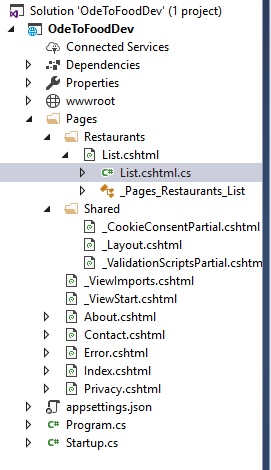
**Tuttorial 1: Đọc dữ liệu từ appsetting.json.**

1. Xây dựng cấu trúc solution như sau:



1. Trong appsetting.json ta có như sau:

{

"Logging": {

"LogLevel": {

"Default": "Warning"

}

},

"AllowedHosts": "\*",

"Message": "Hello from appsetting!"

}

1. Trong List.cshtml.cs ta thêm đoạn code như sau:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using Microsoft.Extensions.Configuration;

namespace OdeToFoodDev.Pages.Restaurants

{

public class ListModel : PageModel

{

private readonly IConfiguration config;

public string Message { get; set; }

public ListModel(IConfiguration config)

{

this.config = config;

}

public void OnGet()

{

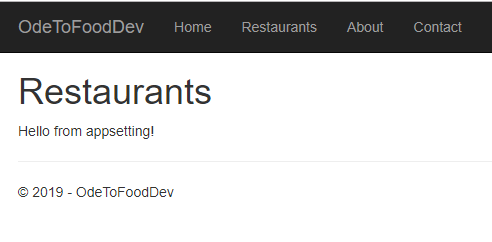
Message = config["Message"];

}

}

}

1. Khi thực thi chương trình ta sẽ được kết quả như sau:



**Tutorial 2: Add new project OdeToFood.Core (class libarary)**

using System;

using System.Collections.Generic;

using System.Text;

namespace OdeToFood.Core

{

public class Restaurant

{

public int Id { get; set; }

public string Name { get; set; }

public string Location { get; set; }

public CuisineType Cuisine { get; set; }

}

}

namespace OdeToFood.Core

{

public enum CuisineType

{

None,

Mexican,

Italian,

Indian

}

}

**Tutorial 3. Add new project OdToFood.Data (class library)**

1. Trong file **OdeToFoodData.csproj** edit lại như sau: (ta cũng có thể sử dụng cách add reference như cách thông thường trong vs).

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<TargetFramework>netcoreapp2.1</TargetFramework>

</PropertyGroup>

<ItemGroup>

<ProjectReference Include="..\OdeToFood.Core\OdeToFood.core.csproj"/>

</ItemGroup>

</Project>

1. Add class và interface.

using System.Collections.Generic;

using OdeToFood.Core;

using System.Linq;

namespace OdeToFood.Data

{

public interface IRestaurantData

{

IEnumerable<Restaurant> GetAll();

}

public class InMemoryRestaurantData : IRestaurantData

{

List<Restaurant> restaurants;

public InMemoryRestaurantData()

{

restaurants = new List<Restaurant>()

{

new Restaurant { Id = 1, Name = "Scott's Pizza", Location="Maryland", Cuisine=CuisineType.Italian},

new Restaurant { Id = 2, Name = "Cinnamon Club", Location="London", Cuisine=CuisineType.Italian},

new Restaurant { Id = 3, Name = "La Costa", Location = "California", Cuisine=CuisineType.Mexican}

};

}

public IEnumerable<Restaurant> GetAll()

{

return from r in restaurants

orderby r.Name

select r;

}

}

}

1. Cấu hình DI trong **Startup.cs.**

public void ConfigureServices(IServiceCollection services)

{

services.AddSingleton<IRestaurantData, InMemoryRestaurantData>();

services.Configure<CookiePolicyOptions>(options =>

{

// This lambda determines whether user consent for non-essential cookies is needed for a given request.

options.CheckConsentNeeded = context => true;

options.MinimumSameSitePolicy = SameSiteMode.None;

});

services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version\_2\_1);

