**참고자료**

<https://www.youtube.com/watch?v=z57FoOwn8Zo&list=PLtht1_et-35DI4cRVKEUUaQAuT3FW2jet&index=7>

**경로의 해석**



1번 id 를 가지고 있는 member 를 보여달라

* / 이후가 없다면 HomeComtroller 가 실행
* Action 이후가 없다면 Index 액션이 실행
* 이 옵션은 기본옵션이고 RouteConfig 로 설정을 변경할 수 있다.

컨트롤러는 주소를 가지고 해석한다

컨트롤러는 C# 클래스다.

HomeController.cs

Home : 컨트롤러 이름

HomeController : 클래스 이름

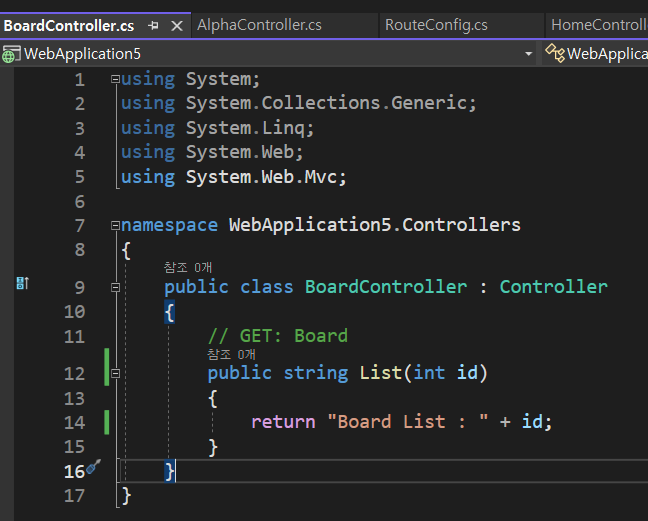
