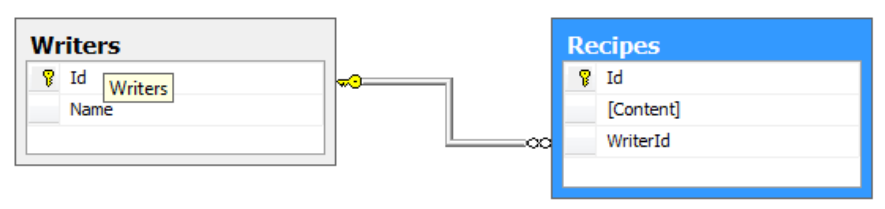
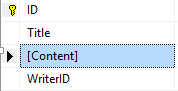
**HƯỚNG DẪN CODE FIRST**

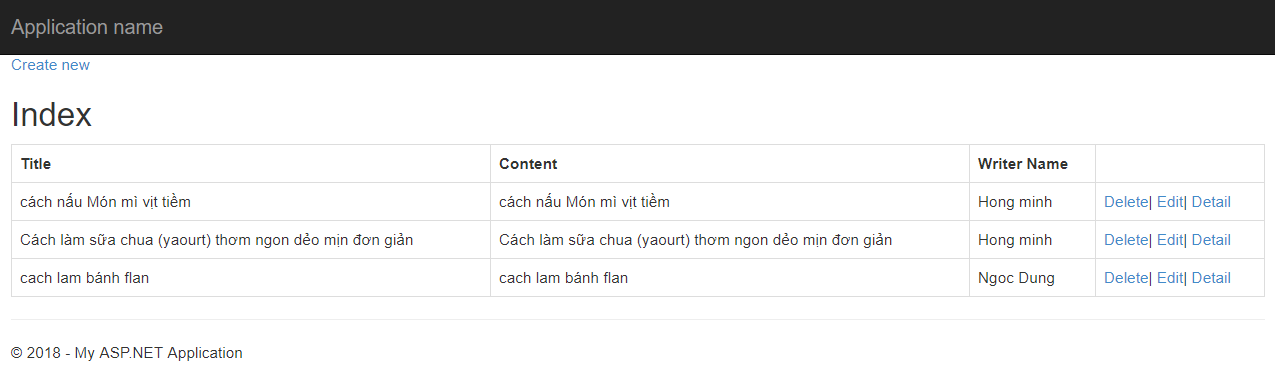
Cho CSDL như sau:



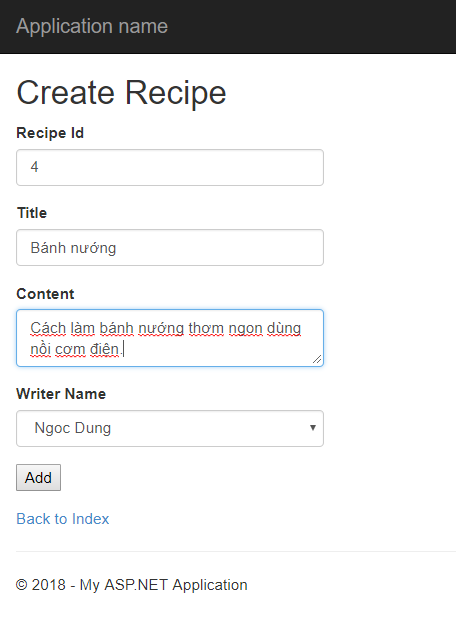
Yêu cầu:

* Tạo database cho ứng dụng dùng code first.
* Kết hợp mô hình 3 lớp và MVC để xây dựng ứng dụng web có các giao diện như sau:

In danh sách các Recipe:

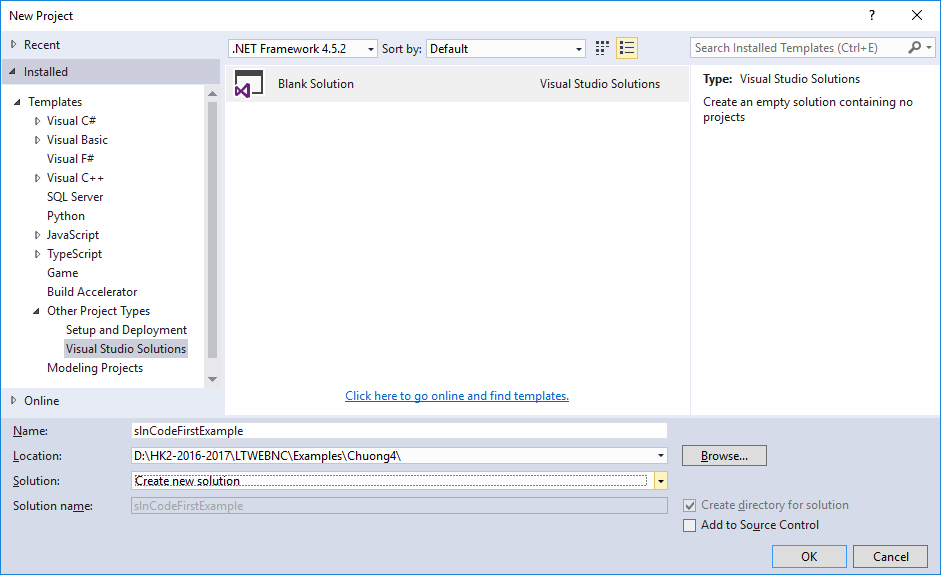


Thêm Recipe mới:



