عنوان: **ساخت دیتابیس sqlite با EF6 Code First**
نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۸/۰۴ ۲۰:۲۵
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, SQLite

تا نسخه EF6 و minorهای آن به دلیل عدم پشتیبانی داریور sqlite از migration، ساخت دیتابیس با code first ممکن نیست برای همین مجبور هستند از پیاده سازی‌های خودشان و موجود بودن دیتابیس از قبل با استفاده از EF با آن کار کنند که یکی از مثال‌های آن در این آدرس قرار دارد و سعی دارد کلاسی مشابه sqlitehelper در اندروید که کار ساخت دیتابیس و مدیریت نسخه را دارد بسازد و از آن استفاده کند. البته در EF7 این مشکل حل شده است و تیم دات نت تمهیداتی را برای آن اندیشیده‌اند. در این نوشتار قصد داریم با استفاده از یک کتابخانه که توسط آقای مارک سالین نوشته شده است کار ساخت دیتابیس را آسانتر کنیم. این کتابخانه که با دات نت 4 به بعد کار میکند خیلی راحت می‌تواند دیتابیس شما را به روش Code First ایجاد کند.

در حال حاضر این کتابخانه از مفاهیم زیر پشتیبانی می‌کند:

تبدیل کلاس به جدول با پشتیبانی از خصوصیت Table
تبدیل پراپرتی‌ها به ستون با پشتیبانی از خصوصیت هایی چون
Column,Key,MaxLength,Required,Notmapped,DatabaseGenerated,Index
پشتیبانی از primarykey و کلیدهای ترکیبی
کلید خارجی و روابط یک به چند و پشتیبانی از cascade on delete
فیلد غیر نال

برای شروع ابتدا کتابخانه مورد نظر را از Nuget با دستور زیر دریافت کنید:

```
Install-Package SQLite.CodeFirst
```

خود این دستور باعث می‌شود که وابستگی‌هایش از قبیل sqlite providerها نیز دریافت گردند.
solution من شامل سه پروژه است یکی برای مدل‌ها که شامل کلاس‌های زیر برای تهیه یک دفترچه تلفن ساده است:

Person

```
public class Person
    {
        public int Id { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }

        public virtual ICollection<PhoneBook> Numbers { get; set; }

    }
```

PhoneBook

```
public class PhoneBook
    {
        public int Id { get; set; }

        public string Field{ get; set; }

        public string Number { get; set; }

        public virtual Person Person { get; set; }
    }
```

پروژه بعدی به نام سرویس که جهت پیاده سازی کلاس‌های EF است و دیگری هم یک پروژه‌ی WPF جهت تست برنامه.
در پروژه‌ی سرویس ما یک کلاس به نام Context داریم که مفاهیم مربوط به پیاده سازی Context در آن انجام شده است:

```
public class Context:DbContext
    {
        public Context():base("constr")
        {
        }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();
            var initializer = new InitialDb(modelBuilder);
            Database.SetInitializer(initializer);
        }

        public DbSet<PhoneBook> PhoneBook { get; set; }
        public DbSet<Person> Persons { get; set; }
    }
```

تا به الان چیز جدیدی نداشتیم و همه چیز طبق روال صورت گرفته است؛ ولی دو نکته‌ی مهم در این کد نهفته است:

اول اینکه در سطر اول متد بازنویسی شده onModelCreating، قرارداد مربوط به نامگذاری جداول را حذف می‌کنیم چرا که در صورت نبودن این خط، اسامی که کلاس sqllite برای آن در نظر خواهد گرفت با اسامی که برای انجام عملیات CURD استفاده می‌شوند متفاوت خواهد بود. برای مثال برای Person جدولی به اسم People خواهد ساخت ولی برای درج، آن را در جدول Person انجام می‌دهد که به خاطر نبودن جدول با خطای چنین جدولی موجود نیست روبرو می‌شویم.