}

**Tutorial 4. Hiển thị danh sách Restaurants.**

1. Trong **List.cshtml.cs** thêm vào các dòng mới như sau:

namespace OdeToFoodDev.Pages.Restaurants

{

public class ListModel : PageModel

{

private readonly IConfiguration config;

private readonly IRestaurantData restaurantData;

public string Message { get; set; }

public IEnumerable<Restaurant> Restaurants { get; set; }

public ListModel(IConfiguration config, IRestaurantData restaurantData)

{

this.config = config;

this.restaurantData = restaurantData;

}

public void OnGet()

{

Message = config["Message"];

Restaurants = restaurantData.GetAll();

}

}

}

1. Trong file List.cshtml ta thêm các dòng code như sau:

@page

@model ListModel

@{

}

<h1>Restaurants</h1>

<table class="table">

@foreach (var restaurant in Model.Restaurants)

{

<tr>

<td>@restaurant.Name</td>

<td>@restaurant.Location</td>

<td>@restaurant.Cuisine</td>

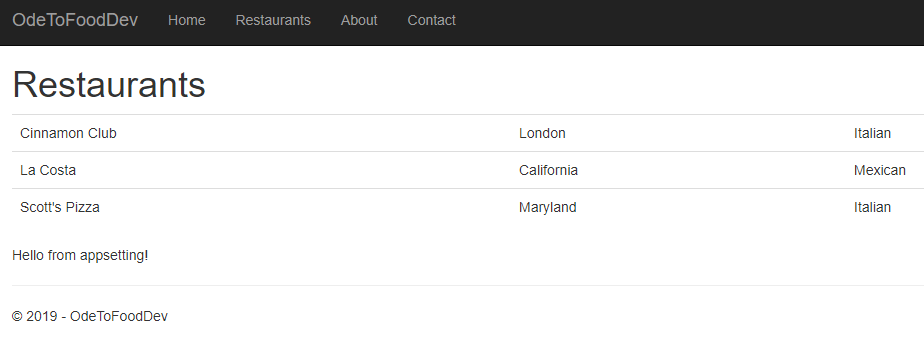
</tr>

}

</table>

<div>@Model.Message</div>

1. Khi ta run sẽ được kết quả như sau:



**WORKING WITH MODEL AND MODEL BINDING.**

**Tutorial 1: Implement search form.**

1. Ta thêm vào List.cshtml các đoạn code sau:

@page

@model ListModel

@{

}

<h1>Restaurants</h1>

<**form** method="get">

<div class="form-group">

<div class="input-group">

<input type="search" class="form-control" value="" />

<span class="input-group-btn">

<button class="btn btn-default">

<i class="glyphicon glyphicon-search"></i>

</button>

</span>

</div>

</div>

</**form**>

<table class="table">

@foreach (var restaurant in Model.Restaurants)

{

<tr>

<td>@restaurant.Name</td>

<td>@restaurant.Location</td>

<td>@restaurant.Cuisine</td>

</tr>

}

</table>

<div>@Model.Message</div>

1. Trong IRestaurantData.cs ta sửa lại method GetAll() như sau

using System.Collections.Generic;

using OdeToFood.Core;

using System.Linq;

namespace OdeToFood.Data

{

public interface IRestaurantData

{

IEnumerable<Restaurant> GetRestaurantsByName(string name);

}

public class InMemoryRestaurantData : IRestaurantData

{

List<Restaurant> restaurants;

public InMemoryRestaurantData()

{

restaurants = new List<Restaurant>()

{

new Restaurant { Id = 1, Name = "Scott's Pizza", Location="Maryland", Cuisine=CuisineType.Italian},

new Restaurant { Id = 2, Name = "Cinnamon Club", Location="London", Cuisine=CuisineType.Italian},

new Restaurant { Id = 3, Name = "La Costa", Location = "California", Cuisine=CuisineType.Mexican}

};

}

public IEnumerable<Restaurant> GetRestaurantsByName(string name = null)

{

return from r in restaurants

where string.IsNullOrEmpty(name)|| r.Name.StartsWith(name)

orderby r.Name

select r;

