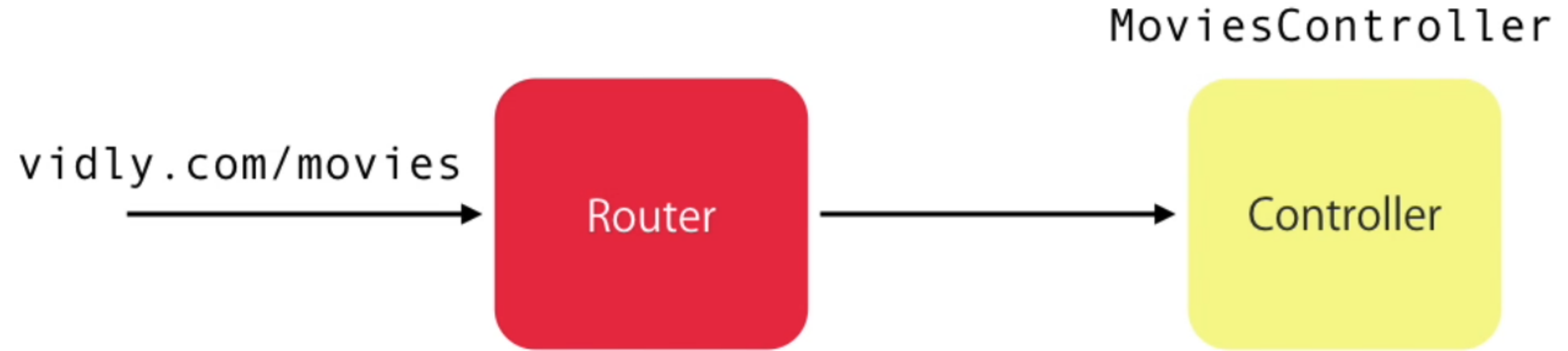


ASP.NET MVC Fundamentals

Tosaporn Saiyim



Router

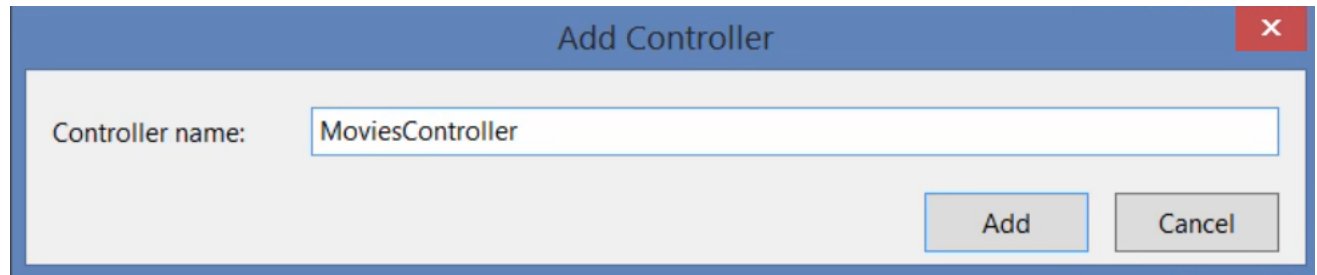
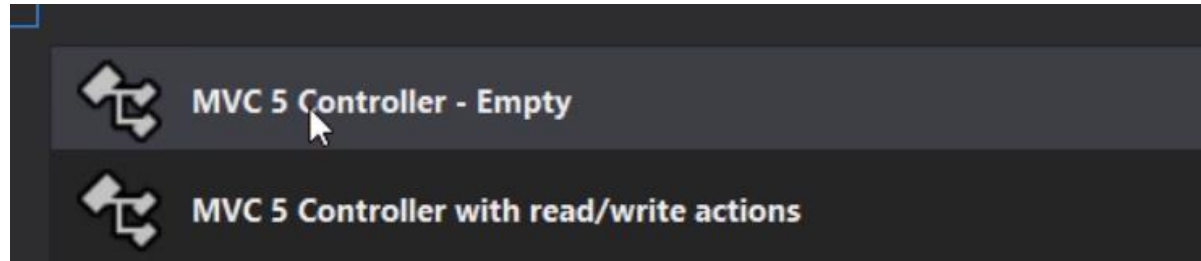


Class ไฟล์ชื่อ Movie

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Vidly.Models
{
    public class Movie
    {
        public int Id { get; set; }
        public string Name { get; set; }
    }
}
```

คลิกขวาที่ controller > Add > Controller



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Vidly.Controllers
{
    public class MoviesController : Controller
    {
        // GET: Movies
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

ปรับ Movie controller

สร้าง Object movie และกำหนดค่า Name = Shrek

```
namespace Vidly.Controllers
{
    public class MoviesController : Controller
    {
        // GET: Movies/Random
        public ActionResult Random()
        {
            var movie = new Movie() { Name = "Shrek!" };

            return View();
        }
    }
}
```

คลิกขวาที่ Random > Add View

Add View

View name: Random

Template: Empty (without model)

Model class:

Data context class:

Options:

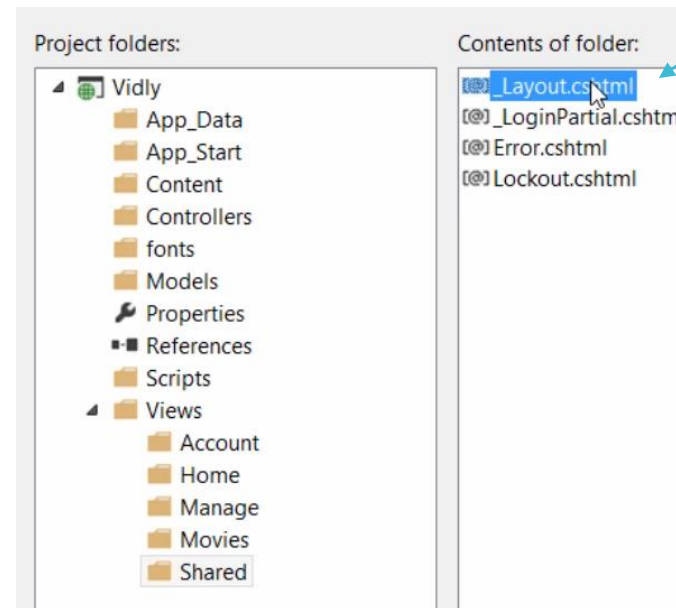
☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)


Add Cancel



เลือกใช้ Layout

```
Movie.cs | MoviesController.cs | Random.cshtml [X]
@{
    ViewBag.Title = "Random";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Random</h2>
```

ส่ง Object “movie” ไปที่หน้า View ด้วย Method Random



```
public class MoviesController : Controller
{
    // GET: Movies/Random
    public ActionResult Random()
    {
        var movie = new Movie() { Name = "Shrek!" };

        return View(movie);
    }
}
```

เรียกใช้โมเดล Movie

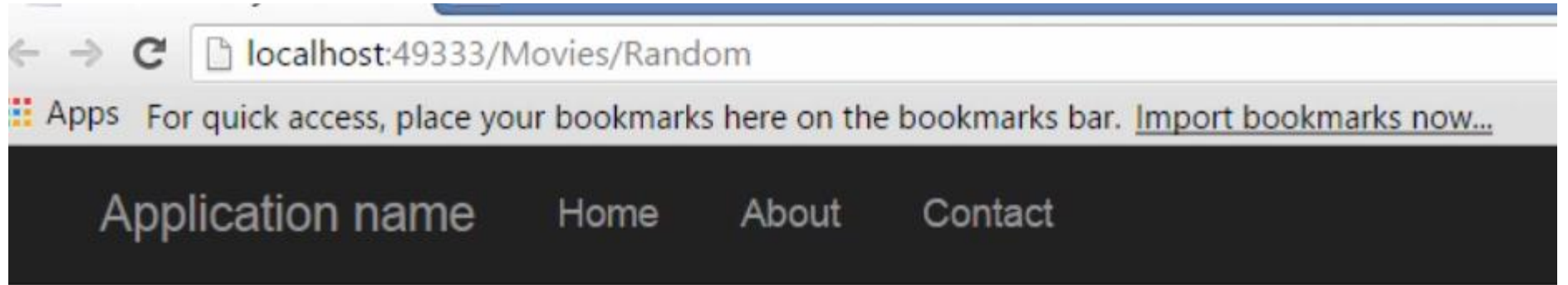


```
Movie.cs | MoviesController.cs • | Random.cshtml • X
@model Vidly.Models.Movie
@{
    ViewBag.Title = "Random";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>@Model.Name</h2>
```

นำค่าที่ส่งมาจาก movie ออกมา ลอง รัน

Controller

Method "Random"



ค่าที่ส่งมาคือ
Name Shrek

→ Shrek!

© 2016 - My ASP.NET Application

Action Result