HomeController.cs : 파일의 이름

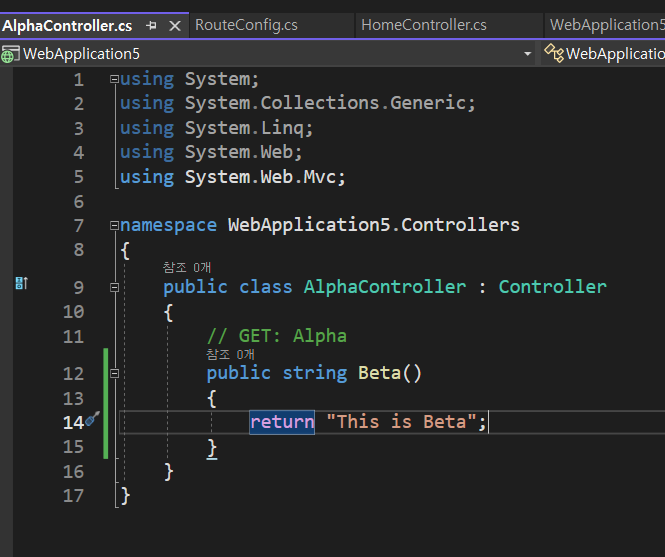
Action

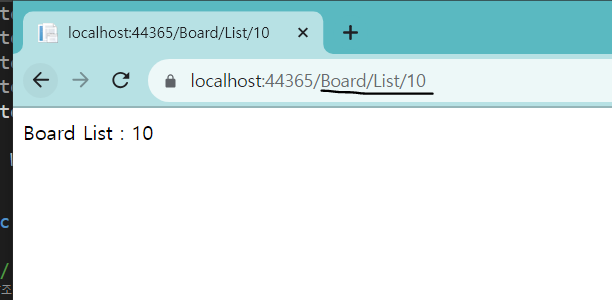
클래스에 속한 메서드

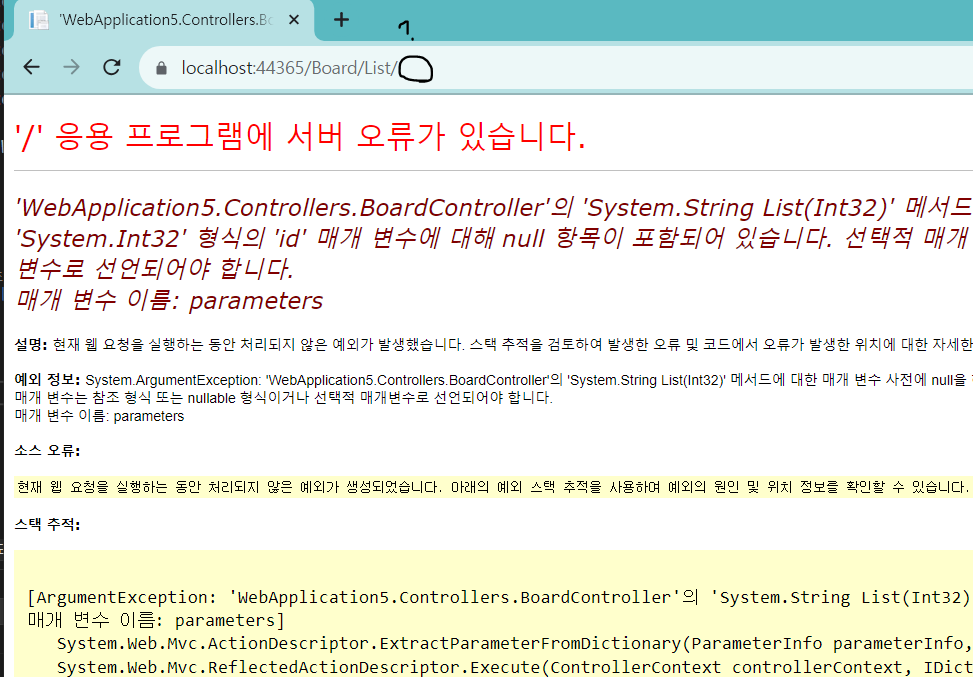
Id

메서드에 올 수 있는 매개변수









**App\_Start\RouteConfig.cs**

namespace WebApplication5

{

public class RouteConfig

{

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "Alpha", action = "Beta", id = UrlParameter.Optional }

);