}

}

}

1. Ta đặt tên cho input ở file List.cshtml

<**form** method="get">

<div class="form-group">

<div class="input-group">

<input type="search" class="form-control" name="searchTerm" value="" />

<span class="input-group-btn">

<button class="btn btn-default">

<i class="glyphicon glyphicon-search"></i>

</button>

</span>

</div>

</div>

</**form**>

1. Trong List.cshtml.cs ta sửa lại method OnGet() như sau:

public void OnGet(string searchTerm)

{

Message = config["Message"];

Restaurants = restaurantData.GetRestaurantsByName(searchTerm);

}

1. Để giữ lại giá trị sau mỗi lần search ta làm như sau:
   1. Ta sửa lại code List.cshtm.cs như sau

namespace OdeToFoodDev.Pages.Restaurants

{

public class ListModel : PageModel

{

private readonly IConfiguration config;

private readonly IRestaurantData restaurantData;

public string Message { get; set; }

public IEnumerable<Restaurant> Restaurants { get; set; }

[BindProperty(SupportsGet =true)]

//Mặc định nó chỉ hỗ trợ POST

public string SearchTerm { get; set; }

public ListModel(IConfiguration config, IRestaurantData restaurantData)

{

this.config = config;

this.restaurantData = restaurantData;

}

public void OnGet()

{

Message = config["Message"];

Restaurants = restaurantData.GetRestaurantsByName(SearchTerm);

}

}

}

* 1. Trong List.cshtml ta modify lại như sau:

<**form** method="get">

<div class="form-group">

<div class="input-group">

<**input** **type**="search" class="form-control" **asp-for**="SearchTerm" />

@\*chỗ này có thể viết tường mình là asp-for=Model.SearchTerm\*@

<span class="input-group-btn">

<button class="btn btn-default">

<i class="glyphicon glyphicon-search"></i>

</button>

</span>

</div>

</div>

</**form**>

**Tutorial 2: Adding Detail page.**

1. Thêm mới trang Detail có nội dung markup như sau:

\

@page "{restaurantId:int}"

@\* Ta có thể khai báo là restaurantId?:int để chỉ ra rằng đó là một optional \*@

@model OdeToFoodDev.Pages.Restaurants.DetailModel

@{

ViewData["Title"] = "Detail";

}

<h2>@Model.Restaurant.Name</h2>

<div>

Id: @Model.Restaurant.Id

</div>

<div>

Location: @Model.Restaurant.Location

</div>

<div>

Cuisine: @Model.Restaurant.Cuisine

</div>

<**a** **asp-page**="./List" class="btn btn-default">All Restaurants</**a**>

1. Trong phần code behind của Detail.cshtml.cs ta thêm code như sau:

namespace OdeToFoodDev.Pages.Restaurants

{

public class DetailModel : PageModel

{

private readonly IRestaurantData restaurantData;

public Restaurant Restaurant { get; set; }

public DetailModel(IRestaurantData restaurantData)

{

this.restaurantData = restaurantData;

}

public IActionResult OnGet(int restaurantId)

{

Restaurant = restaurantData.GetById(restaurantId);

if (Restaurant == null)

return RedirectToPage("./NotFound");

return Page();

}

}

}

1. Trong trang List.cshtml ta sửa lại để có link tới từng sản phẩm.

<table class="table">

@foreach (var restaurant in Model.Restaurants)

{

<tr>

<td>@restaurant.Name</td>

<td>@restaurant.Location</td>

<td>@restaurant.Cuisine</td>

<td>

@\*<a class="btn btn-lg" href="/Restaurant/Detail?restaurantId=@restaurant.Id">\*@

@\* Ta có thể sử dụng cách bên trên để tạo link tới từng sản phẩm \*@

<**a** class="btn btn-lg"

**asp-page**="./Detail"

**asp-route-restaurantId** ="@restaurant.Id">

<i class="glyphicon glyphicon-zoom-in"></i>

</**a**>

</td>

</tr>

}

</table>

1. Trong IRestaurantData.cs

namespace OdeToFood.Data

{

public interface IRestaurantData

{

IEnumerable<Restaurant> GetRestaurantsByName(string name);

Restaurant GetById(int id);

}

public class InMemoryRestaurantData : IRestaurantData

{

List<Restaurant> restaurants;

public InMemoryRestaurantData()