```
namespace Vidly.Controllers
{
    public class MoviesController : Controller
    {
        // GET: Movies/Random
        public ActionResult Random()
        {
            var movie = new Movie() { Name = "Shrek!" };

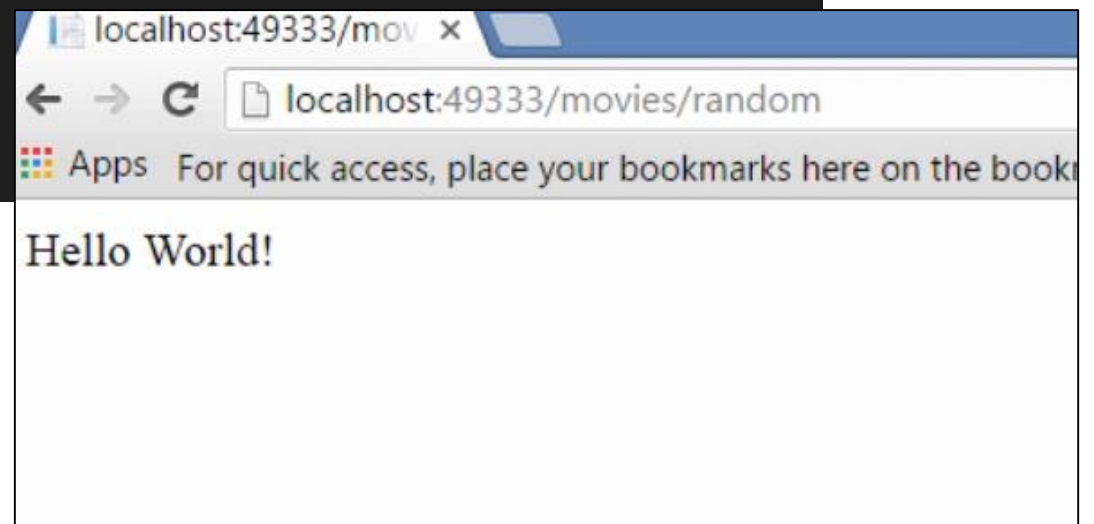
            return View(movie);
        }
    }
}
```


Action Results

Type	Helper Method
ViewResult	View()
PartialViewResult	PartialView()
ContentResult	Content()
RedirectResult	Redirect()
RedirectToRouteResult	RedirectToAction()
JsonResult	Json()
FileResult	File()
HttpNotFoundResult	HttpNotFound()
EmptyResult	

```
namespace Vidly.Controllers
{
    public class MoviesController : Controller
    {
        // GET: Movies/Random
        public ActionResult Random()
        {
            var movie = new Movie() { Name = "Shrek!" };

            // return View(movie);
            return Content("Hello World!");
        }
    }
}
```



```
namespace Vidly.Controllers
{
    public class MoviesController : Controller
    {
        // GET: Movies/Random
        public ActionResult Random()
        {
            var movie = new Movie() { Name = "Shrek!" };

            // return View(movie);
            // return Content("Hello World!");
            return HttpNotFound();
        }
    }
}
```

IIS 8.0 Detailed Error x

localhost:49333/movies/random

Apps For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks](#)

HTTP Error 404.0 - Not Found

The resource you are looking for has been removed, had its name changed, or is temporarily unavailable.

Most likely causes:

- The directory or file specified does not exist on the Web server.
- The URL contains a typographical error.
- A custom filter or module, such as URLScan, restricts access to the file.

```

namespace Vidly.Controllers
{
    public class MoviesController : Controller
    {
        // GET: Movies/Random
        public ActionResult Random()
        {
            var movie = new Movie() { Name = "Shrek!" };

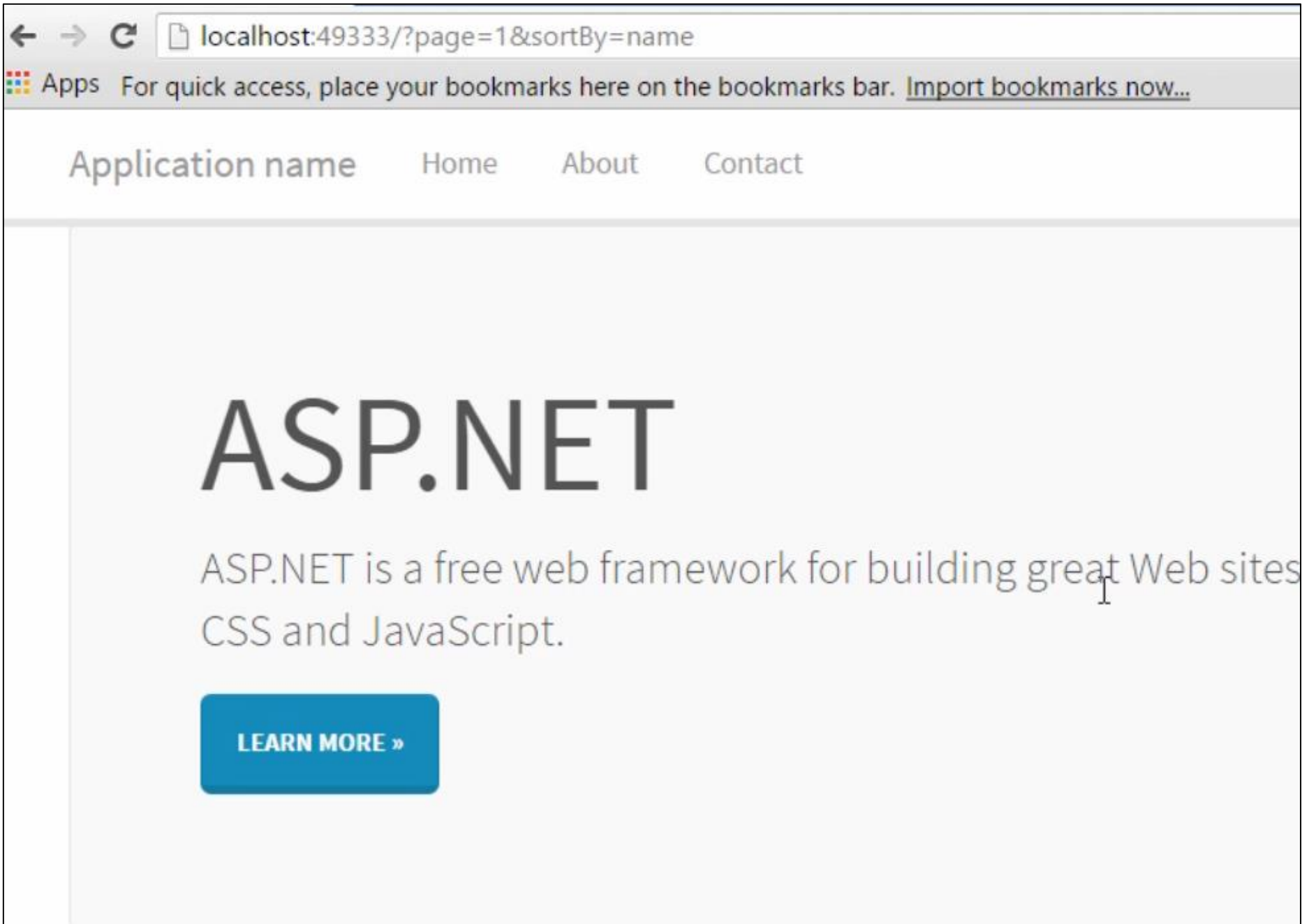
            // return View(movie);
            // return Content("Hello World!");
            // return HttpNotFound();
            // return new EmptyResult();
            return RedirectToAction("Index", "Home");
        }
    }
}

```

```

return new EmptyResult();
RedirectToAction("Index", "Home", new { page = 1, sortBy = "name" });

```



Parameter Binding



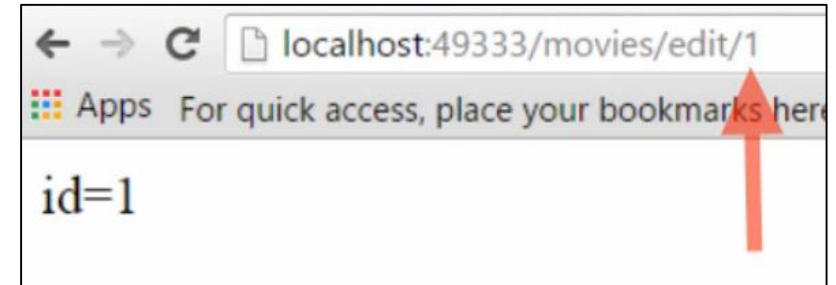
Parameter Sources

- In the URL: /movies/edit/1
- In the query string: /movies/edit?id=1
- In the form data: id=1

