# [Configure Domain Classes（配置领域类）【EF Code-First 系列】](http://www.cnblogs.com/caofangsheng/p/5022347.html)

在前面的部分中，我们学习了Code-First默认约定，Code-First使用默认的约定，根据你的领域类，然后生成概念模型。

Code-First模式，发起了一种编程模式：约定大于配置。这也就是说，当你需要的时候，你可以重写这些约定，通过配置你的领域类。这里有两种方式来配置你的领域类实体：

1. DataAnnotations（数据注解）
2. Fluent API（姑且翻译为：流畅API）

### **数据注解：**

数据注解是基于配置的简单特性，你可以应用到你的领域类或者其属性中。你可能会发现大多数的特性都在这个命令空间下面：

[System.ComponentModel.DataAnnotations](http://msdn.microsoft.com/en-us/library/system.componentmodel.dataannotations(v=vs.110).aspx" \t "http://www.cnblogs.com/caofangsheng/p/_blank)， 然而，数据注解提供的特性仅仅是Fluent API配置的一部分子集而已，所以，如果你，在数据注解中，没有找到的属性，可以使用Fluent API来配置。

下面的例子中，是一个简单使用数据注解的例子：

using System;using System.Collections.Generic;using System.ComponentModel.DataAnnotations;using System.ComponentModel.DataAnnotations.Schema;using System.Linq;using System.Text;using System.Threading.Tasks;

namespace EF2

{

[Table("StudentInfo")]

public class Student

{

[Key]

public int SID { get; set; }

[Column("Name",TypeName="ntext")]

[MaxLength(20)]

public string StudentName { get; set; }

[NotMapped()]

public int? Age { get; set; }

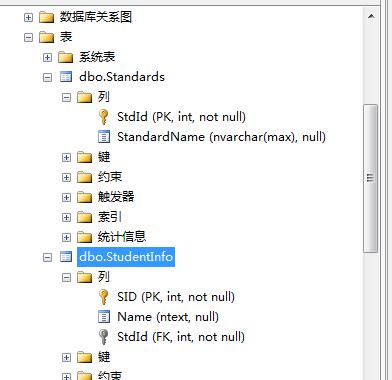
public int StdId { get; set; }

[ForeignKey("StdId")]

public virtual Standard Standard { get; set; }

}

}



生成的数据库是这样的。

### **Fluent API**

Fluent API配置，利用EF从你的实体类中构建模型，你可以注入这个配置，通过重写DbContext类的“OnModelCreating”方法，例如下面的：

using System;using System.Collections.Generic;using System.Data.Entity;using System.Linq;using System.Text;using System.Threading.Tasks;

namespace EF2

{

public class DbContextClass:DbContext

{

public DbContextClass() : base("Constr") { }

public DbSet<Student> Students { get; set; }

public DbSet<Standard> Standards { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{  
 //这里面配置领域类实体，通过使用Dluent API

Database.SetInitializer(new DropCreateDatabaseIfModelChanges<DbContextClass>());

base.OnModelCreating(modelBuilder);

}

}

}

You can use modelBuilder, which is an object of DbModelBuilder class, to configure domain classes.

你可以使用modelBuider,他是DBModelBuilder类的对象，用来配置领域类的。

Let's see DataAnnotation and Fluent API in detail in the next chapter.

我们在后面的章节中将会详细的了解数据注解和Fluent API。