{

restaurants = new List<Restaurant>()

{

new Restaurant { Id = 1, Name = "Scott's Pizza", Location="Maryland", Cuisine=CuisineType.Italian},

new Restaurant { Id = 2, Name = "Cinnamon Club", Location="London", Cuisine=CuisineType.Italian},

new Restaurant { Id = 3, Name = "La Costa", Location = "California", Cuisine=CuisineType.Mexican}

};

}

public Restaurant GetById(int id)

{

return restaurants.SingleOrDefault(r => r.Id == id);

}

public IEnumerable<Restaurant> GetRestaurantsByName(string name = null)

{

return from r in restaurants

where string.IsNullOrEmpty(name)|| r.Name.StartsWith(name)

orderby r.Name

select r;

}

}

}

**EDIT DATA WITH RAZOR PAGE**

**Tutorial 1: Edit dữ liệu.**

1. Thêm mới trang Edit.cshtml

@page "{restaurantId:int}"

@model OdeToFoodDev.Pages.Restaurants.EditModel

@{

ViewData["Title"] = "Edit";

}

<h2>Editing @Model.Restaurant.Name</h2>

<**form** method="post">

<**input** **type**="hidden" **asp-for**="Restaurant.Id" />

<div class="form-group">

<**label** **asp-for**="Restaurant.Name"></**label**>

<**input** **asp-for**="Restaurant.Name" class="form-control" />

<**span** class="text-danger" **asp-validation-for**="Restaurant.Name"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Restaurant.Location"></**label**>

<**input** **asp-for**="Restaurant.Location" class="form-control" />

<**span** class="text-danger" **asp-validation-for**="Restaurant.Location"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Restaurant.Cuisine"></**label**>

<**select** class="form-control" **asp-for**="Restaurant.Cuisine" **asp-items**="Model.Cuisines"></**select**>

<**span** class="text-danger" **asp-validation-for**="Restaurant.Cuisine"></**span**>

@\* vì thằng asp-items có thể nhận dữ liệu từ bất cứ từ đâu nên phải chỉ rõ ra là Cuisines này là của thằng nào. \*@

</div>

@\* Nếu sử dụng asp-for nó sẽ tự tạo tên và value \*@

<button type="submit" class="btn btn-primary">Save</button>

</**form**>

1. Trong phần code behind để xử lý trang này.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using Microsoft.AspNetCore.Mvc.Rendering;

using OdeToFood.Core;

using OdeToFood.Data;

namespace OdeToFoodDev.Pages.Restaurants

{

public class EditModel : PageModel

{

private readonly IRestaurantData restaurantData;

private readonly IHtmlHelper htmlHelper;

[BindProperty]

public Restaurant Restaurant { get; set; }

public IEnumerable<SelectListItem> Cuisines { get; set; }

public EditModel(IRestaurantData restaurantData, IHtmlHelper htmlHelper)

{

this.restaurantData = restaurantData;

this.htmlHelper = htmlHelper;

}

public IActionResult OnGet(int restaurantId)

{

Cuisines = htmlHelper.GetEnumSelectList<CuisineType>();

Restaurant = restaurantData.GetById(restaurantId);

if (Restaurant == null)

return RedirectToPage("./NotFound");

return Page();

}

public IActionResult OnPost()

{

if (ModelState.IsValid)

{

Restaurant = restaurantData.Update(Restaurant);

// Ta có thể viết như sau

// restaurantData.Update(Restaurant);

restaurantData.Commit();

return RedirectToPage("./Detail", new { restaurantId = Restaurant.Id });

}

Cuisines = htmlHelper.GetEnumSelectList<CuisineType>();

return Page();

}

}

}

1. Trong IRestaurantData.cs ta thêm vào 2 methods.

using OdeToFood.Core;

using System.Collections.Generic;

using System.Linq;

namespace OdeToFood.Data

{

public interface IRestaurantData

{

IEnumerable<Restaurant> GetRestaurantsByName(string name);

Restaurant GetById(int id);

Restaurant Update(Restaurant updateRestaurant);

int Commit();