```
public class MoviesController : Controller
{
    // GET: Movies/Random
    public ActionResult Random()
    {
        var movie = new Movie() { Name = "Shrek!" };

        return View(movie);
    }

    public ActionResult Edit(int id)
    {
        return Content("id=" + id);
    }
}
```



Custom Route

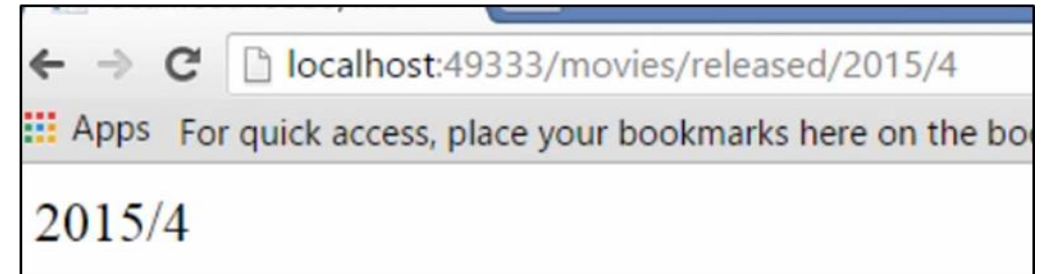
/movies/released/2015/04

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

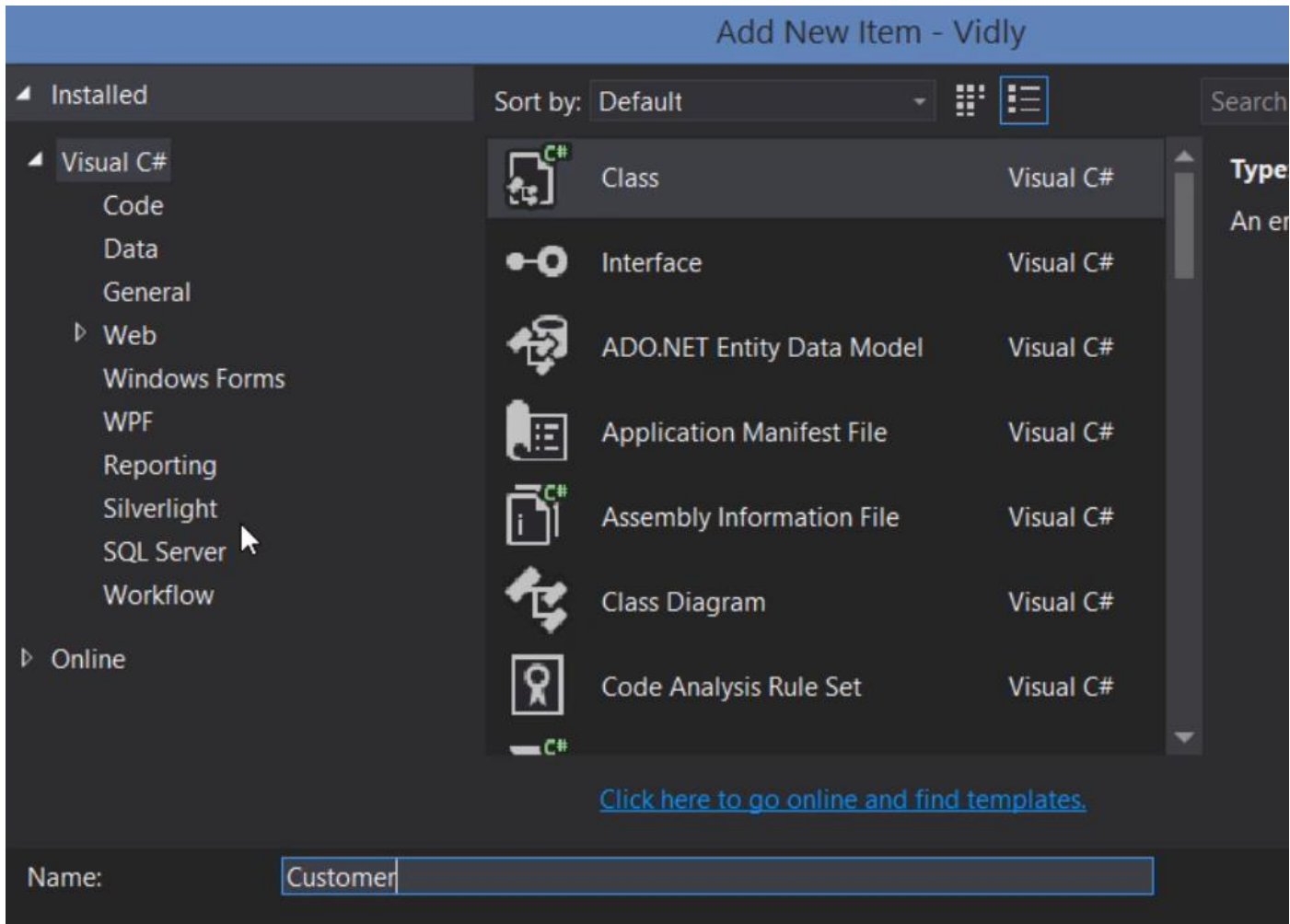
    routes.MapRoute(
        "MoviesByReleaseDate",
        "movies/released/{year}/{month}",
        new { controller = "Movies", action = "ByReleaseDate" });
}
```

```
public class MoviesController : Controller
{
    // GET: Movies/Random
    public ActionResult Random()...

    public ActionResult ByReleaseDate(int year, int month)
    {
        return Content(year + "/" + month);
    }
}
```

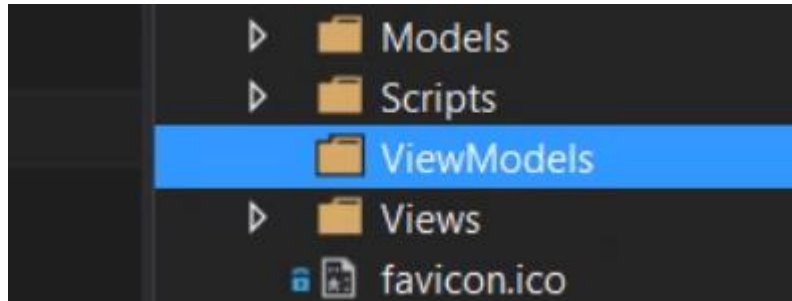


New Class -> Customer

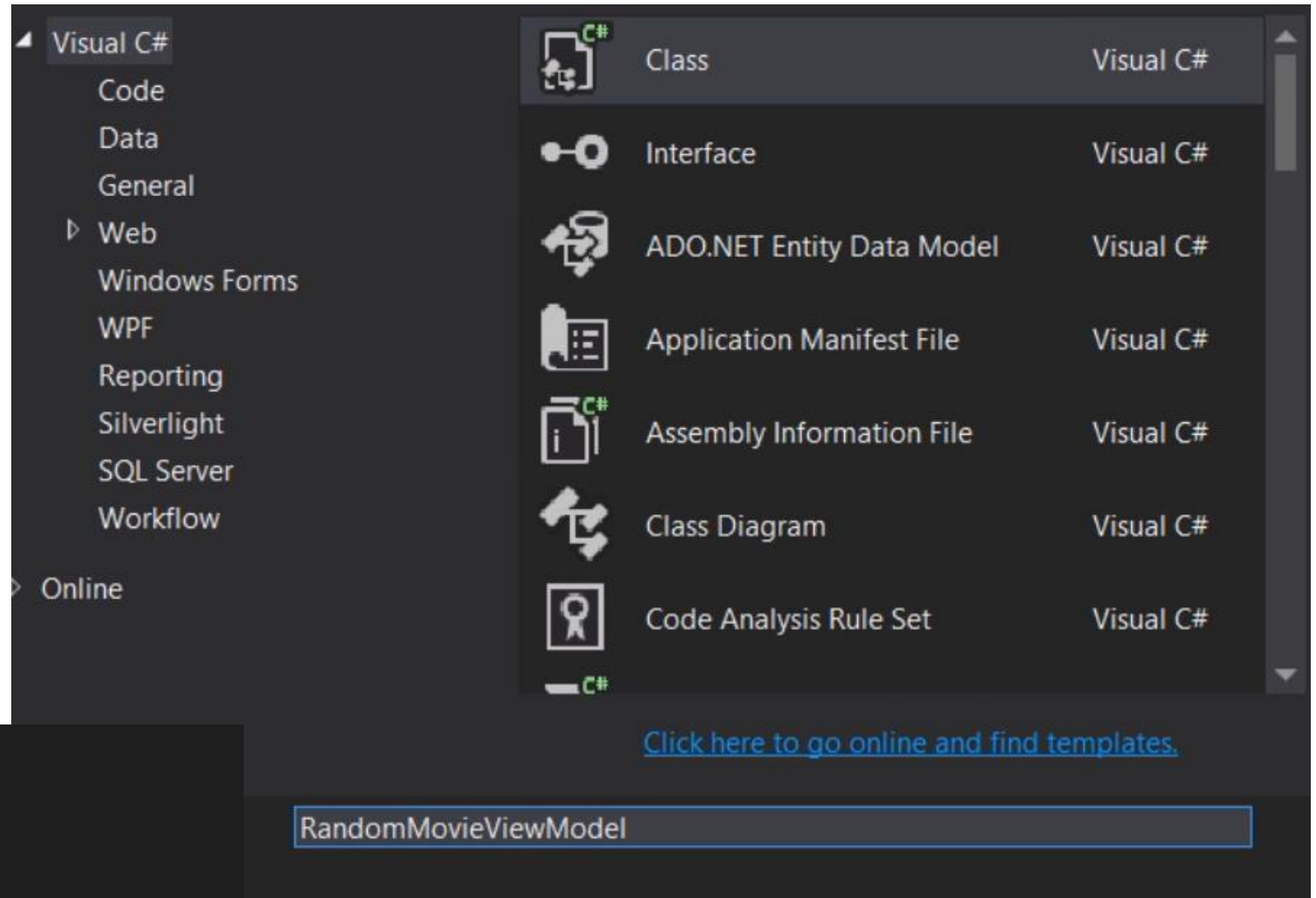


```
namespace Vidly.Models
{
    public class Customer
    {
        public int Id { get; set; }
        public string Name { get; set; }
    }
}
```