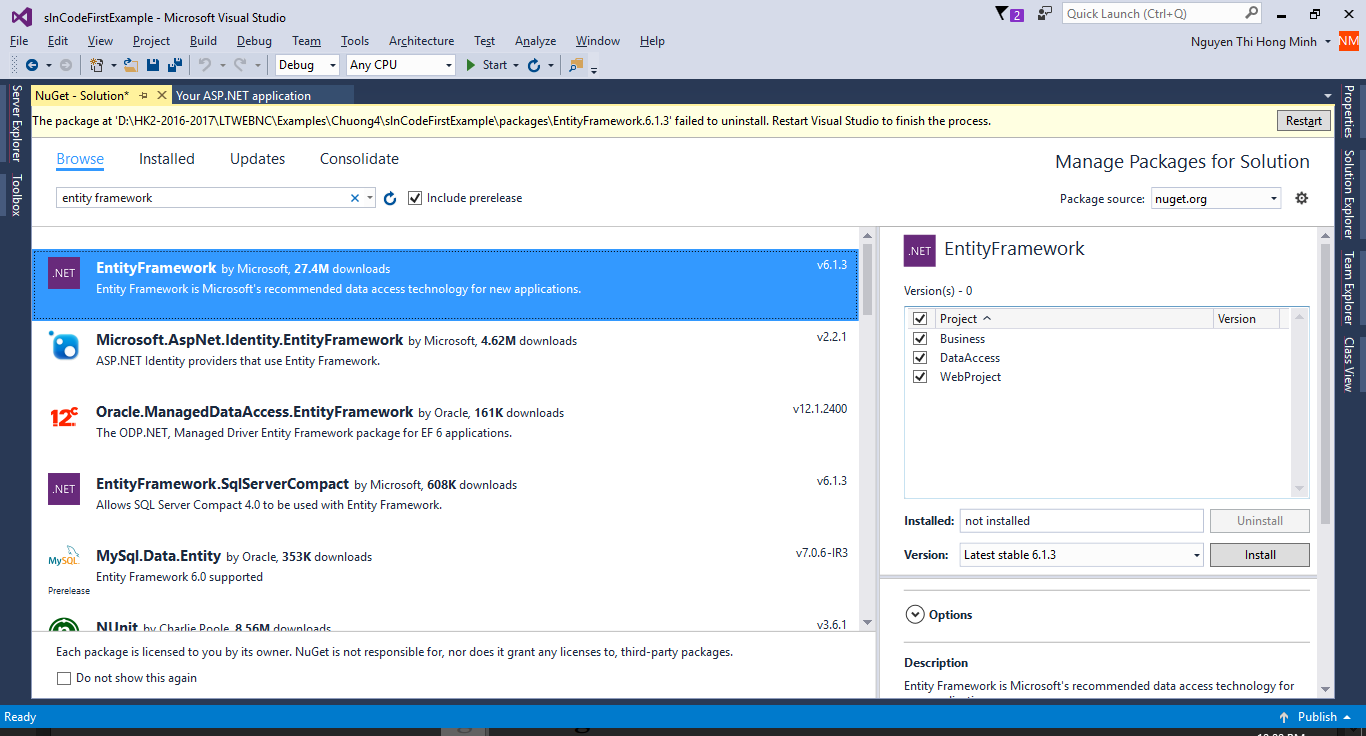
1. Tạo blank solution:

File 🡪 New Project:



1. Trong solution tạo prjModel tier (Class Library):
2. Trong solution tạo DataAccess tier project (Class Library): references prjModel
3. Trong solution tạo Business tier (Class Library): references DataAccess, prjModel
4. Trong solution tạo WebProject tier (MVC Project): references Business, prjModel
5. Install Entity Framework cho solution:

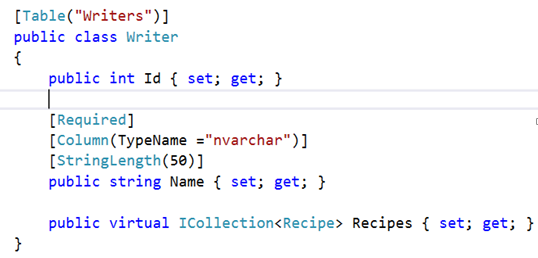
Tools 🡪 Nuget Pakage Manager 🡪Manage Nuget Pakages for Solution

