public class Order

{

public int OrderId { get; set; }

public DateTime OrderDate { get; set; }

public string Username { get; set; }

[Required]

**[StringLength(160)]**

**[StringLength(160, MinimumLength=3)]**

public string FirstName { get; set; }

[Required]

**[Required(ErrorMessage="Your {0} is required.")]**

[StringLength(160, ErrorMessage="{0} is too long.")]

[MaxWords(10, ErrorMessage="There are too many words in {0}")]

public string LastName { get; set; }

public string Address { get; set; }

[Required]

[DataType(DataType.Password)]

**[Display(Name="Password")]**

public string Password { get; set; }

public string City { get; set; }

public string State { get; set; }

public string PostalCode { get; set; }

public string Country { get; set; }

public string Phone { get; set; }

[RegularExpression(@"[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}",

**ErrorMessage="Email doesn't look like a valid email address.")]**

public string Email { get; set; }

**[Compare("Email")]**

public string EmailConfirm { get; set; }

public decimal Total { get; set; }

[Range(35,44)

public int Age{get;set;}

[Remote("CheckUserName", "Account")]

//CheckUserName is Action Method

public JsonResult CheckUserName(string username)

{

var result = Membership.FindUsersByName(username).Count == 0;

return Json(result, JsonRequestBehavior.AllowGet);

}

public string UserName { get; set; }

[DisplayFormat(ApplyFormatInEditMode=true, DataFormatString="{0:c}")]

public decimal Total { get; set; }

public List<OrderDetail> OrderDetails { get; set; }

}

public class Order : IValidatableObject

{

public IEnumerable<ValidationResult> Validate(

ValidationContext validationContext)

{

if (

)

{

yield return new ValidationResult("The last name has too many words!",

new []{"LastName"});

}

}

// rest of Order implementation and properties

// ...

}

[Required]

[StringLength(160)]

**[Display(Name="Last Name)]**

[MaxWords(10, ErrorMessage="There are too many words in {0}")]

[Order]

public string LastName { get; set; }

[ScaffoldColumn(false)]

public string Username { get; set; }

[Required]

[StringLength(160, MinimumLength=3)]

**[Display(Name="First Name", Order=15000)]**

public string FirstName { get; set; }

1. **public class Category**
2. **{**
3. **public int CategoryID { get; set; }**
4. **public string Name { get; set; }**
5. **}**
7. **public class Product**
8. **{**
9. **public int ProductID { get; set; }**
10. **public string Name { get; set; }**
11. **public decimal Price { get; set; }**
12. **public int CatID { get; set; }**
14. **public virtual Category Category { get; set; }**
15. **}**
16. **public class DataContext : DbContext**
17. **{**
18. **public DataContext():base("DefaultConnection"){ }**
20. **protected override void OnModelCreating(DbModelBuilder modelBuilder)**
21. **{**
22. **modelBuilder.Entity<Category>().HasKey(c => c.CategoryID);**
23. **modelBuilder.Entity<Category>().Property(p=> p.Name).HasColumnType("VARCHAR").IsRequired().HasMaxLength(50);**
24. **modelBuilder.Entity<Product>().HasKey(p => p.ProductID);**
25. **modelBuilder.Entity<Product>().Property(p=> p.Name).HasColumnType("VARCHAR").IsRequired().HasMaxLength(50);**
26. **modelBuilder.Entity<Product>().Property(p=> p.Price).HasColumnType("DECIMAL").IsRequired();**
28. **modelBuilder.Entity<Product>().HasRequired(m=> m.Category).WithMany().HasForeignKey(c => c.CatID);**
30. **}**
31. **public DbSet<Product> Product { get; set; }**
32. **public DbSet<Category> Category { get; set; }**
33. **}**