New Folder -> ViewModels



New Class -> ใน Folder ViewModels ชื่อ RandomMovieViewModel



```
namespace Vidly.ViewModels
{
    public class RandomMovieViewModel
    {
        public Movie Movie { get; set; }
        public List<Customer> Customers { get; set; }
    }
}
```

```

public class MoviesController : Controller
{
    // GET: Movies/Random
    public ActionResult Random()
    {
        var movie = new Movie() { Name = "Shrek!" };
        var customers = new List<Customer>
        {
            new Customer { Name = "Customer 1" },
            new Customer { Name = "Customer 2" }
        };

        var viewModel = new RandomMovieViewModel
        {
            Movie = movie,
            Customers = customers
        };

        return View(viewModel);
    }
}

```

```

Customer.cs • MoviesController.cs • Random.cshtml • X RandomMovieViewMo
@model Vidly.ViewModels.RandomMovieViewModel
@{
    ViewBag.Title = "Random";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>@Model.Movie.Name</h2>
@Model.Customers

```

```
ViewBag.Title = "Random";  
Layout = "~/Views/Shared/_Layout.cshtml";  
}  
  
<h2>@Model.Movie.Name</h2>  
  
@if (Model.Customers.Count == 0)  
{  
    <text>No one has rented this movie before.</text>  
}  
else  
{  
    <ul>  
        @foreach (var customer in Model.Customers)  
        {  
            <li>@customer.Name</li>  
        }  
    </ul>  
}
```

```
Layout = "~/Views/Shared/_Layout.cshtml";
```

```
@{
```

```
var className = Model.Customers.Count > 5 ? "popular" : null;
```

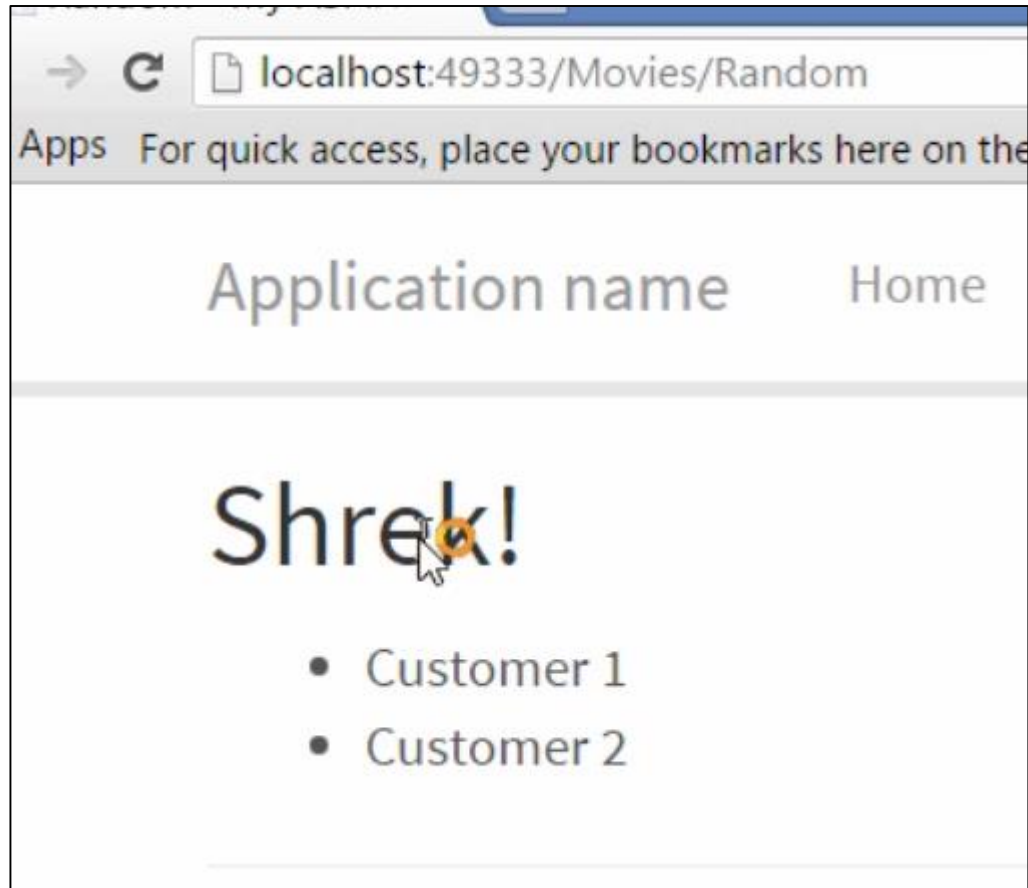
```
}
```

```
<h2 class="@className">@Model.Movie.Name</h2>
```

```
@if
```

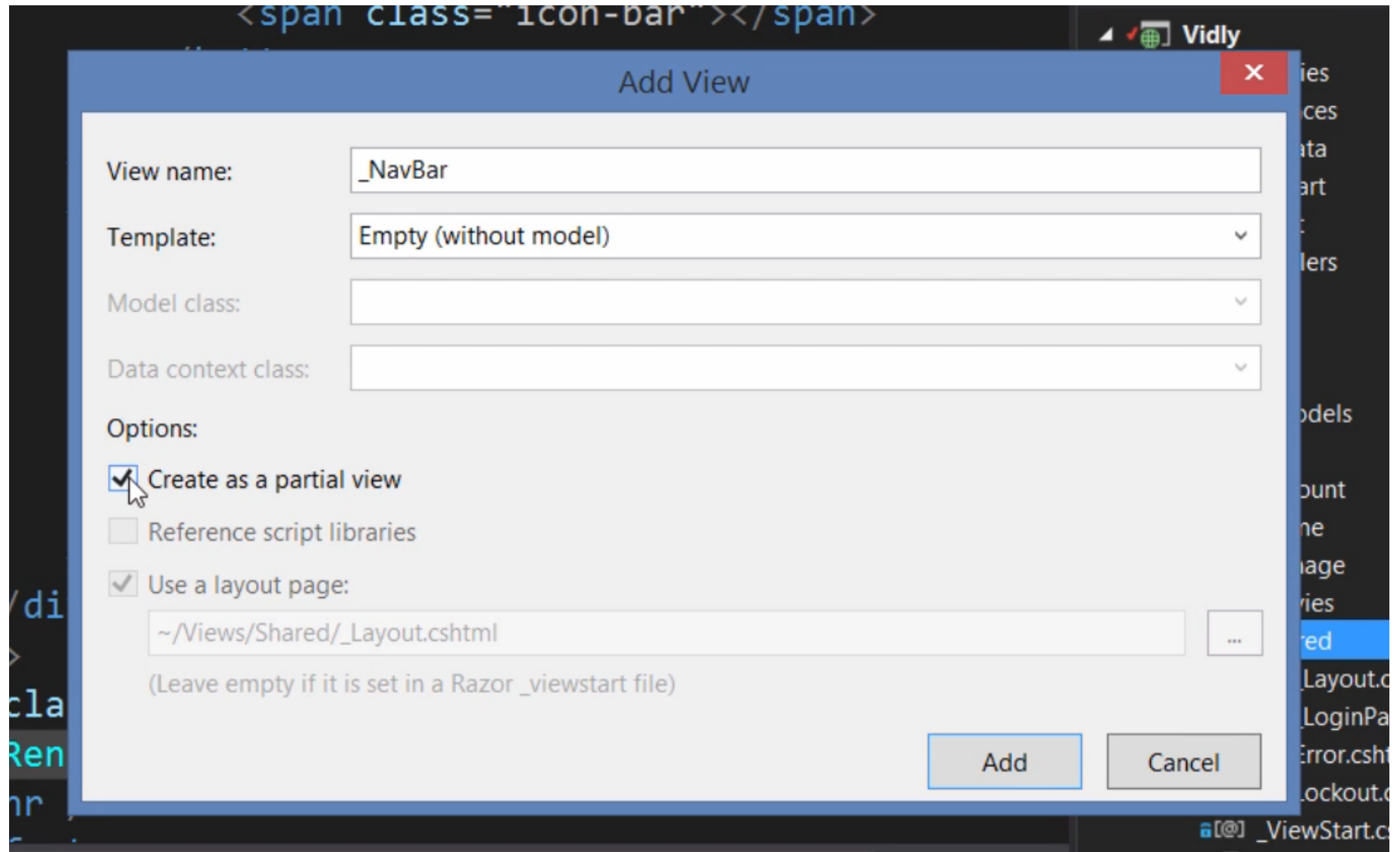
```
(Model.Customers.Count == 0)
```

```
{
```



