# EntityFramework

导航属性：

导航属性通常被定义成virtual，这样我们便可以利用EntityFramework的优势——慢加载（“慢加载”将在“读取相关数据”一篇中介绍）。如果一个导航属性是包含多个实体的，它必定是ICollection，无论是“多对多”还是“多对一”的情况下。

EF实体模型是要定义为Domain还是数据库模型。EF模型跟领域模型区别？

按照Nop Domain模型定义的正式EF模型。

Q:默认情况下，EntityFramework把“ID”或者是“类名ID”形式翻译理解为主键。如何修改这个默认规则？

实体集 –》表

实体 -》表中的一条记录

Q:如何避免生成数据表名复数化：

A：public class SchoolContext : DbContext  
 {  
 public DbSet<Student> Students { get; set; }  
 public DbSet<Enrollment> Enrollments { get; set; }  
 public DbSet<Course> Courses { get; set; }  
  
 protected override void OnModelCreating(DbModelBuilder modelBuilder)  
 {  
 modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();  
 }  
 }

EntityFramework将会寻找和连接字符串同名的数据库上下文类进行创建

## 自动创建数据库：

数据模型实体与数据表不匹配时候创建：

程序运行时候创建：

public class SchoolInitializer : DropCreateDatabaseIfModelChanges<SchoolContext>  
 {  
 protected override void Seed(SchoolContext context)  
 {  
 var students = new List<Student>  
 {  
 new Student { FirstMidName = "Carson", LastName = "Alexander", EnrollmentDate = DateTime.Parse("2005-09-01") }  
 };  
 students.ForEach(s => context.Students.Add(s));  
 context.SaveChanges();  
  
 var courses = new List<Course>  
 {  
 new Course { Title = "Chemistry", Credits = 3, },  
 };  
 courses.ForEach(s => context.Courses.Add(s));  
 context.SaveChanges();  
  
 var enrollments = new List<Enrollment>  
 {  
 new Enrollment { StudentID = 1, CourseID = 1, Grade = 1 },  
 };  
 enrollments.ForEach(s => context.Enrollments.Add(s));  
 context.SaveChanges();  
 }  
 }

2）在“Application\_Start”方法中，调用EntityFramework一个方法运行初始化代码：

Database.SetInitializer<SchoolContext>(new SchoolInitializer());

## 一些默认转化

* 实体类复数形式的名字被用于表。
* 实体类公共属性被当成表字段。
* 实体类带有ID或者是“类名ID”的公共属性被当成是主键。
* Entity Framework通过寻找和你数据上下文类同名的那个名称从而确定你的连接字符串（示例中： SchoolContext）。

这些转换可以被重写（比如你指定表不能被复数化）