**انجام مراحل لازم از مرحله طراحی تا پياده‏سازی**

**تهيه و تنظيم: داريوش تصديقی**

نسخه: 1.5 - تاريخ: 04/09/1392

**توصيه‏های کلی**

1. تمام Namespace ها، Class ها، Method ها، Field ها، Property ها و پوشه های پروژه بايد با اسامی بامعنی و کامل و با قانون Pascal نوشته شوند و به هيچ عنوان از (\_) در اين اسامی نبايد استفاده کنيد.
2. تنها کلمه‏ای که به صورت مختصر نوشته می‏شود، Id بوده و بايد دقيقا به همين شکل نوشته شود.
3. در زمان تعريف متغيری از جنس Collection، بايد نام متغير اسم جمع باشد.
4. هر آنچه که اگر ننويسيم، Compiler به طور اتوماتيک می‏نويسد، بايد به صراحت نوشته شود.

**کارهايی که بايد در Class Library، مربوط به Models انجام گيرد:**

1. ايجاد Entity (Model) مورد نظر در Root پروژه
   1. دقت کنيد، در صورتی که پروژه بزرگی در دست داريد و اين پروژه از زيرسيستم‏های مختلف و بعضا مستقلی تشکيل شده‏اند، بهتر است که با توجه به نام آن زيرسيستم‏ها، در داخل Root پروژه، پوشه‏ای به همان نام ايجاد کرده و مدل خود را داخل آن قرار دهيد. (نمونه‏ای از پوشه‏ها در اين خصوص: HumanResources، E-Shop، OfficeAutomation، ...) و صرفا مدل‏های عمومی در Root قرار گيرند، مانند User, Gender, Role و غيره...)
   2. دقت کنيد که اگر به عنوان مثال مدلی به نام Religion (دين/مذهب) در داخل پوشه‌ای به نام HumanResources ايجاد می‌کنيد، بايد Namespace آنرا نيز (که البته به طور اتوماتيک درست می‌شود) به صورت تکميلی بنويسيد: Models.HumanResources
   3. دقت کنيد که اساسا اين کار صرفا به دليل دسته‌بندی نيست! برای همين اگر فقط به همين توصيه ها عمل کنيد، جدول Religions در Schema‌ی پيش فرض يعنی dbo ايجاد می‌گردد. در صورتی که در زيرسيستم‌های مختلف، ممکن است که مدل‌های هم نام داشته باشيم که شايد ارتباطی به زيرسيستم‌های مختلف نداشته باشد، در اين صورت در زمان توليد جداول مربوط به اينگونه مدل‌ها، دچار مشکل خواهيم شد. برای حل اين مشکل، برای کلاس مدل، از Attribute ذيل استفاده می‌کنيم:

[System.ComponentModel.DataAnnotations.Schema.Table("Religions", Schema = "HumanResources")]

در اين صورت، پس از آنکه جدول مربوطه در بانک اطلاعاتی ايجاد گرديد، Schema‌ی آن ديگر مقدار پيش‌فرض dbo نبوده و مقدار آن به HumanResources تغيير می‌کند.

1. Inherit کردن مدل از BaseEntity لازم
2. درج Configuration
3. درج Property های لازم
4. ايجاد پوشه‌ای به نام Resources
5. ايجاد Resource مربوط به مدل، به زبان انگليسی
   1. دقت داشته باشيد که در بالای پنجره Resource، حتما Access Modifier را به گزينه Public، تغيير دهيد.
   2. دقت کنيد که نکاتی که در قسمت 5.a بيان گرديد، در اين قسمت نيز صدق می‌کند! يعنی برای ايجاد Resource مربوط به Religion، در پوشه Resources، ابتدا يک پوشه به نام HumanResources ايجاد کرده و سپس اقدام به ايجاد Religion.resx می‌نماييم.
6. ايجاد Resource به زبان فارسی (Religion.fa-IR.resx)
7. ايجاد فايل tt مربوط به Resource (Religion.tt)
   1. برای اين منظور، از پوشه Documents، (که نکاتی کليدی و آموزشی را در داخل آن قرار می‌دهم) فايل SmartStringResourcesTemplate.tt را به محل مورد نظر کپی کرده و نام آنرا به نام مدل تغيير می‌دهيم. (Religion.tt)
8. درج Attribute های لازم در داخل مد.
9. ايجاد ارتباط بين Entity های ديگر
10. اصلاح Configuration مدل
11. درج يک Collection از Entity، در DatabaseContext
12. درج Configuration Entity در تابع OnModelCreating داخل DatabaseContext
13. در صورت نياز، درج اطلاعات لازم (اوليه) در داخل تابع Seed، DatabaseContextInitializer

**توصيه‏هايی در رابطه با ايجاد Class Library، Models**

1. اصطلاحا به کلاس‏هايی که از آنها Inherit نمی‏شود، Entity می‏گويند.
2. اصطلاحا به کلاس‏هايی که از آنها Inherit می‏شود، BaseEntity می‏گويند.
3. به هيچ عنوان در داخل BaseEntity ها، Property ای از جنس Entity يا BaseEntity های موجود استفاده نمی‏کنيم.
4. به هيچ عنوان در داخل Entity ها، Property ای از جنس BaseEntity های موجود استفاده نمی‏کنيم.
5. به طور کلی در داخل Entity ها، صرفا Property ای از جنس Entity های موجود استفاده می‏کنيم.

**کارهايی که بايد در Class Library مربوط به DAL انجام گيرد:**

1. درج Interface
2. درج توابع لازم در Interface مربوطه
3. درج کلاس
4. درج يک Property در IUnitOfWork
5. درج يک Property در UnitOfWork