Partial Views

Add View → Layout ชื่อ _NavBar



Copy Code ส่วนนี้ ไปไว้ใน Navbar

_Layout.cshtml X _NavBar.cshtml

```
</head>
<body>
  <div class="navbar navbar-inverse navbar-fixed-top">...</div>
  <div class="container body-content">
    @RenderBody()
    <hr />
    <footer>
      <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
    </footer>
  </div>

  @Scripts.Render("~/bundles/jquery")
  @Scripts.Render("~/bundles/bootstrap")
  @RenderSection("scripts", required: false)
</body>
</html>
```

```
<div class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      @Html.ActionLink("Application name", "Index", "Home", new { area = "" },
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li>@Html.ActionLink("Home", "Index", "Home")</li>
        <li>@Html.ActionLink("About", "About", "Home")</li>
        <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
      </ul>
      @Html.Partial("_LoginPartial")
    </div>
  </div>
```


@model Vidly.ViewModels.RandomMovieViewModel

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>@ViewBag.Title - My ASP.NET Application</title>

@Styles.Render("~/Content/css")

@Scripts.Render("~/bundles/modernizr")

</head>

<body>

@Html.Partial("_NavBar", Model.Movie)

<div class="container body-content">

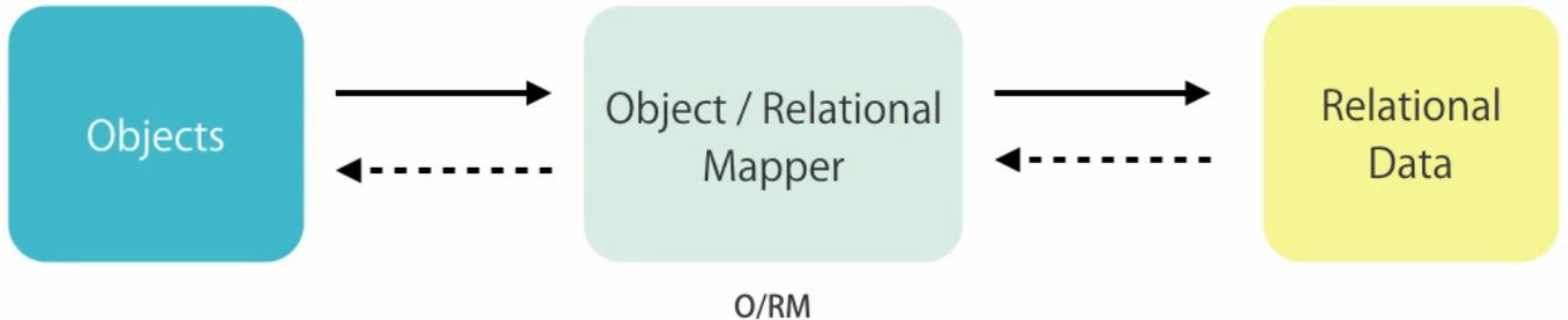
@RenderBody()

