**[# Code First 配置](https://www.cnblogs.com/alunchen/p/6903753.html)**

**前言**

所谓code first，是代码优先，而不是传统的数据库优先。

code first的好处多多，可不用考虑数据库，只考虑面向对象。下面我们介绍一下code first的配置。

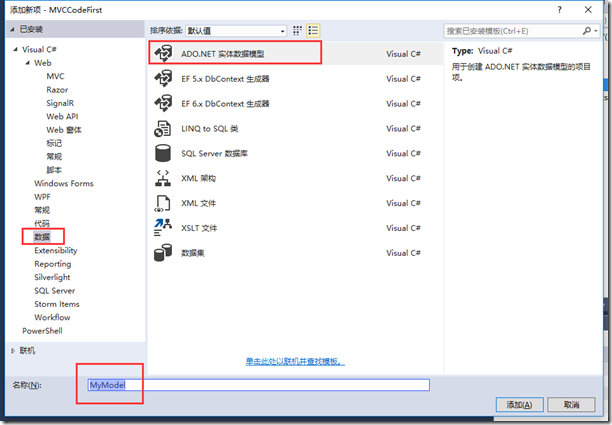
下一篇文章地址：[C# Code First 配置（二）](http://www.cnblogs.com/alunchen/p/7737169.html)

**开始**

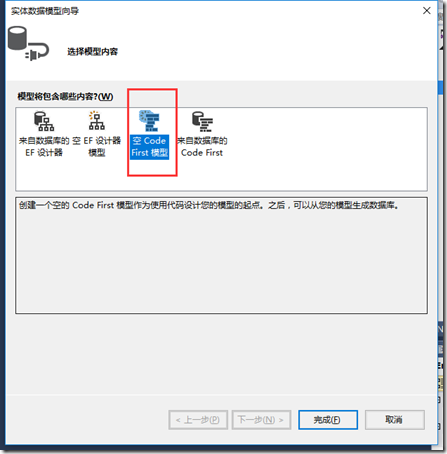
我们这里以建立一个MVC的Code first为例，名字为MVCCodeFirst。

**1）建立模型**

建立一个数据的新建项，然后选择ADO.NET实体数据模型，名字为MyModel

[](http://images2015.cnblogs.com/blog/914305/201705/914305-20170525144534122-2017903330.png)

然后选择Code First模型，空Code First模型，你也可以选择来自数据库的CodeFirst。我们这里选择空CodeFirst模型

[](http://images2015.cnblogs.com/blog/914305/201705/914305-20170525144551794-1858647031.png)

确定之后，模型已经建立了

[](http://images2015.cnblogs.com/blog/914305/201705/914305-20170525144553654-1555372403.png)

**2）建立实体**

我们建立一个实体School，有一个主键与学校名字

[复制代码](javascript:void(0);)

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace MVCCodeFirst.Entity

{

public class School

{

[Key]

public long Key { get; set; }

public string Name { get; set; }

}

}

[复制代码](javascript:void(0);)

然后把School添加到DbContext的上下文中。

[复制代码](javascript:void(0);)

using System.ComponentModel.DataAnnotations;

namespace MVCCodeFirst.Entity

{

using System;

using System.Data.Entity;

using System.Linq;

public class MyModel : DbContext

{

public MyModel()

: base("name=MyModel")

{

}

//我们添加的School

public virtual DbSet<School> Schools { get; set; }

}

}

[复制代码](javascript:void(0);)

**3）启动 Enable-Migrations**

上面已经建立了表的实体，Code First要创建表，我们要通过命令行来自动生成SQL语句，所以要启动这个功能。

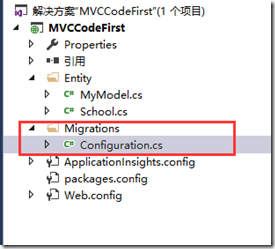
在【程序包管理器控制台】中，启动自动生成SQL语句

*Enable-Migrations*

如下：

[](http://images2015.cnblogs.com/blog/914305/201705/914305-20170525144600169-1434103630.png)