نکته‌ی دوم اینکه در همین کلاس Context ما یک پیاده سازی جدید بر روی کلاس InitialDb داشته ایم که در زیر نمونه کد آن را می‌بینید:

```
public class InitialDb:SQLite.CodeFirst.SqliteCreateDatabaseIfNotExists<Context>
    {
        public InitialDb(DbModelBuilder modelBuilder) : base(modelBuilder)
        {
        }

        protected override void Seed(Context context)
        {
            var person = new Person()
            {
                FirstName = "ali",
                LastName = "yeganeh",
                Numbers = new List<PhoneBook>()
                {
                    new PhoneBook()
                    {
                        Field = "Work",
                        Number = "031551234"
                    },
                    new PhoneBook()
                    {
                        Field = "Mobile",
                        Number = "09123456789"
                    },
                    new PhoneBook()
                    {
                        Field = "Home",
                        Number = "031554321"
                    }
                }
            };

            context.Persons.Add(person);
            base.Seed(context);
        }
    }
```

در این کد کلاس InitialDb از کلاس SqliteCreateDatabaseIfNotExists ارث بری کرده‌است و متد seed آن را هم بازنویسی کرده‌ایم. کلاس SqliteCreateDatabaseIfNotExists برای زمانی کاربر دارد که اگر دیتابیس موجود نیست آن را ایجاد کند، در غیر اینصورت خیر. به غیر از آن، کلاس دیگری به نام SqliteDropCreateDatabaseAlways هم وجود دارد که با هر بار اجرا، جداول قبلی را حذف و مجددا آن‌ها را ایجاد میکند.
سپس در پروژه‌ی اصلی WPF در فایل AppConfig رشته اتصالی مورد نظر را وارد نمایید:

```
<connectionStrings>
    <add name="constr" connectionString="data source=.\phonebook.sqlite;foreign keys=true"
providerName="System.Data.SQLite" />
  </connectionStrings>
```

نکته‌ی مهم اینکه با افزودن کتابخانه از طریق nuget فایل app.config به روز می‌شود؛ ولی به نظر می‌رسد که تنظیمات به درستی انجام نمی‌شوند. در صورتیکه به مشکل زیر برخوردید و نتوانستید برنامه را اجرا کنید، کد زیر را که قسمتی از فایل app.config است، مطالعه فرمایید و موارد مربوط به آن را اصلاح کنید:

خطا:

```
The ADO.NET provider with invariant name 'System.Data.SQLite' is either not registered in the machine
or application config file, or could not be loaded
```

قسمتی از فایل app.config:

```
<entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory,
EntityFramework">
      <parameters>
        <parameter value="mssqllocaldb" />
      </parameters>
    </defaultConnectionFactory>
    <providers>
         <provider invariantName="System.Data.SqlClient"
type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
      <provider invariantName="System.Data.SQLite" type="System.Data.SQLite.EF6.SQLiteProviderServices,
System.Data.SQLite.EF6" />

    </providers>
  </entityFramework>
  <system.data>
    <DbProviderFactories>
      <remove invariant="System.Data.SQLite.EF6" />
      <remove invariant="System.Data.SQLite" />
       <add name="SQLite Data Provider" invariant="System.Data.SQLite" description=".Net Framework Data
Provider for SQLite" type="System.Data.SQLite.SQLiteFactory, System.Data.SQLite" />
    </DbProviderFactories>
  </system.data>
```

کد Load پروژه WPF:

```
public MainWindow()
        {
            InitializeComponent();
           var context=new Context();

            var list= context.Persons.ToList();

            var s = "";

            foreach (var person in list)
            {
                s += person.FirstName + " " + person.LastName +
            " has these numbers:" + Environment.NewLine;

                foreach (var number in person.Numbers)
                {
                    s += number.Field + " : " + number.Number + Environment.NewLine;
                }
```

```
                s += Environment.NewLine;
            }
            MessageBox.Show(s);

        }
```

دانلود مثال