<hr />

<footer>

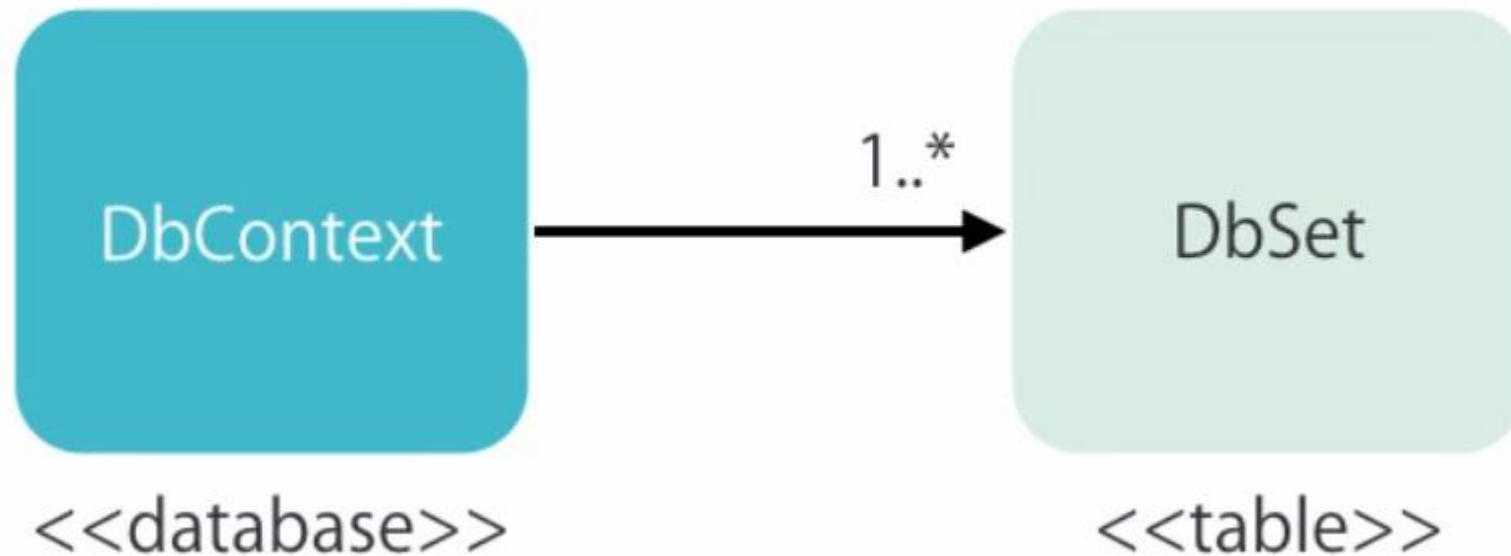
Working with Data

Entity Framework



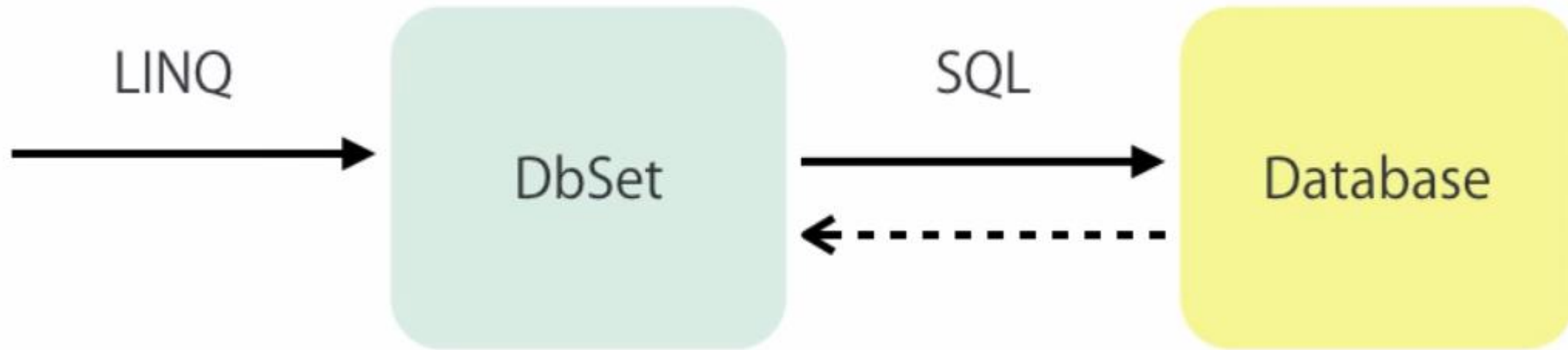
Working with Data

Entity Framework



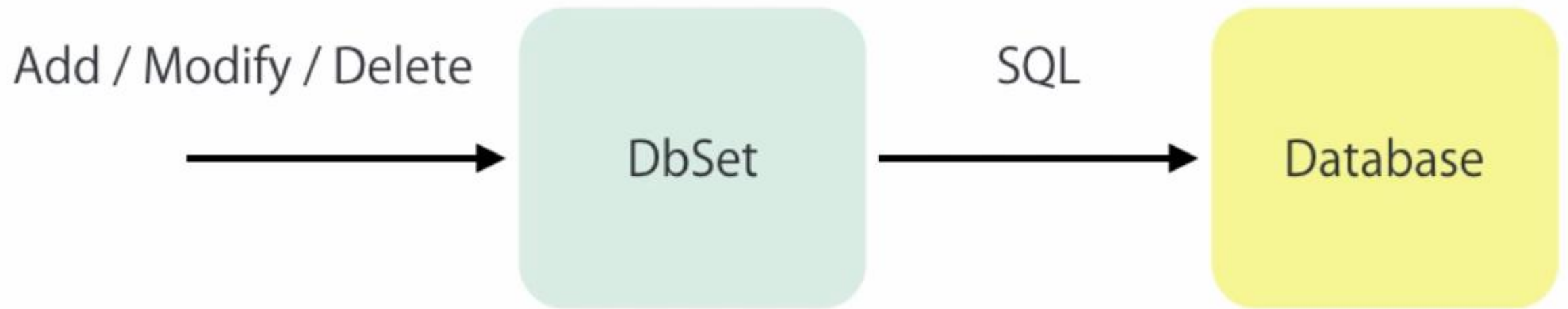
Working with Data

LINQ




Working with Data

LINQ



Working with Data

WORKFLOW



The diagram consists of a light gray rectangular background. Inside this background are two rounded square boxes. The box on the left is yellow and contains the text 'Database First'. The box on the right is teal and contains the text 'Code First'.

Database
First

Code
First

Working with Data

DbFirst WORKFLOW



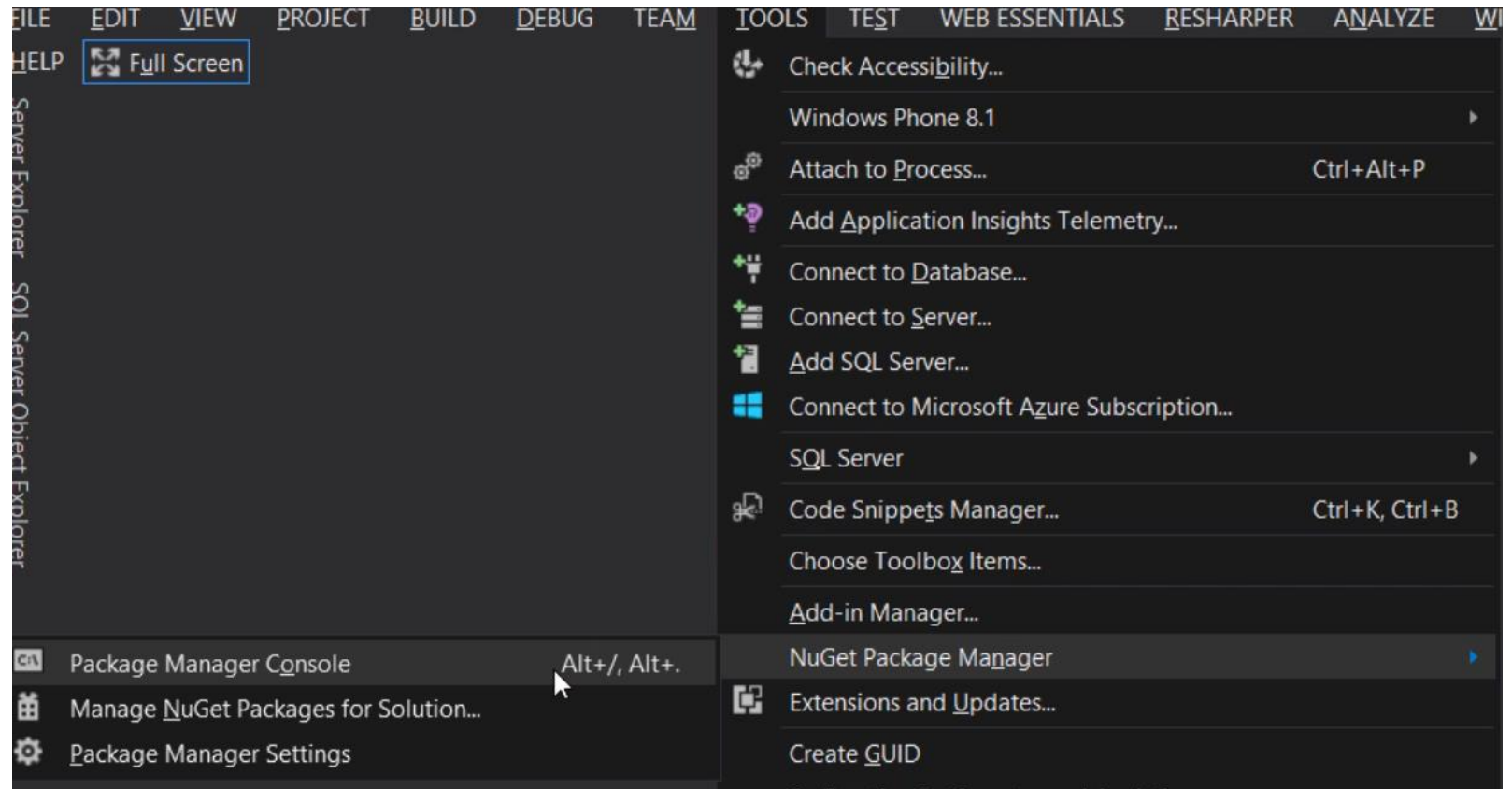
CodeFirst WORKFLOW



```
namespace Vidly.Models
{
    public class Customer
    {
        public int Id { get; set; }
        public string Name { get; set; }
    }
}
```


Code-first Migrations : แปลงจากโค้ดไปสู่ฐานข้อมูล

เปิด NuGet Package Manager Console

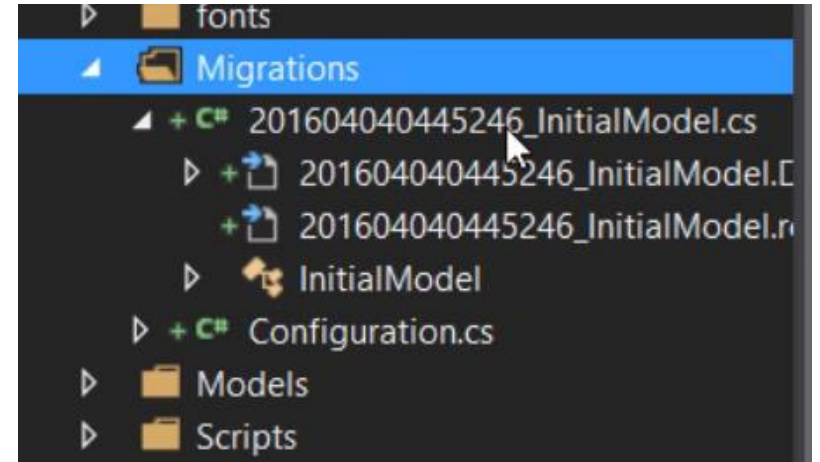


1.เปิดการ Migration

```
PM> enable-migrations|
```

2.Add Migration

```
PM> add-migration InitialModel
```



เปิดไฟล์ InitialModel และเพิ่มคลาส Customer

```
public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
{
    public DbSet<Customer> Customers { get; set; }

    public ApplicationDbContext()
        : base("DefaultConnection", throwIfV1Schema: false)
    {
    }
}
```

Add Migration InitialModel -force (เพื่ออัปเดต)

