариев • 3 месяца назад



**Эдуард Макаров** — Подскажите, пожалуйста, как делать отладку Angular в Visual Studio 2017? В VS точки останова не срабатывают,



---

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

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