}

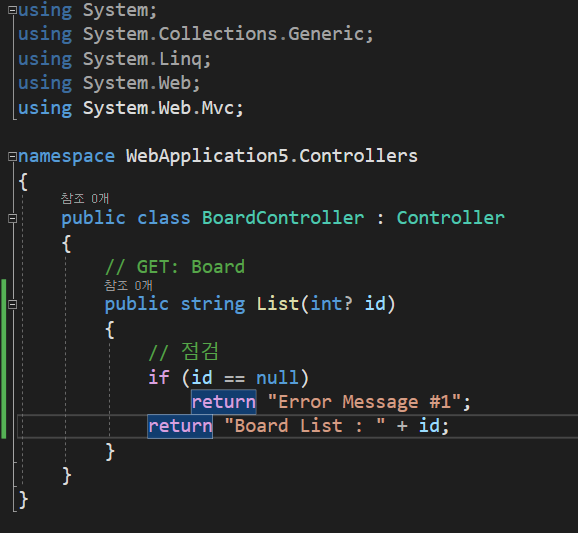
}

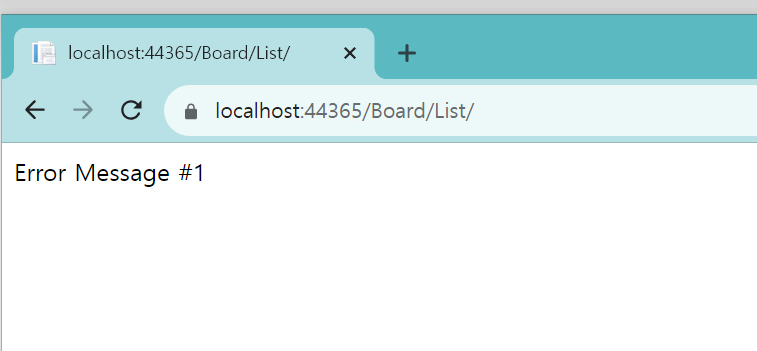
}

**컨트롤러의 역할**

1. 클라이언트 수신
2. 지시
3. 점검





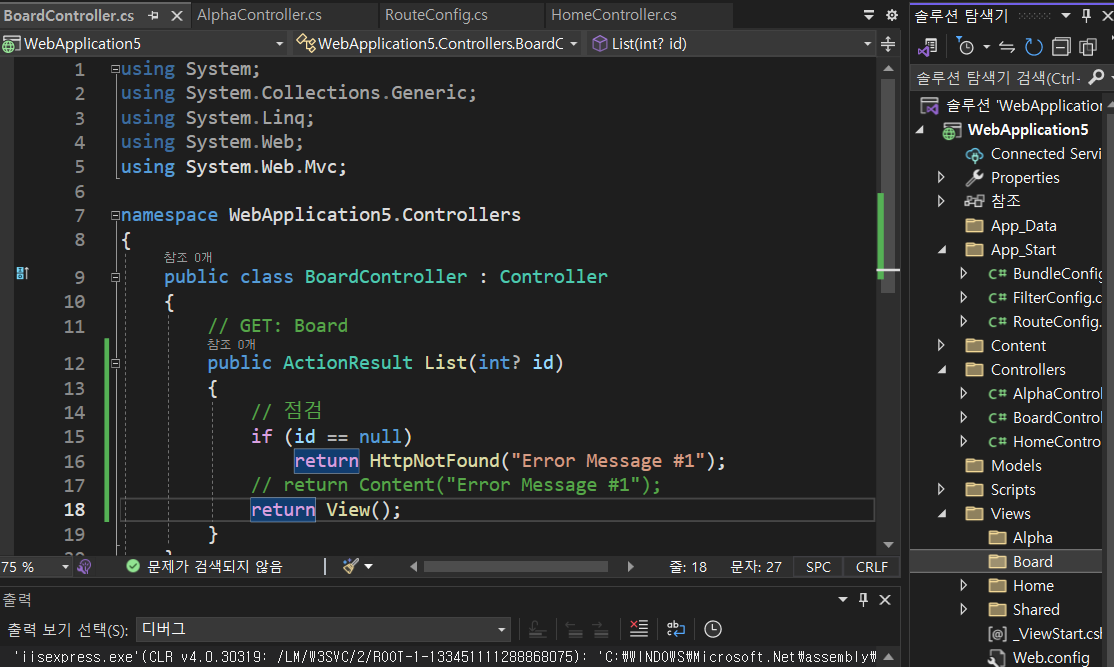


**뷰**

출력

확장자 **cshtml**, aspx, ascx, vbhtml

뷰엔진 : Razor – c# 문법 사용