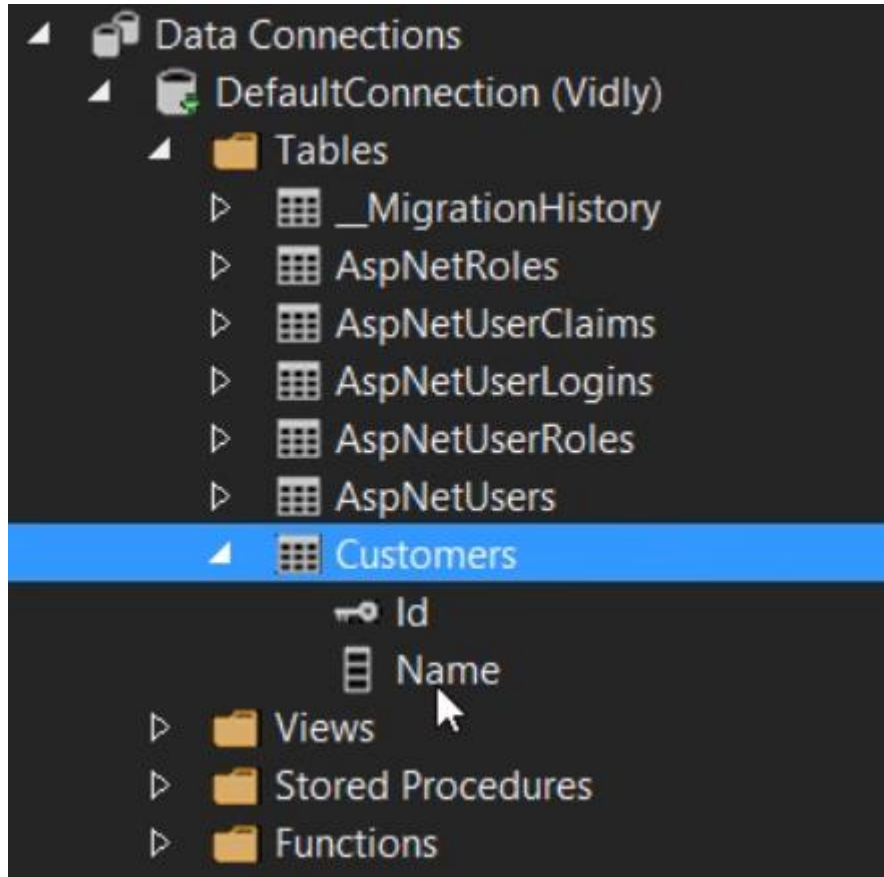
```
PM> add-migration InitialModel -force  
Re-scaffolding migration 'InitialModel'.  
PM>
```



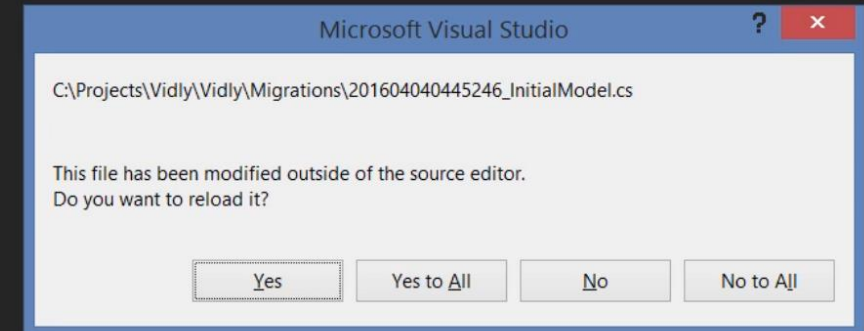
Add Migration InitialModel -force (เพื่ออัปเดต)

(เพื่ออัปเดต)

```
PM> update-database
```



```
PM> add-migration InitialModel -force
Re-scaffolding migration 'InitialModel'.
PM>
```



DomainModel

Customer

IsSubscribedToNewsletter
MembershipType

```
namespace Vidly.Models
{
    public class Customer
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public bool IsSubscribedToNewsletter { get; set; }
    }
}
```

เพิ่ม Seed ชื่อ AddIsSubscribedToCustomer

```
PM> add-migration AddIsSubscribedToCustomer
```

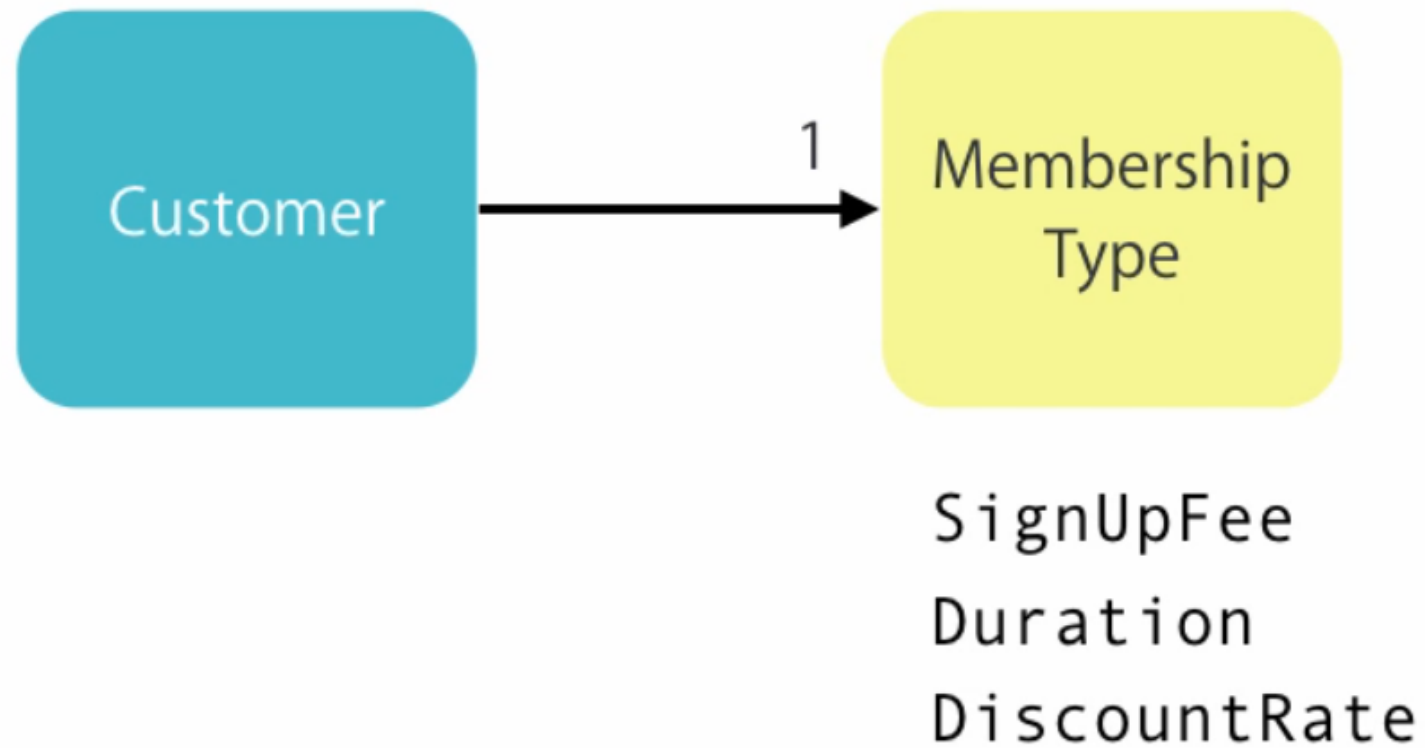
(เพื่ออัปเดต)

```
PM> update-database
```

