



## Строка подключения

Последнее обновление: 31.10.2015

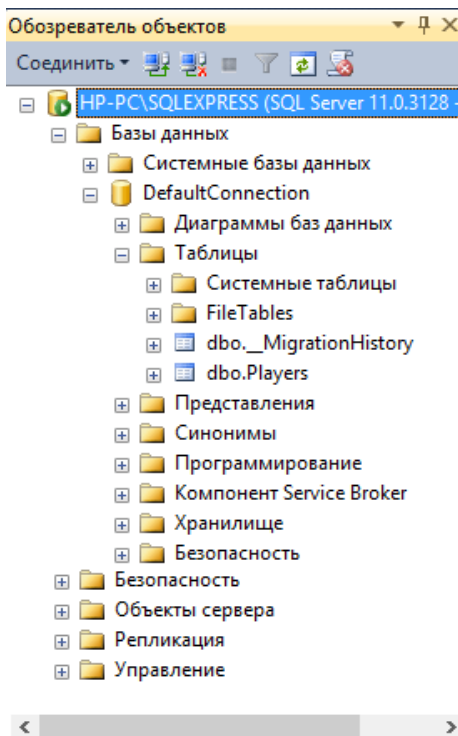


В предыдущей теме использовался следующий контекст данных:

```
class SoccerContext : DbContext
{
    public SoccerContext()
        : base("DefaultConnection")
    { }

    public DbSet<Player> Players { get; set; }
}
```

В конструктор базового класса передается наименование подключения - DefaultConnection. И так как в данном случае использовался подход Code First, то при обращении к базе данных, если ее нет, создавалась новая база данных с названием DefaultConnection на MS SQL Server. А после создания все запросы также шли к этой базе данных. Мы можем подключиться к MS SQL Server через Visual Studio или воспользоваться специальной средой **SQL Server Management Studio**, чтобы ее увидеть:



Но в реальности может потребоваться динамически изменять местоположение или название базы данных, к которой идет подключения. В этом случае нам надо указать строку подключения. По умолчанию Visual Studio добавляет в проекты файл конфигурации. В проектах для десктопных приложений (Windows Forms, консольные приложения и т.д.) этот файл называется **App.config**, в проектах веб-приложений (ASP.NET) это файл **Web.config**. Но вне зависимости от типа проекта

файл конфигурации имеет определенную структуру и элементы, среди которых также есть также элемент `connectionStrings`, определяющий строки подключения.

Итак, что делать, если мы хотим определить базу данных со случайным именем, которая будет находиться на MS SQL Server? Добавим в файл `App.config` перед закрывающим тегом `configuration` следующий код:

```
<connectionStrings>
  <add name="DefaultConnection" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=Players;Integrated
Security=True"
      providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Элемент `add` добавляет в секцию `connectionStrings` определение строки подключения к БД. Таких элементов `add` можно определить несколько, если мы планируем использовать различные подключения.

Атрибут `name="DefaultConnection"` указывает на название подключения. Имя должно согласоваться с контекстом данных, который использует данное подключение. Так, в конструктор базового класса (`base("DefaultConnection")`) мы передаем название `DefaultConnection`, поэтому и подключение носит такое же название. Но если бы определили бы контекст без вызова конструктора базового класса:

```
class SoccerContext : DbContext
{
    public SoccerContext()
    { }
    public DbSet<Player> Players { get; set; }
}
```

В этом случае название подключения должно было бы отражать название контекста: `name="SoccerContext"`

Следующий элемент подключения определяет параметры строки подключения. Этот элемент разбивается на несколько частей:

- **Data Source:** название сервера. Для MS SQL Express этот параметр имеет значение `.\SQLEXPRESS`
- **Initial Catalog:** название каталога базы данных. В данном случае `Players`, поэтому при использовании подхода Code First на сервере будет создаваться база данных `Players.mdf`
- **Integrated Security:** устанавливает проверку подлинности

И последний элемент устанавливает провайдер: `providerName="System.Data.SqlClient"`

И таким образом, в процессе работы приложения будет создана (если еще не существует) и использоваться база данных `Players`.

## Строка подключения в Model First и Database First

При использовании подходов Model First и Database First строка подключения выглядит иначе. Например:

```
<connectionStrings>
  <add name="persondbEntities" providerName="System.Data.EntityClient"
      connectionString="metadata=res://*/Person.csdl|res://*/Person.ssdl|res://*/Person.msl;provider=System.Data.SqlClient;
      provider connection string="data source=HP-PC\SQLEXPRESS;
      initial catalog=persondb;integrated security=True;
      MultipleActiveResultSets=True;App=EntityFramework"" />
</connectionStrings>
```

Здесь для нас важны следующие параметры строки подключения:

- параметр `metadata`, содержащий метаданные модели ( так как в данном случае модель называется `Person`, то метаданные включают ресурсы `Person.csdl`, `Person.ssdl`, `Person.msl`)
- параметр `data source` также устанавливает сервер MS SQL
- а параметр `initial catalog` также устанавливает каталог базы данных

И в случае изменения базы данных, ее положения, наименования, или использования другой модели, нам надо соответственно изменить эти параметры.

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



---

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

Контакты для связи: metanit22@mail.ru

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