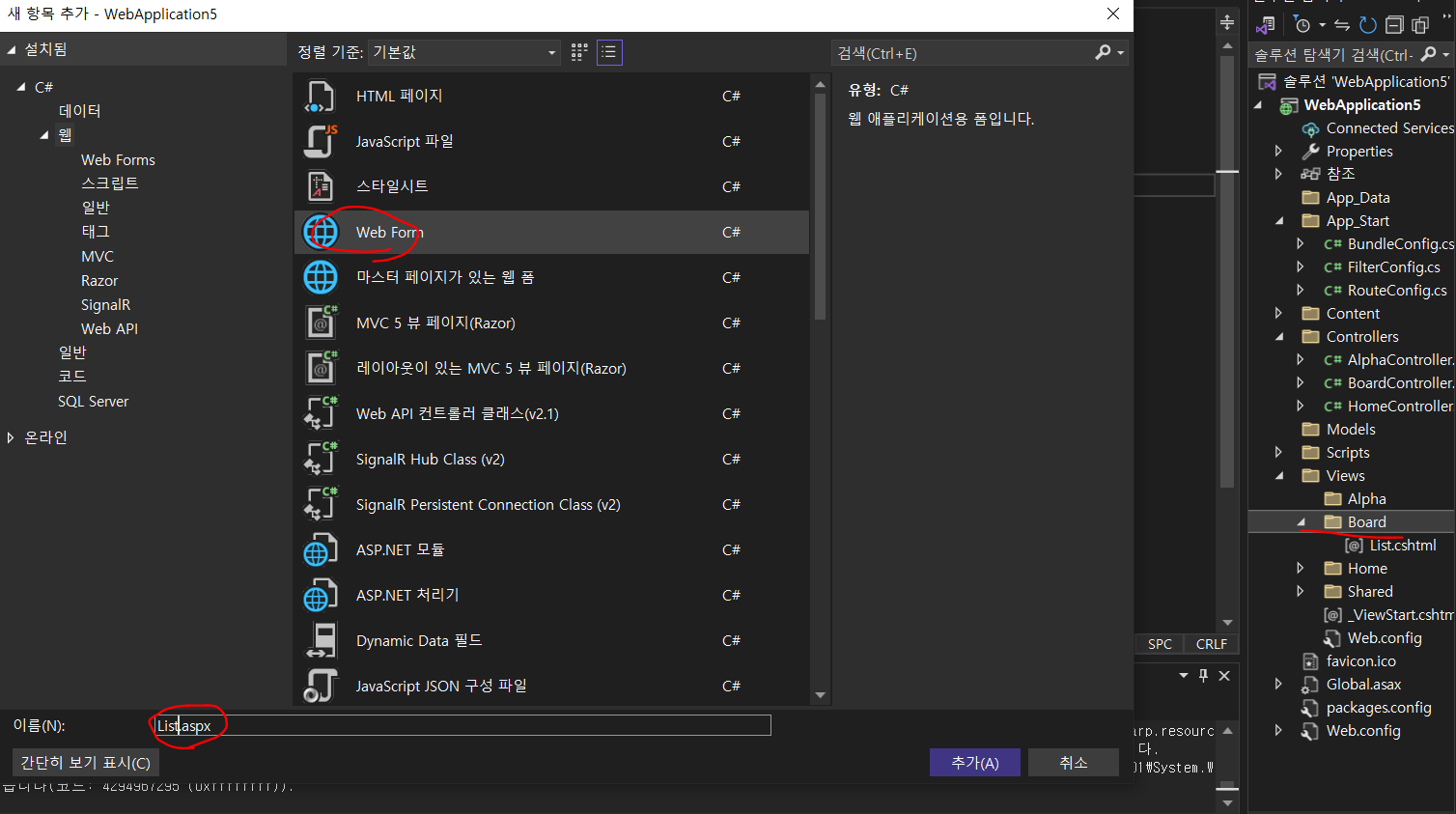


View 의 Alpha 폴더 내의 List.cshtml 파일을 찾아서 리턴한다. (지금은 안만들어서 없다)

view 를 추가하는 방법

메소드에 우클릭해서 뷰추가.

뷰 파일은 컴파일을 할 필요가 없다 (새로고침만 해주면 된다)



우클릭해서 webform 추가 (aspx)

오류가 있으나, 우선적으로 aspx 를 view 로 사용하는 것을 알 수 있다.