}

public class InMemoryRestaurantData : IRestaurantData

{

private List<Restaurant> restaurants;

public InMemoryRestaurantData()

{

restaurants = new List<Restaurant>()

{

new Restaurant { Id = 1, Name = "Scott's Pizza", Location="Maryland", Cuisine=CuisineType.Italian},

new Restaurant { Id = 2, Name = "Cinnamon Club", Location="London", Cuisine=CuisineType.Italian},

new Restaurant { Id = 3, Name = "La Costa", Location = "California", Cuisine=CuisineType.Mexican}

};

}

public int Commit()

{

return 0;

}

public Restaurant GetById(int id)

{

return restaurants.SingleOrDefault(r => r.Id == id);

}

public IEnumerable<Restaurant> GetRestaurantsByName(string name = null)

{

return from r in restaurants

where string.IsNullOrEmpty(name) || r.Name.StartsWith(name)

orderby r.Name

select r;

}

public Restaurant Update(Restaurant updateRestaurant)

{

var restaurant = restaurants.SingleOrDefault(r => r.Id == updateRestaurant.Id);

if (restaurant != null)

{

restaurant.Name = updateRestaurant.Name;

restaurant.Location = updateRestaurant.Location;

restaurant.Cuisine = updateRestaurant.Cuisine;

}

return restaurant;

}

}

}

1. Trong List.cshtml.

@page

@model ListModel

@{

}

<h1>Restaurants</h1>

<**form** method="get">

<div class="form-group">

<div class="input-group">

<**input** **type**="search" class="form-control" **asp-for**="SearchTerm" />

@\*chỗ này có thể viết tường mình là asp-for=Model.SearchTerm\*@

<span class="input-group-btn">

<button class="btn btn-default">

<i class="glyphicon glyphicon-search"></i>

</button>

</span>

</div>

</div>

</**form**>

<table class="table">

@foreach (var restaurant in Model.Restaurants)

{

<tr>

<td>@restaurant.Name</td>

<td>@restaurant.Location</td>

<td>@restaurant.Cuisine</td>

<td>

@\*<a class="btn btn-lg" href="/Restaurant/Detail?restaurantId=@restaurant.Id">\*@

@\* Ta có thể sử dụng cách bên trên để tạo link tới từng sản phẩm \*@

<**a** class="btn btn-lg"

**asp-page**="./Detail"

**asp-route-restaurantId**="@restaurant.Id">

<i class="glyphicon glyphicon-zoom-in"></i>

</**a**>

<**a** class="btn btn-lg"

**asp-page**="./Edit"

**asp-route-restaurantId**="@restaurant.Id">

<i class="glyphicon glyphicon-edit" ></i>

</**a**>

</td>

</tr>

}

</table>

<div>@Model.Message</div>

1. Để có được validate trong Restaurant.cs ở project core ta thêm vào như sau:

namespace OdeToFood.Core

{

public class Restaurant

{

// Nếu muốn viết custome validate thì ta cho model kế thừa

// từ IValidatableObject.

public int Id { get; set; }

[Required]

[StringLength(80)]

public string Name { get; set; }

[Required, StringLength(255)]

public string Location { get; set; }

public CuisineType Cuisine { get; set; }

}

}

**Tutorial 2: Thêm mới dữ liệu.**

1. Trong IRestaurantData.cs.

public Restaurant Add(Restaurant newRestaurant)

{

restaurants.Add(newRestaurant);

newRestaurant.Id = restaurants.Max(r=>r.Id)+1;

return newRestaurant;

}

1. Trong Edit.cshtml.

@page "{restaurantId:int?}"

@model OdeToFoodDev.Pages.Restaurants.EditModel

@{

ViewData["Title"] = "Edit";

}

<h2>Editing @Model.Restaurant.Name</h2>

<**form** method="post">

<**input** **type**="hidden" **asp-for**="Restaurant.Id" />

<div class="form-group">

<**label** **asp-for**="Restaurant.Name"></**label**>

<**input** **asp-for**="Restaurant.Name" class="form-control" />

<**span** class="text-danger" **asp-validation-for**="Restaurant.Name"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Restaurant.Location"></**label**>