完成之后，你在代码里面可以看到，自动添加了一个文件夹为Migrations，并且类为Configuration.cs

[](http://images2015.cnblogs.com/blog/914305/201705/914305-20170525144602732-881215341.png)

打开可以看到，这里主要用户初识化一些默认的数据

[复制代码](javascript:void(0);)

namespace MVCCodeFirst.Migrations

{

using System;

using System.Data.Entity;

using System.Data.Entity.Migrations;

using System.Linq;

internal sealed class Configuration : DbMigrationsConfiguration<MVCCodeFirst.Entity.MyModel>

{

public Configuration()

{

AutomaticMigrationsEnabled = false;

}

protected override void Seed(MVCCodeFirst.Entity.MyModel context)

{

//初始化数据

}

}

}

[复制代码](javascript:void(0);)

例如，我们可以初识化学校的数据。这个初始化数据是在输入命令update-database的时候才会调用

//初始化数据格式

context.Schools.AddOrUpdate(new School() {Key = 1, Name = "望牛墩中学"});

context.SaveChanges();

**4)配置数据库**

打开web.config，可以看到数据库连接现在是

<connectionStrings>

<add name="MyModel" connectionString="data source=(LocalDb)\MSSQLLocalDB;initial catalog=MVCCodeFirst.Entity.MyModel;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework" providerName="System.Data.SqlClient" />

</connectionStrings>

我们修改成本地连接

<add name="MyModel" connectionString="Server=localhost; Database=MyModel; Trusted\_Connection=True;" providerName="System.Data.SqlClient" />

**5）用命令创建数据库、表**

初识化数据

*add-migration init*

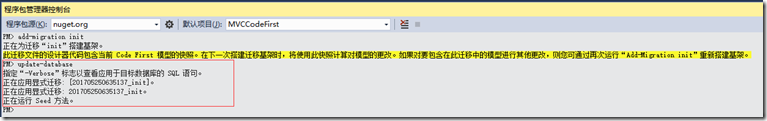
[](http://images2015.cnblogs.com/blog/914305/201705/914305-20170525144610997-9261949.png)

就会生成创建的SQL类在Migrations文件夹下

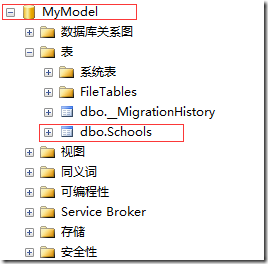
[image](http://images2015.cnblogs.com/blog/914305/201705/914305-20170525144624747-2084630532.png)

创建数据库、表。在命令中输入：

*update-database*

[](http://images2015.cnblogs.com/blog/914305/201705/914305-20170525144627200-845132029.png)

我们可以看到本地数据库中有创建了此数据库与表：

[](http://images2015.cnblogs.com/blog/914305/201705/914305-20170525144633575-1657030140.png)

**3.其他**

1）果想指定表的名字，可以在实体类上用注解Table，例如

[Table("School")]

public class School

2)主键注解 [Key]

[Key]

public long Id { get; set; }

3)如果想创建一个自增长的主键

在DB上下文实体中，继承OnModelCreating方法，并且实现实体的主键自增长

[复制代码](javascript:void(0);)

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace MVCCodeFirst.Entity

{

using System;

using System.Data.Entity;

using System.Linq;

public class MyModel : DbContext

{

//您的上下文已配置为从您的应用程序的配置文件(App.config 或 Web.config)

//使用“MyModel”连接字符串。默认情况下，此连接字符串针对您的 LocalDb 实例上的

//“MVCCodeFirst.Entity.MyModel”数据库。

//

//如果您想要针对其他数据库和/或数据库提供程序，请在应用程序配置文件中修改“MyModel”

//连接字符串。

public MyModel()

: base("name=MyModel")

{

}

//我们添加的Schools

public virtual DbSet<School> Schools { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

//自增长主键

modelBuilder.Entity<School>()

.Property(o => o.Id)

.HasDatabaseGeneratedOption(DatabaseGeneratedOption.Identity);

}

}

}

[复制代码](javascript:void(0);)