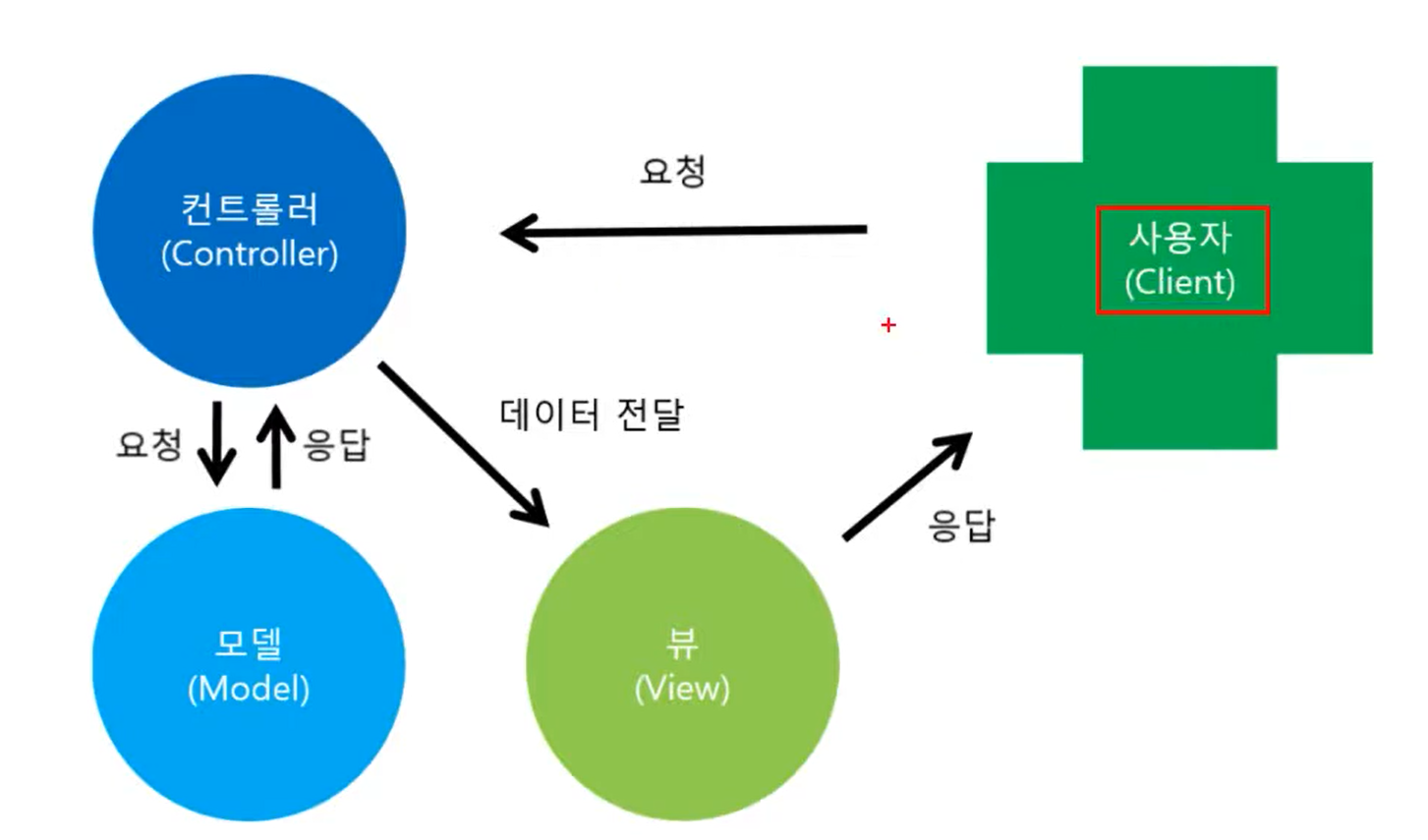
**모델**

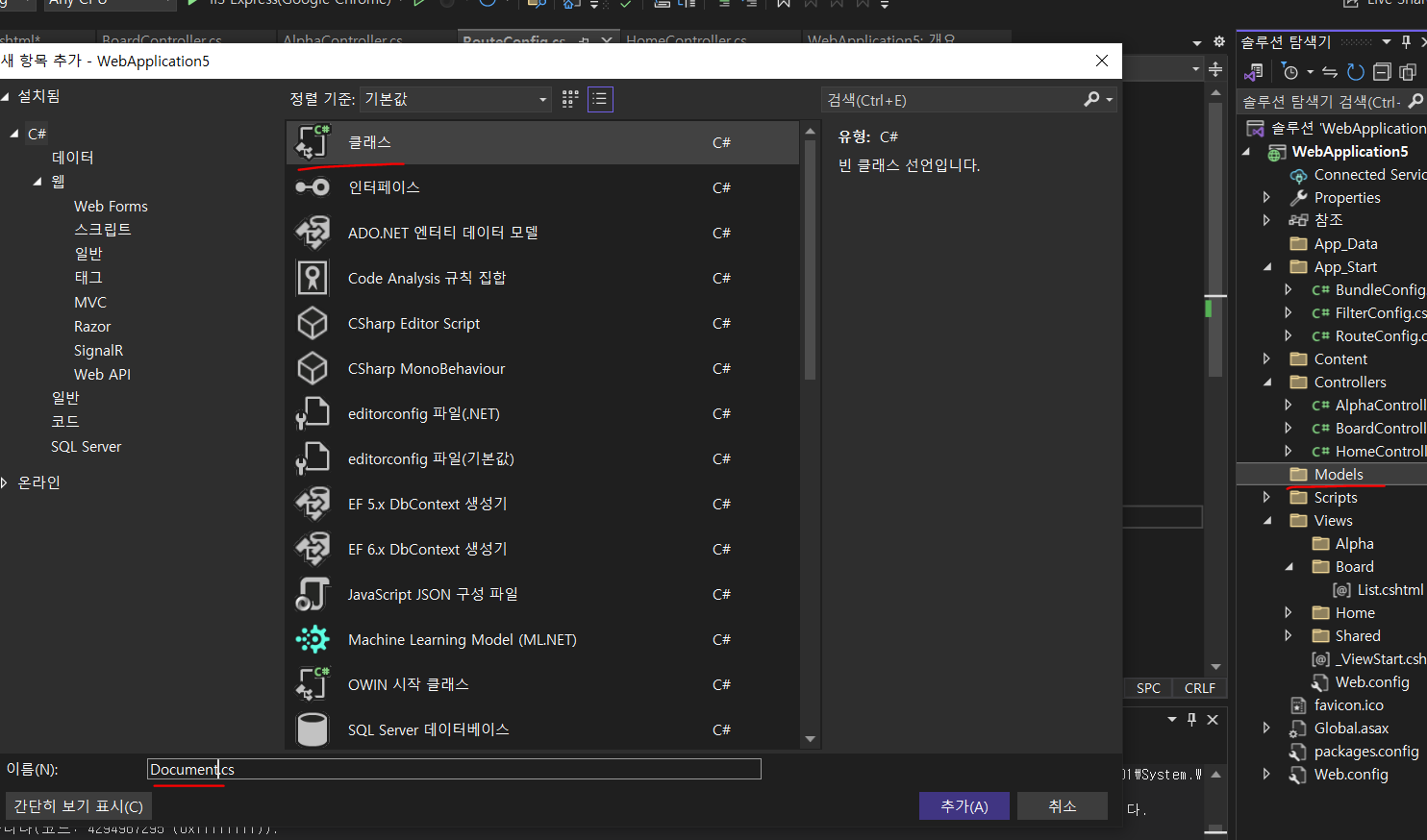
데이터와 관련된 역할을 함

1. 데이터의 속성 정의 (C# 의 Property)
2. 데이터의 처리 (Business Logic 을 처리)
   1. 회원의 정보를 가져온다던지
   2. 주로 DB 에 관한 처리

모델의 생성 (cs 파일)