<**input** **asp-for**="Restaurant.Location" class="form-control" />

<**span** class="text-danger" **asp-validation-for**="Restaurant.Location"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Restaurant.Cuisine"></**label**>

<**select** class="form-control" **asp-for**="Restaurant.Cuisine" **asp-items**="Model.Cuisines"></**select**>

<**span** class="text-danger" **asp-validation-for**="Restaurant.Cuisine"></**span**>

@\* vì thằng asp-items có thể nhận dữ liệu từ bất cứ từ đâu nên phải chỉ rõ ra là Cuisines này là của thằng nào. \*@

</div>

@\* Nếu sử dụng asp-for nó sẽ tự tạo tên và value \*@

<button type="submit" class="btn btn-primary">Save</button>

</**form**>

1. Trong Edit.cshtml.cs

namespace OdeToFoodDev.Pages.Restaurants

{

public class EditModel : PageModel

{

private readonly IRestaurantData restaurantData;

private readonly IHtmlHelper htmlHelper;

[BindProperty]

public Restaurant Restaurant { get; set; }

public IEnumerable<SelectListItem> Cuisines { get; set; }

public EditModel(IRestaurantData restaurantData, IHtmlHelper htmlHelper)

{

this.restaurantData = restaurantData;

this.htmlHelper = htmlHelper;

}

public IActionResult OnGet(int? restaurantId)

{

Cuisines = htmlHelper.GetEnumSelectList<CuisineType>();

if (restaurantId.HasValue)

{

Restaurant = restaurantData.GetById(restaurantId.Value);

}

else

{

Restaurant = new Restaurant();

}

if (Restaurant == null)

{

return RedirectToPage("./NotFound");

}

return Page();

}

public IActionResult OnPost()

{

if (!ModelState.IsValid)

{

Cuisines = htmlHelper.GetEnumSelectList<CuisineType>();

return Page();

}

if (Restaurant.Id > 0)

{

Restaurant = restaurantData.Update(Restaurant);

}

else

{

restaurantData.Add(Restaurant);

}

// Ta có thể viết như sau

// restaurantData.Update(Restaurant);

TempData["Message"] = "Restaurant Saved!";

restaurantData.Commit();

return RedirectToPage("./Detail", new { restaurantId = Restaurant.Id });

}

}

}

1. Trong detail.cshtml.cs

namespace OdeToFoodDev.Pages.Restaurants

{

public class DetailModel : PageModel

{

private readonly IRestaurantData restaurantData;

[TempData]

public string Message { get; set; }

public Restaurant Restaurant { get; set; }

public DetailModel(IRestaurantData restaurantData)

{

this.restaurantData = restaurantData;

}

public IActionResult OnGet(int restaurantId)

{

Restaurant = restaurantData.GetById(restaurantId);

if (Restaurant == null)

return RedirectToPage("./NotFound");

return Page();

}

}

}

1. Trong Detail.cshtml.

@page "{restaurantId:int}"

@\* Ta có thể khai báo là restaurantId?:int để chỉ ra rằng đó là một optional \*@

@model OdeToFoodDev.Pages.Restaurants.DetailModel

@{

ViewData["Title"] = "Detail";

}

<h2>@Model.Restaurant.Name</h2>

<div>

Id: @Model.Restaurant.Id

</div>

<div>

Location: @Model.Restaurant.Location

</div>

<div>

Cuisine: @Model.Restaurant.Cuisine

</div>

@if (Model.Message !=null )

{

<div class="alert alert-info">@Model.Message</div>

}

<**a** **asp-page**="./List" class="btn btn-default">All Restaurants</**a**>

WORKING WITH SQL SERVER AND THE EF CORE.

Nếu muốn sử dụng EF core cần cài các thứ sau:

* **Microsoft.EntityFrameworkCore**
* **Microsoft.EntityFrameworkCore.Design**
* **Microsoft.EntityFrameworkCore.SqlServer**

1. ***dotnet ef dbcontext info*** // kiểm tra information của dbcontext.
2. ***dotnet ef migrations add initial -s ..\OdeToFoodDev\OdeToFoodDev.csproj***