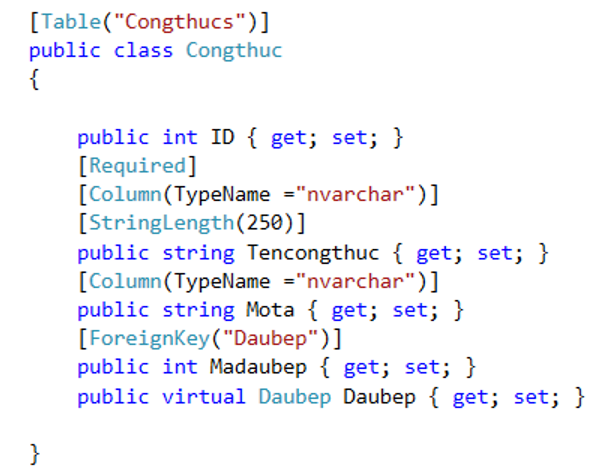


1. Tạo các lớp model trong prjModel:



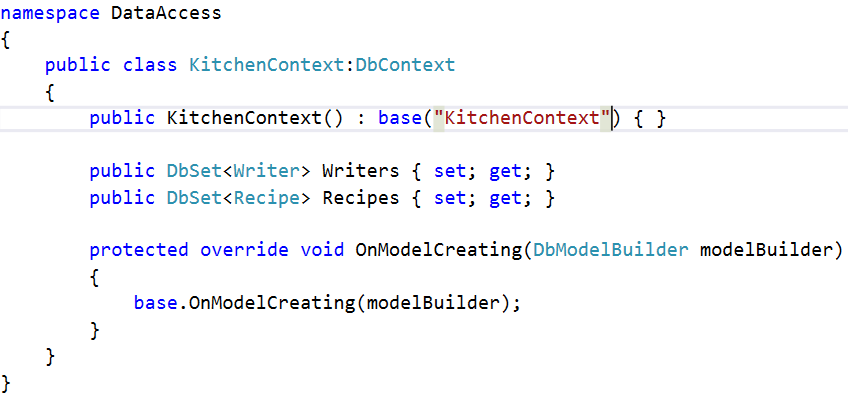
Code:





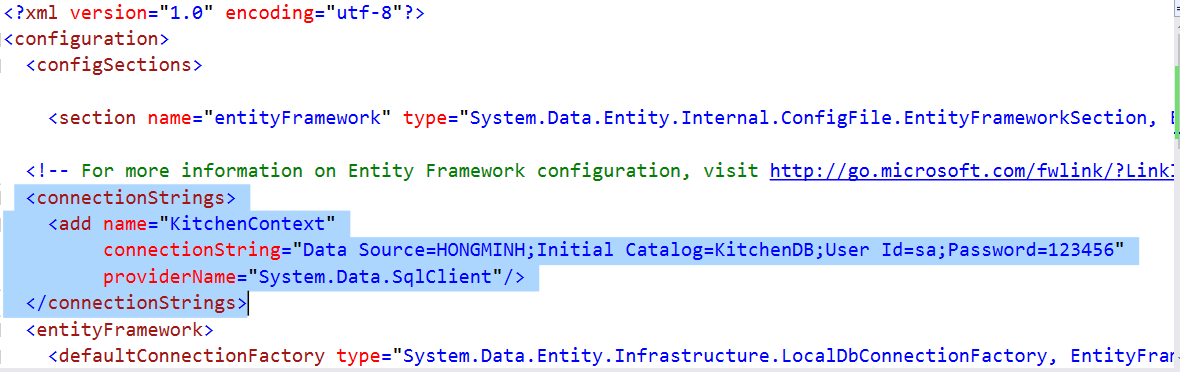
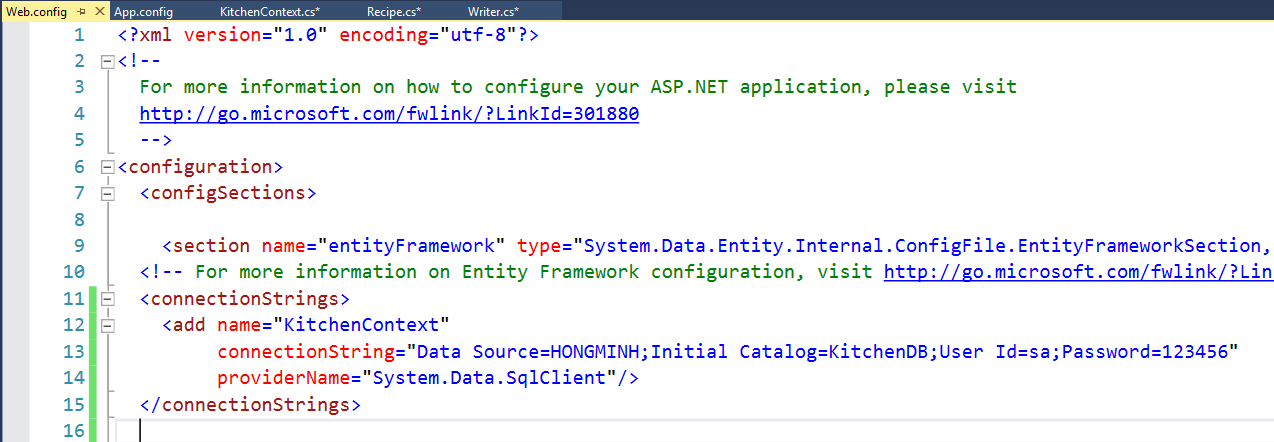
1. Tạo file DataContext trong DataAccess:





1. File Configure:

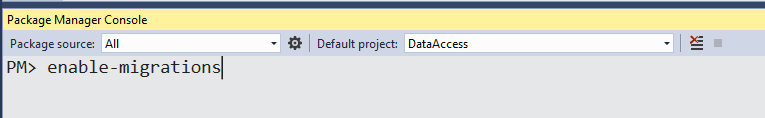
Mở tập tin App.Config trong DataAccess nhập vào thêm <connectionString> như sau:

Mở file Web.Config trong WebProject, copy <connectionString> vào:

1. Tạo database cho Model (migration database):

Tools 🡪 NuGet Pakage Manager 🡪 Pakage Manager Console

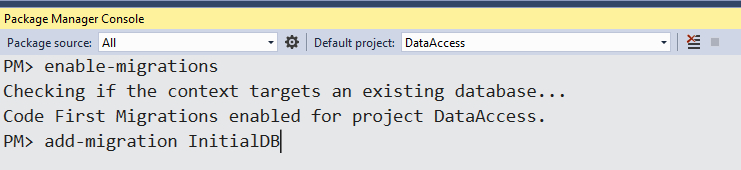
Enable migration=true



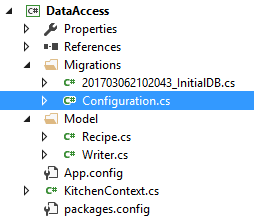
migration

code

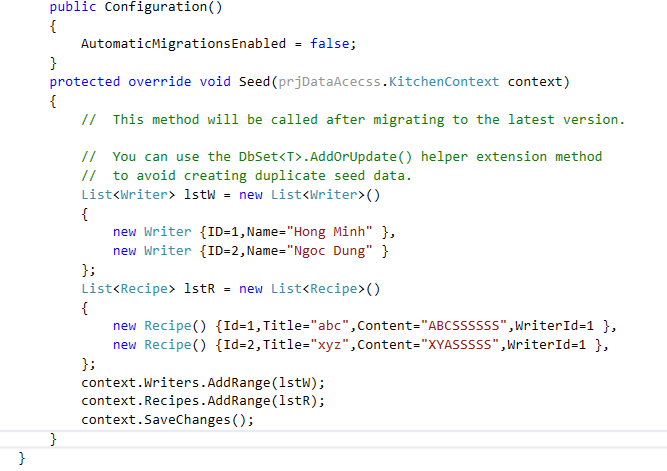
model code



Kiểm tra lại trong DataAcess sẽ thấy có thêm thư mục Migrations chứa 2 tập tin: …\_InitialDB.cs và Configuration.cs:

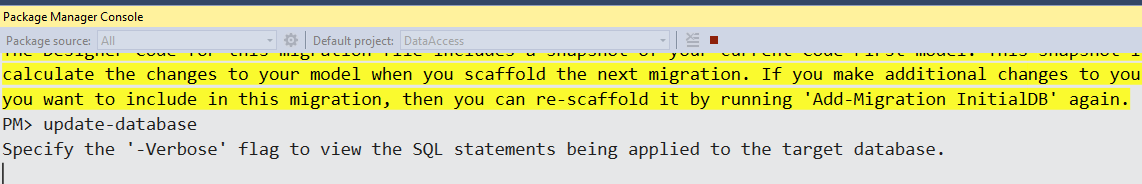


[Sau khi Migration phải thực thi trong Configration.cs giá trị của Database

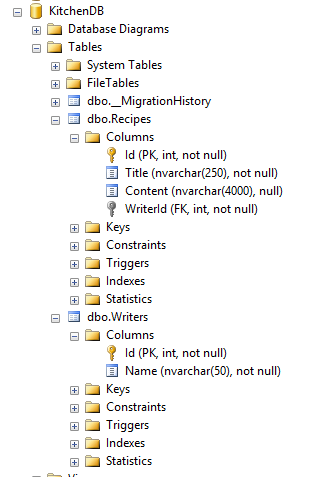


]

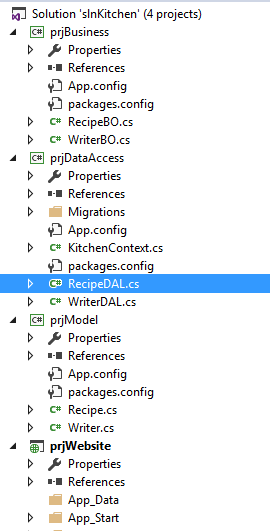
Update-database:



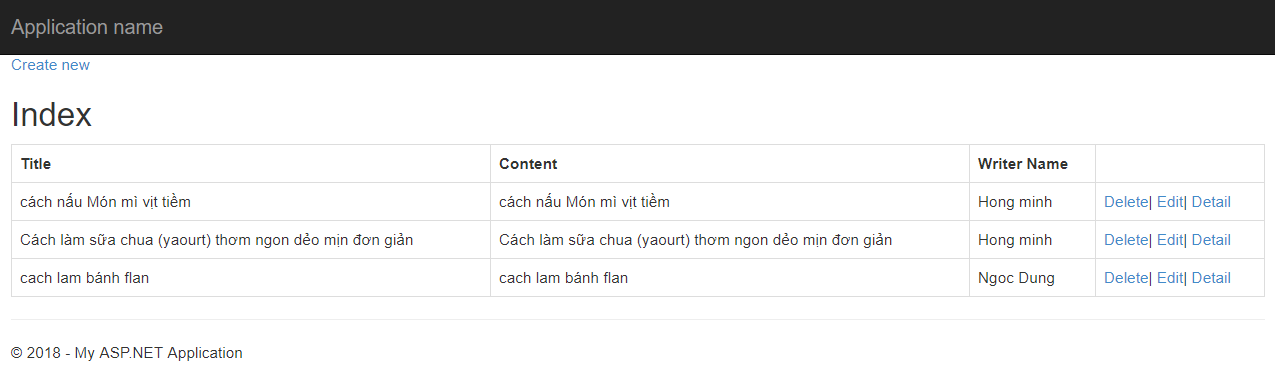
Kiểm tra trong Sql Server database tên KitchenDB:



Cấu trúc trong ứng dụng:



* Tạo trang Xem danh sách Recipe với giao diện như sau:

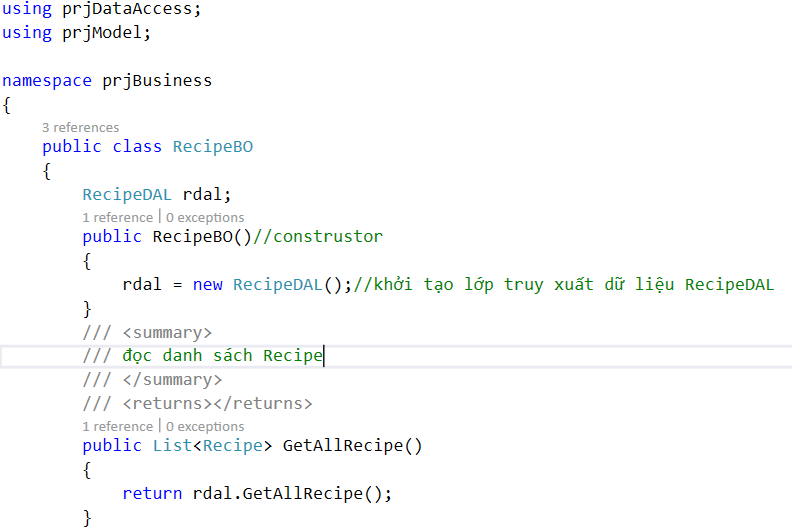


Các bước thực hiện:

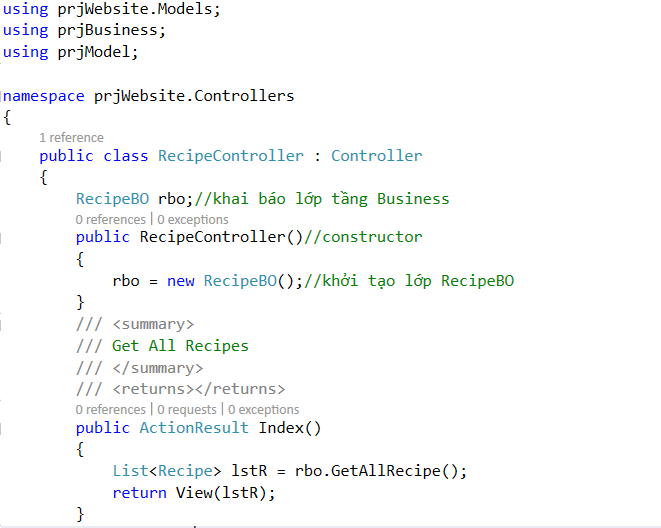
1. Tạo class RecipeDAL trong prjDataAccess dùng để thao tác trên table Recipe với phương thức đọc danh sách tất cả Recipe như sau:



1. Tạo class RecipeBO trong prjBusiness dùng để gọi phương thức đọc danh sách Recipe trong RecipeDAL cung cấp cho prjWebsite như sau:

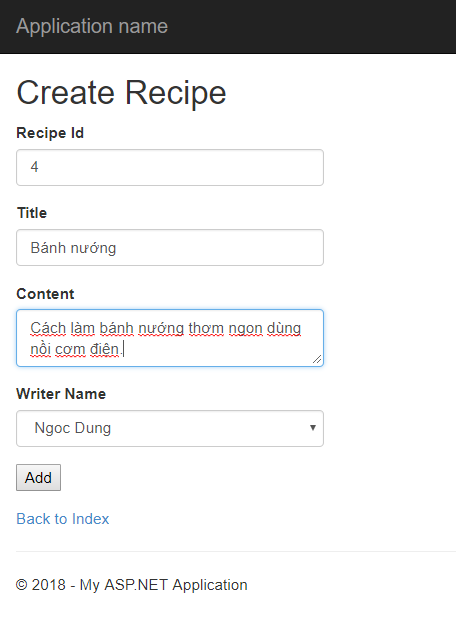


1. Tạo RecipeController với Action Index đọc dữ liệu từ Recipe như sau:



1. Tạo View Index tương ứng với action Index:



* Tạo trang Thêm mới Recipe khi nhấn vào link Create new trên trang Index với giao diện như sau:  
  

