بسم الله الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

اضافه کردن models و controller

مدرس : مهندس افشین رفوآ

دوره آموزشی Web API

در این قسمت، ما کلاس های model را برای مشخص کردن ماهیت دیتابیس تعریف می کنیم. سپس شما می توانید Controller های Web API که عملیات CRUD روی آن موجودیت ها را انجام می دهد اضافه کنید.

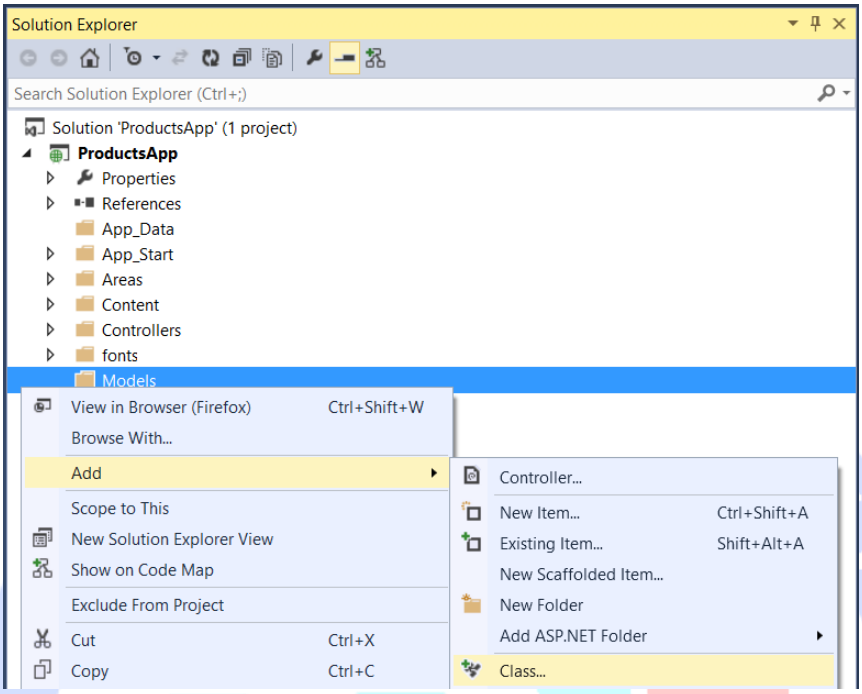
اضافه کردن کلاس های Model

در این مقاله ما با استفاده از Code First و EntityFramework دیتابیس خود را می سازیم. با Code First شما می توانید کلاس های C# را که با جداول دیتابیس مطابقت دارند با EF بسازید.

ما کار خود را با تعریف object ها به عنوان POCO شروع می کنیم. در ادامه POCO های زیر را خواهیم ساخت:

- نویسنده
- کتاب

در solution explorer روی فولدر models کلیک راست کنید و Add را انتخاب کنید و سپس Class را بزنید و نام آن را Author بگذارید.



قطعه کد زیر را با کدهای درون Author.cs جایگزین کنید.

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
namespace BookService.Models
{
    public class Author
    {
        public int Id { get; set; }
        [Required]
        public string Name { get; set; }
    }
}
```

کلاس دیگری با نام Book بسازید و به صورت زیر کد نویسی کنید.