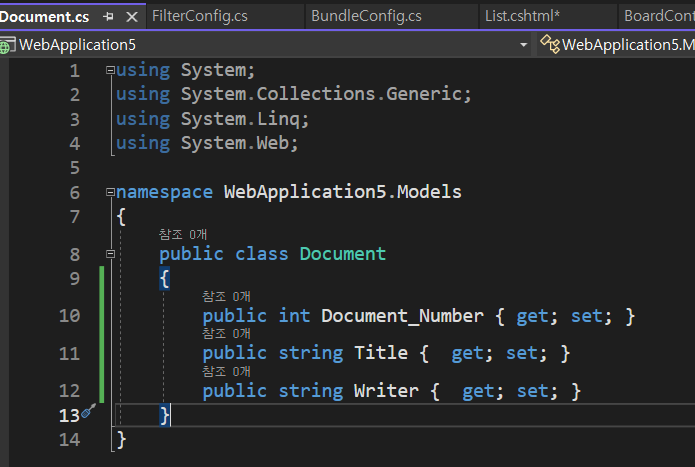
Models 폴더 내에서 클래스 생성





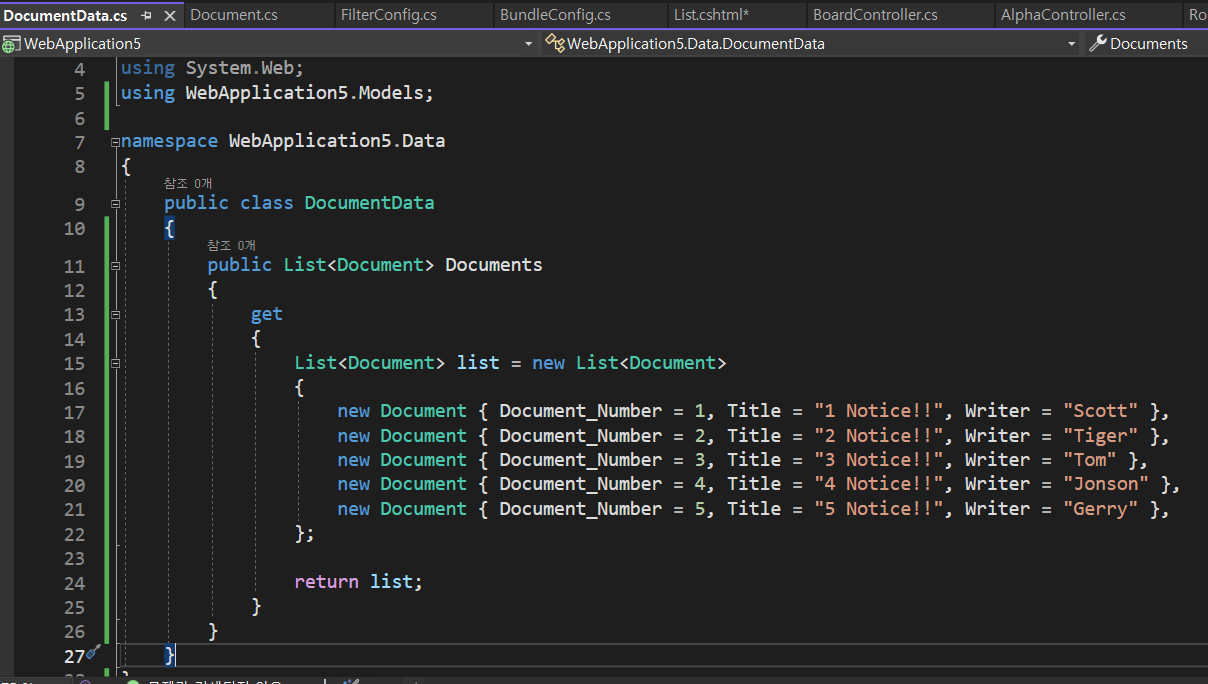
property 정의

prop + tab + tab

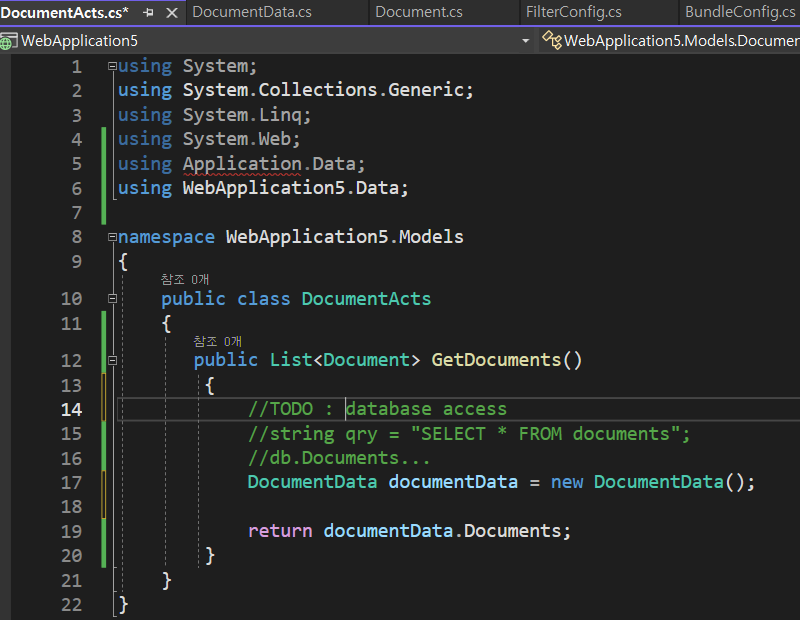


데이터를 만들고 엑세스 해보자.