Khi thêm xong trả về trang **Index**

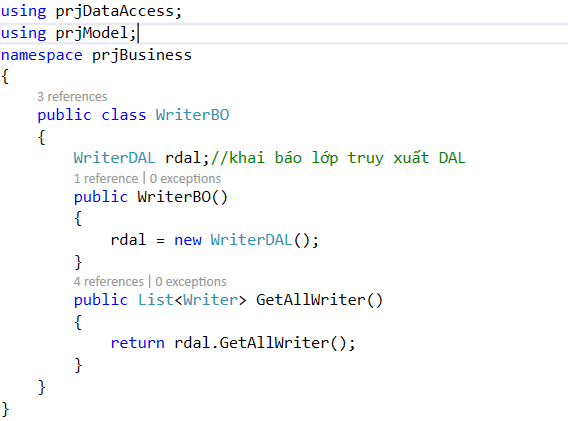
Để thực hiện cho form giao diện thêm, đồng thời trong giao diện có thêm **DropdownList** **Writer**, bạn phải thêm vào class **WriterDAL** (với phương thức **GetAllWriter** trong **prjDataAccess**), **WriterBO** (đọc từ DAL trong **prjBusiness**)

Các bước thực hiện:

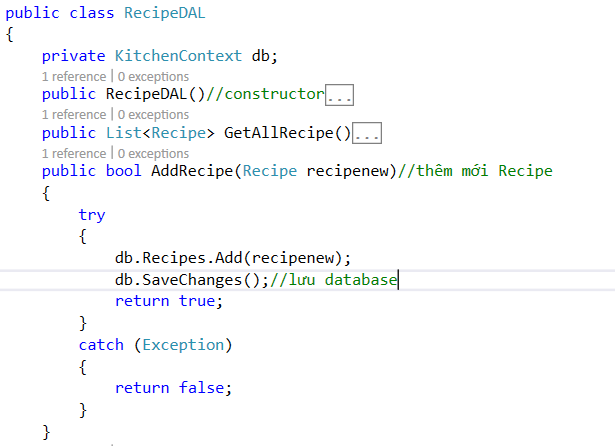
1. Thêm Class WriterDAL trong prjDataAccess:



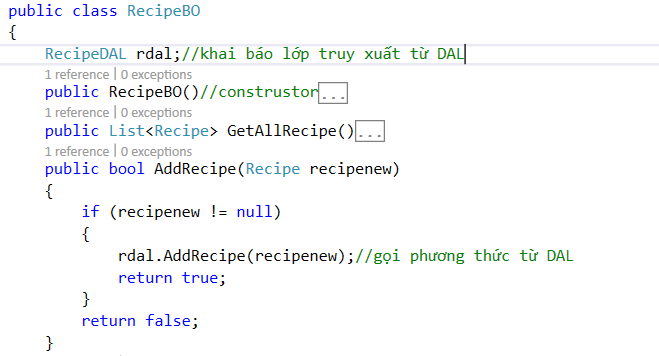
1. Thêm class WriterBO trong prjBusiness như sau:



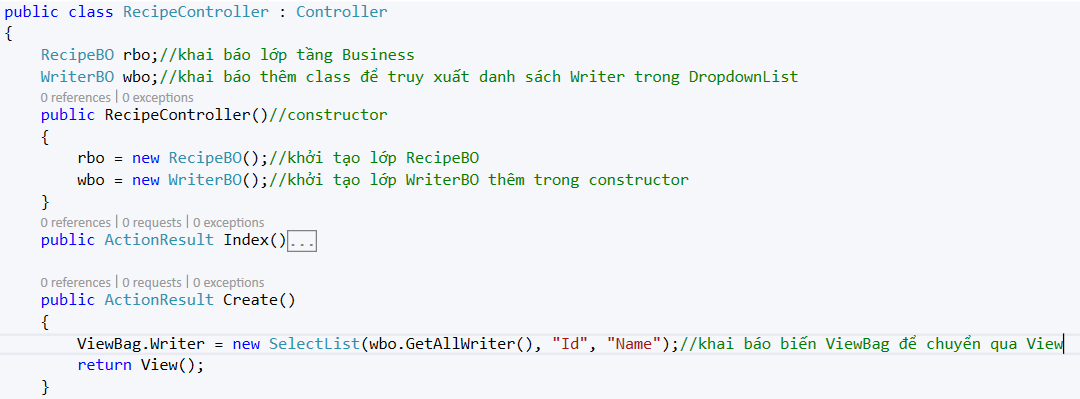
1. Mở class RecipeDAL trong prjDataAccess thêm phương thức AddRecipe như sau:



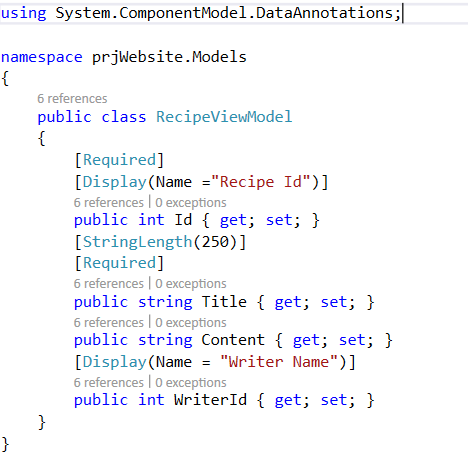
1. Mở class RecipeBO trong prjBusiness thêm phương thức AddRecipe như sau:



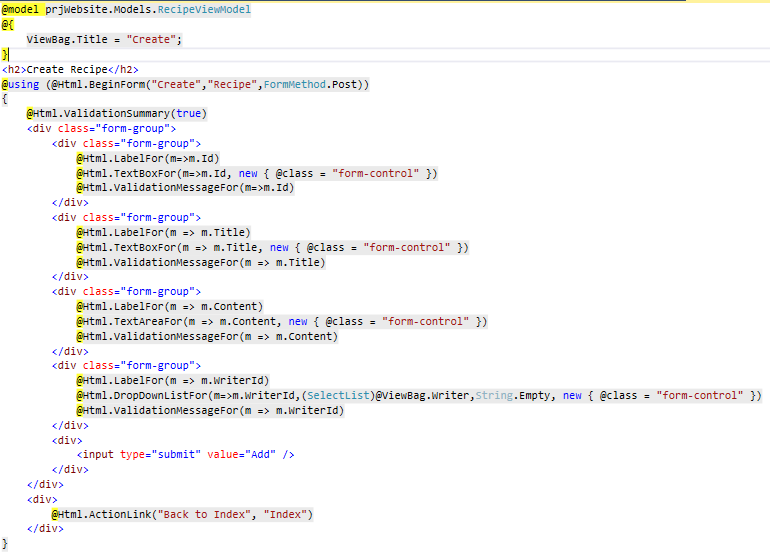
1. Mở RecipeController trong prjWebsite thực hiện thêm Action Create để thực hiện trả về View Create như sau:



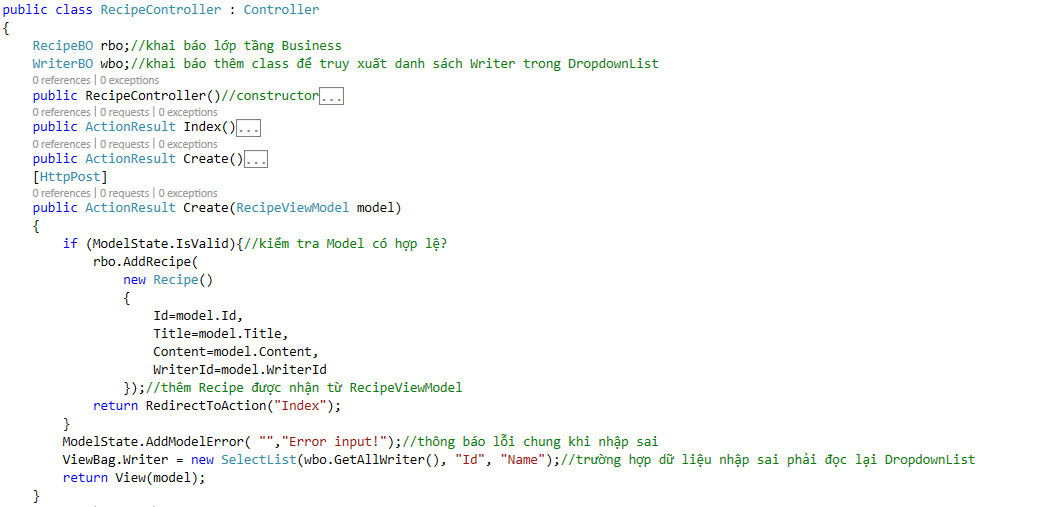
1. Để tạo View **Create** có kiểm tra hợp lệ dữ liệu, bạn thêm trong thư mục **Model** của **prjWebsite** 1 Model Class dùng kiểm tra dữ liệu cho bảng Recipe với tên **RecipeViewModel** để phân biệt với Recipe như sau:



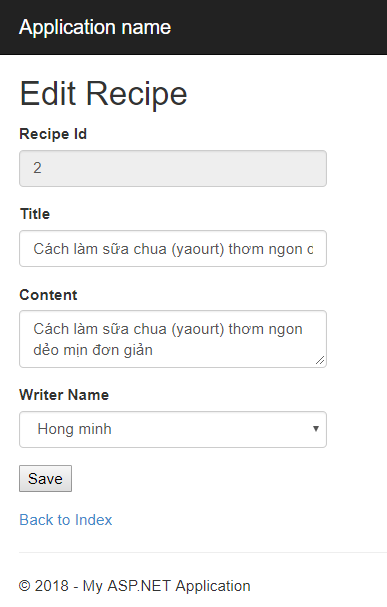
1. Tạo View Create với code như sau (chú ý biến model là RecipeViewModel và DropdownList nhận biến ViewBag.Writer):



1. Khi người dùng nhấn nút Add (submit) trong View Create thì gọi Action Create với method=post, bạn viết thêm Action Create thứ 2 với code như sau:

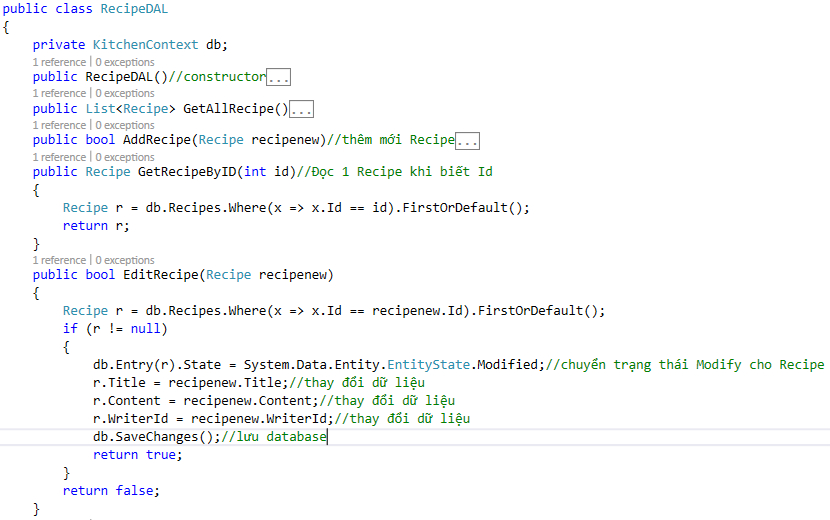


* Tạo trang sửa thông tin Recipe khi nhấn vào link Edit trên trang Index với giao diện như sau:



Các bước thực hiện:

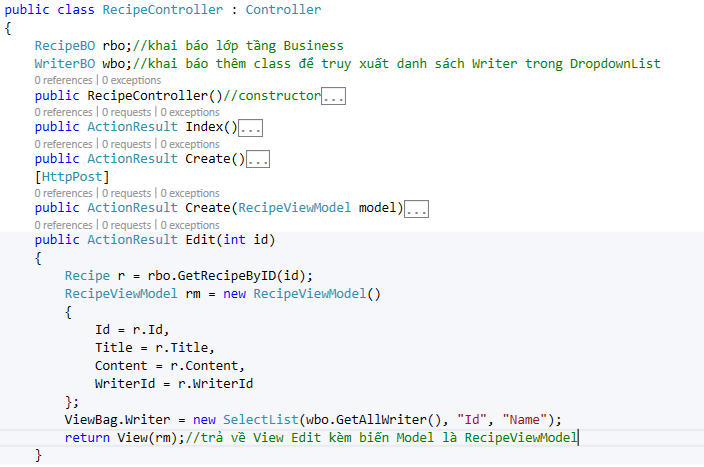
1. Mở class RecipeDAL trong prjDataAccess thêm phương thức EditRecipe và phương thức đọc 1 Recipe khi biết Id: **GetRecipeByID** (vì khi gọi Edit truyền Id từ View về Controller) như sau:



1. Mở class RecipeBO trong prjBusiness thêm phương thức EditRecipe và **GetRecipeByID** như sau:



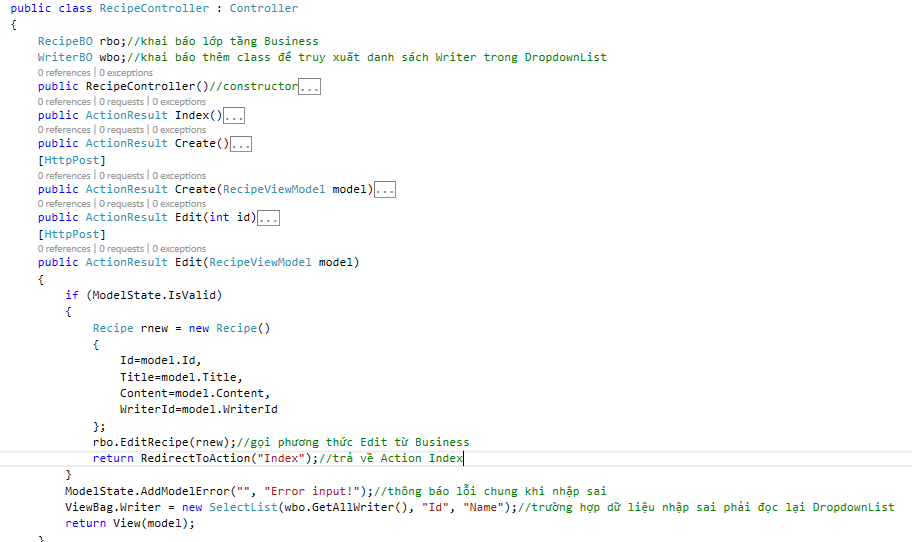
1. Mở RecipeController trong prjWebsite thực hiện thêm Action Edit để thực hiện trả về View Edit như sau:



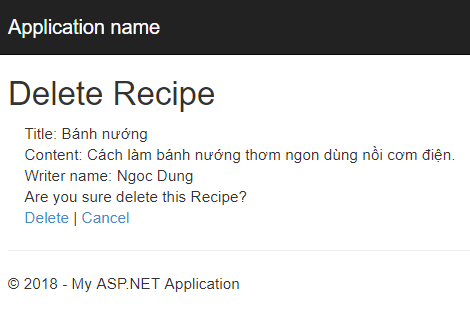
1. Tạo View Edit với code như sau (chú ý biến model là RecipeViewModel và DropdownList nhận biến ViewBag.Writer):



1. Khi người dùng nhấn nút Save (submit) trong View Edit thì gọi Action Edit với method=post, bạn viết thêm Action Edit thứ 2 với code như sau:



* Tạo trang Xóa Recipe khi nhấn vào link Delete trên trang Index với giao diện như sau: (SV tự làm)



* Tạo trang Xem chi tiết Recipe khi nhấn vào link Detail trên trang Index với giao diện như sau: (SV tự làm)