```
using System.ComponentModel.DataAnnotations;
namespace BookService.Models
{
    public class Book
    {
        public int Id { get; set; }
        [Required]
        public string Title { get; set; }
        public int Year { get; set; }
        public decimal Price { get; set; }
        public string Genre { get; set; }
        // Foreign Key
        public int AuthorId { get; set; }
        // Navigation property
        public Author Author { get; set; }
    }
}
```

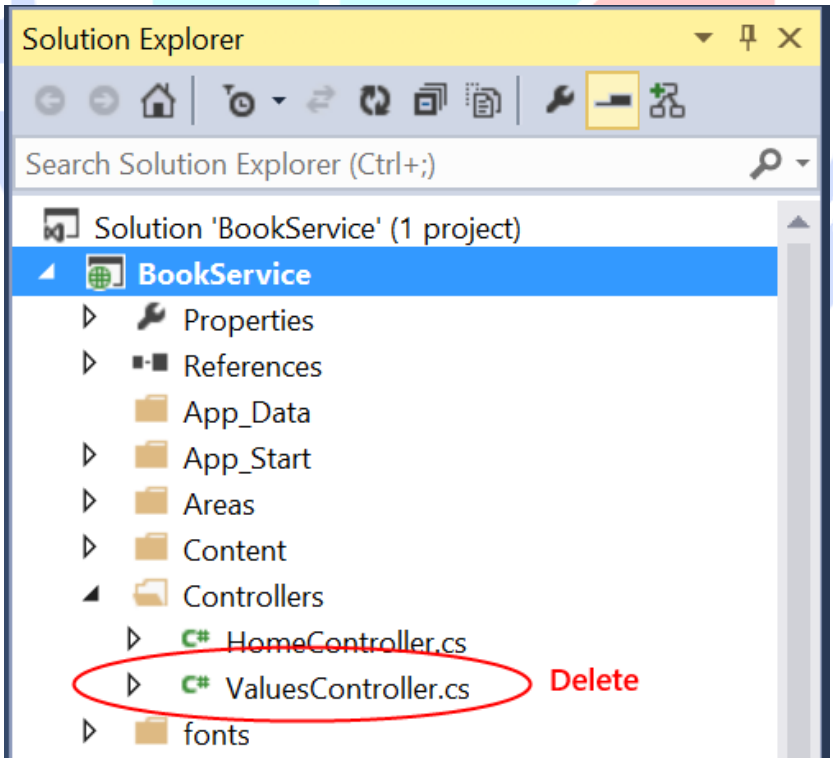
Entityframework از این مدل ها برای ساخت جداول دیتابیس استفاده می کند. برای هر مدل،خاصیت Id در نقش کلید اصلی جدول دیتابیس می باشد.

در کلاس book، AuthorId یک کلید خارجی در جدول Author تعریف می کند.(برای سادگی، فرض می کنیم هر book تنها یک author دارد.) همچنین کلاس book شامل خاصیت جهت یابی (navigation)برای Author مربوط می باشد. شما می توانید از خاصیت navigation برای دسترسی به Author مربوط استفاده کنید.