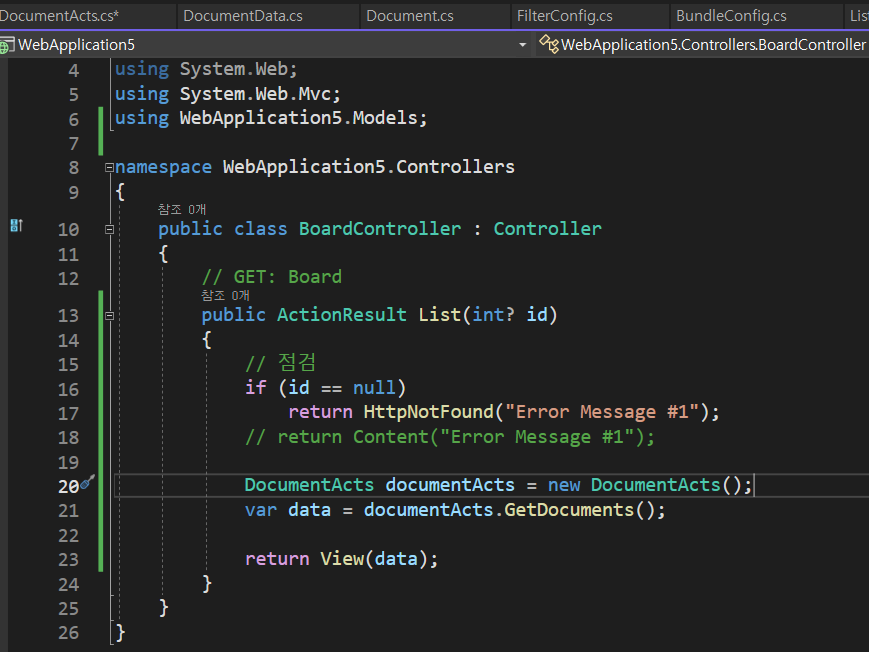
* DB 에 접근하는것이 맞지만, 복잡하므로 Class 로 만들어서 접근해보자
* 새 폴더를 만들고
* DocumentData.cs 파일을 만들고
* 데이터 추가



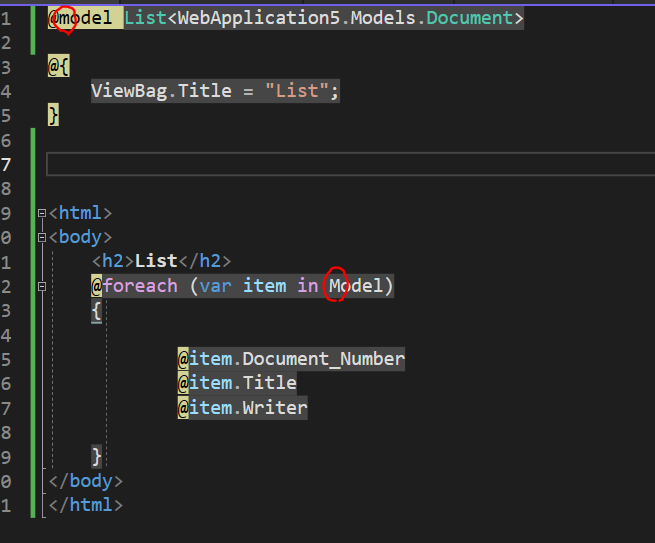
데이터처리를 위한 클래스 추가



이제 이걸 컨트롤러에서 사용하게 해보자.



이제 이것을 보여줄 view 를 만들어 보자 (Razer 를 사용하자)



ViewBag & ViewData

controller 에서 view 에게 data 를 전달할 수 있는 방법



하나의 데이터만 전달할 수 있기때문에 문제가 있다 .



여러개의 데이터를 전달할 수 있다.