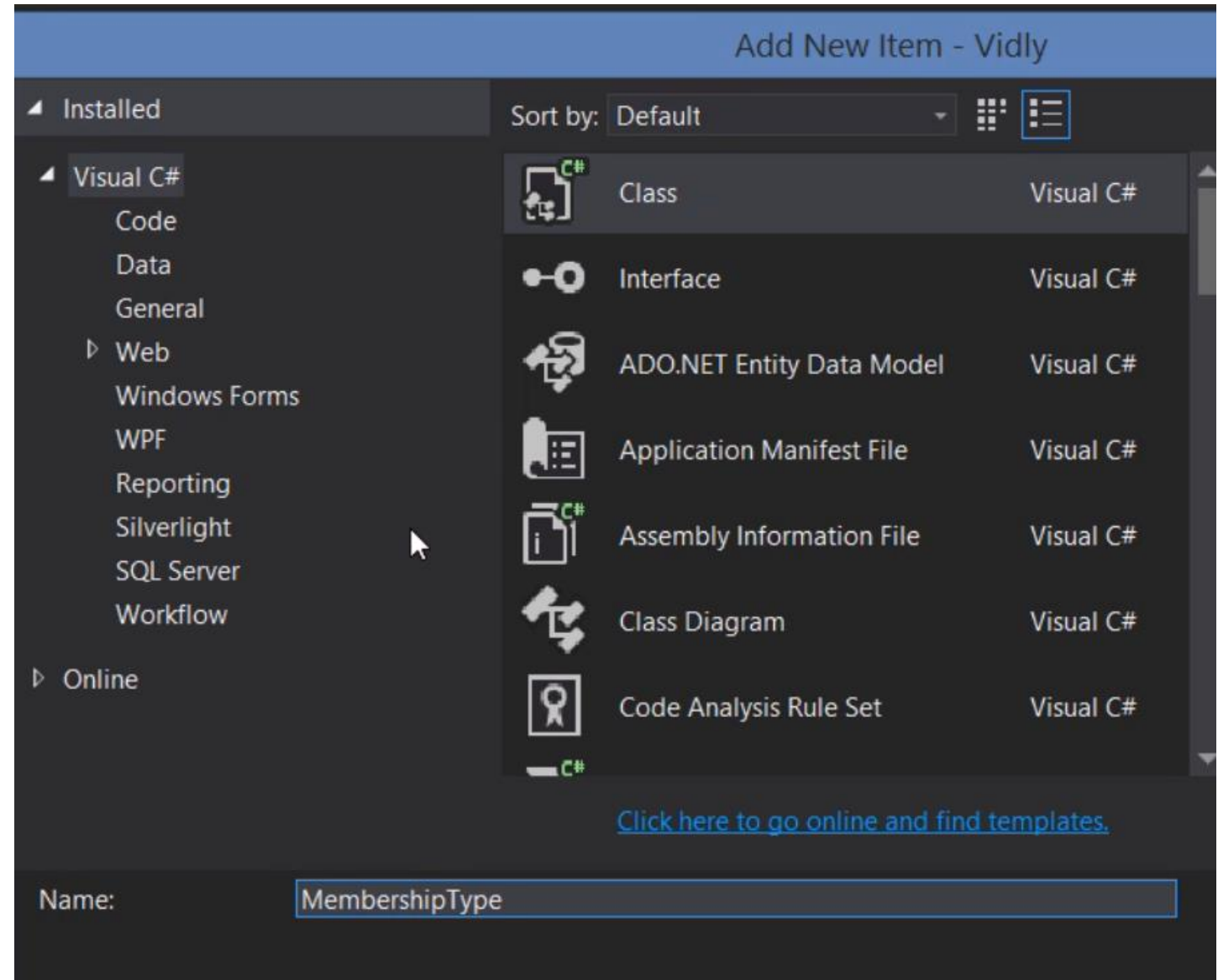
Rules

Membership	Sign Up Fee	Discount
Pay as You Go	Free	0%
Monthly	\$30	10%
Quarterly	\$90	15%
Annual	\$300	20%

Domain Model



New Class = MembershipType



MembershipType



```
namespace Vidly.Models
{
    public class MembershipType
    {
        public byte Id { get; set; }
        public short SignUpFee { get; set; }
        public byte DurationInMonths { get; set; }
        public byte DiscountRate { get; set; }
    }
}
```

Customer

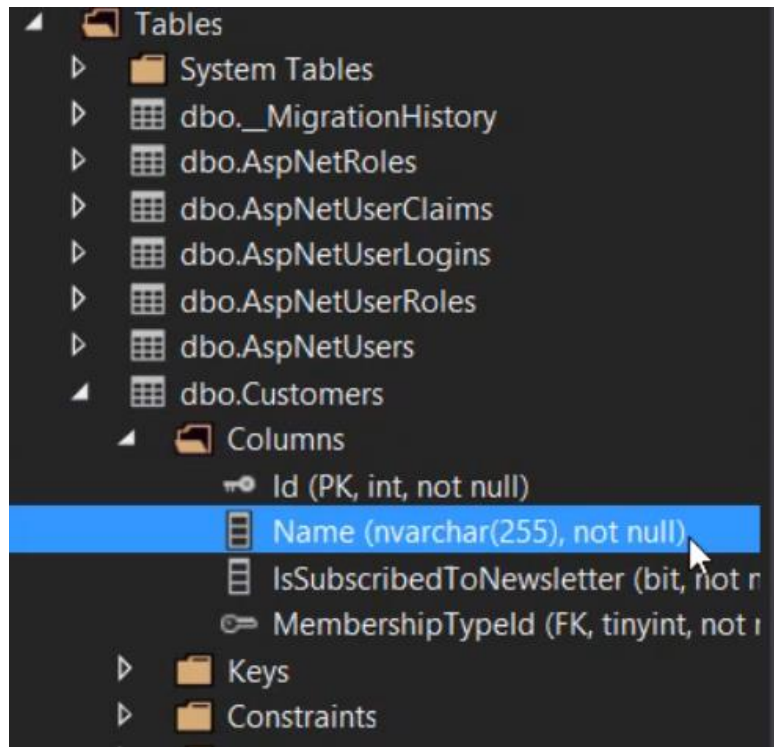


```
namespace Vidly.Models
{
    public class Customer
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public bool IsSubscribedToNewsletter { get; set; }
        public MembershipType MembershipType { get; set; }
        public byte MembershipTypeId { get; set; }
    }
}
```

```
PM> add-migration AddMembershipType
```

```
PM> update-database
```

เพิ่มเงื่อนไขการรับค่าลงฐานข้อมูลคอลัมน์ Name ว่าเป็นข้อความไม่เกิน 255



```
namespace Vidly.Models
{
    public class Customer
    {
        public int Id { get; set; }
        [Required]
        [StringLength(255)]
        public string Name { get; set; }
        public bool IsSubscribedToNewsletter { get; set; }
        public MembershipType MembershipType { get; set; }
        public byte MembershipTypeId { get; set; }
    }
}
```

```
PM> add-migration ApplyAnnotationsToCustomerName
```

```
PM> update-database
```

```

public class CustomersController : Controller
{
    private ApplicationDbContext _context;

    public CustomersController()
    {
        _context = new ApplicationDbContext();
    }

    protected override void Dispose(bool disposing)
    {
        _context.Dispose();
    }

    public ActionResult Index()
    {
        var customers = _context.Customers.ToList();
        return View(customers);
    }
}

```

← สร้างคลาสเชื่อมต่อฐานข้อมูล

← สร้างคลาส **Constructor**

← ดึงรายการข้อมูล

```

public IActionResult Index()
{
    var customers = _context.Customers.ToList();

    return View(customers);
}

public IActionResult Details(int id)
{
    var customer = _context.Customers.SingleOrDefault(c => c.Id == id);

    if (customer == null)
        return HttpNotFound();

    return View(customer);
}

```

	Id	Name	IsSubscribe...	Membershi...
	1	John Smith	False	1
	2	Mary Willia...	True	2
➤	NULL	NULL	NULL	NULL

การกรอกรายการข้อมูล


```

public ActionResult Index()
{
    var customers = _context.Customers.ToList();

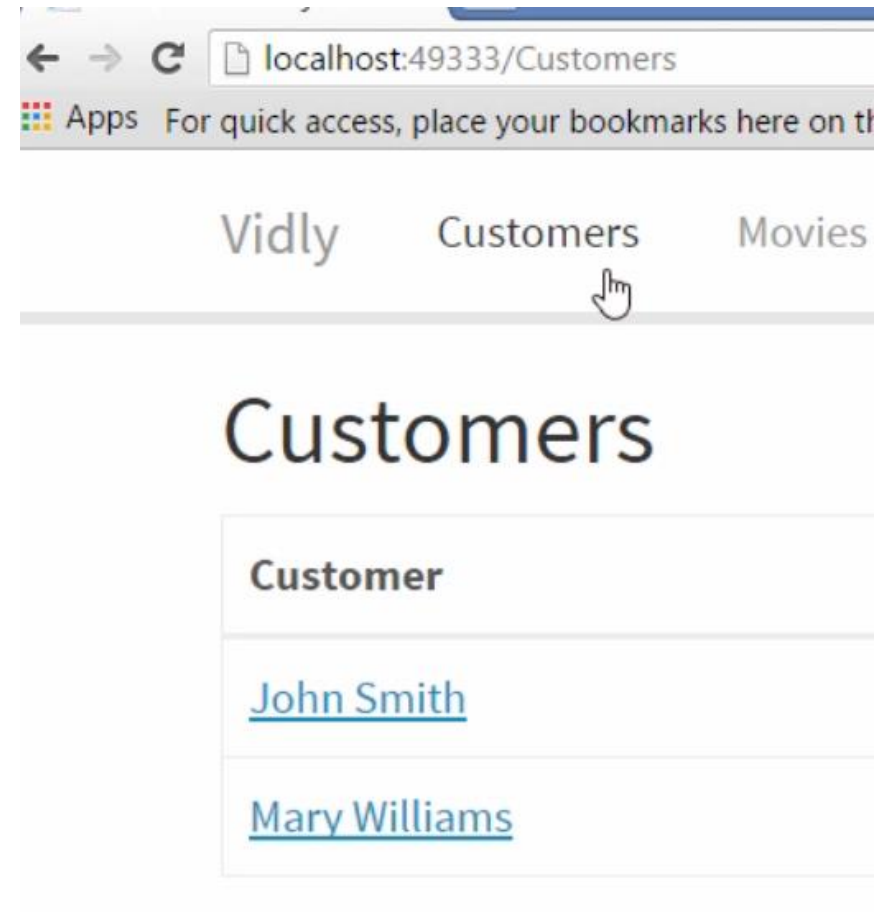
    return View(customers);
}

public ActionResult Details(int id)
{
    var customer = _context.Customers.SingleOrDefault(c => c.Id == id);

    if (customer == null)
        return HttpNotFound();

    return View(customer);
}

```



	Id	Name	IsSubscribe...	Membershi...
	1	John Smith	False	1
	2	Mary Willia...	True	2
	NULL	NULL	NULL	NULL

การกรอกรายการข้อมูล