اضافه کردن Comtroller های Web API

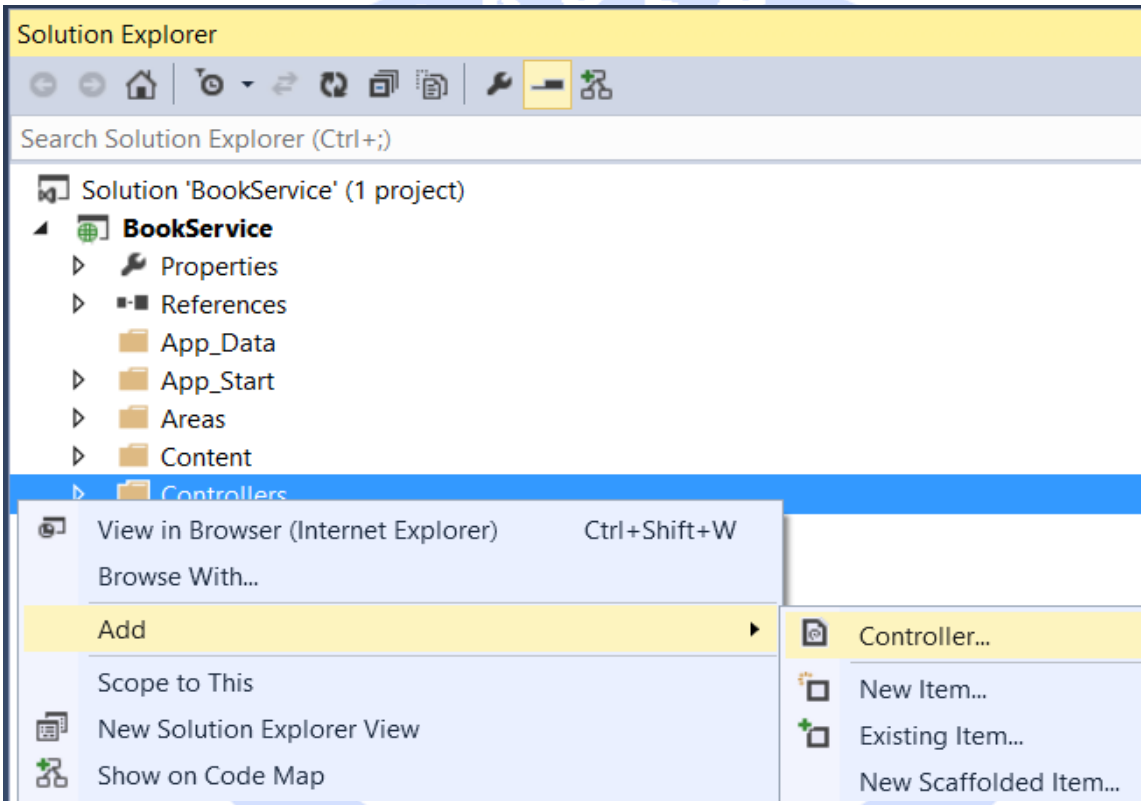
در این قسمت، ما Controller هایی اضافه می کنیم که عملیات CRUD(ساختن،خواندن،به روز رسانی و حذف کردن) را پشتیبانی کنند. Controller از Entity Framework برای ارتباط با لایه دیتابیس استفاده می کنیم.

ابتدا، شما می توانید فایل Controllers/ValuesController.cs را حذف کنید. در این فایل یک مثال از Web API Controller موجود است که در این مقاله به آن نیاز نداریم.

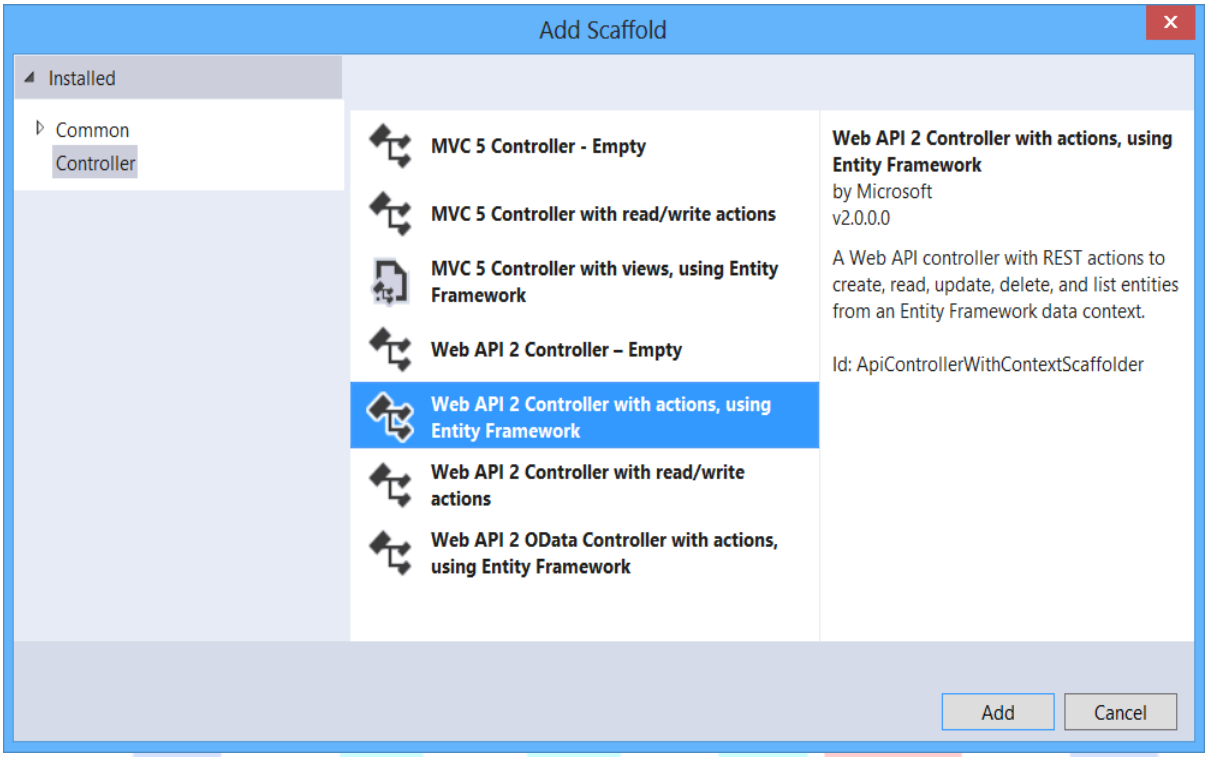


در مرحله بعد، پروژه را Build کنید. روش Scaffold کردن Web API از reflection برای پیدا کردن کلاس های model استفاده می کند و برای همین به کامپایل نیاز دارد.

در solution explorer روی فولدر Models کلیک راست کنید و ابتدا Add و سپس Controller را انتخاب کنید.



در پنجره Add Scaffold، Web API 2 Controller with actions, using Entity Framework را انتخاب کرده و Okکنید.



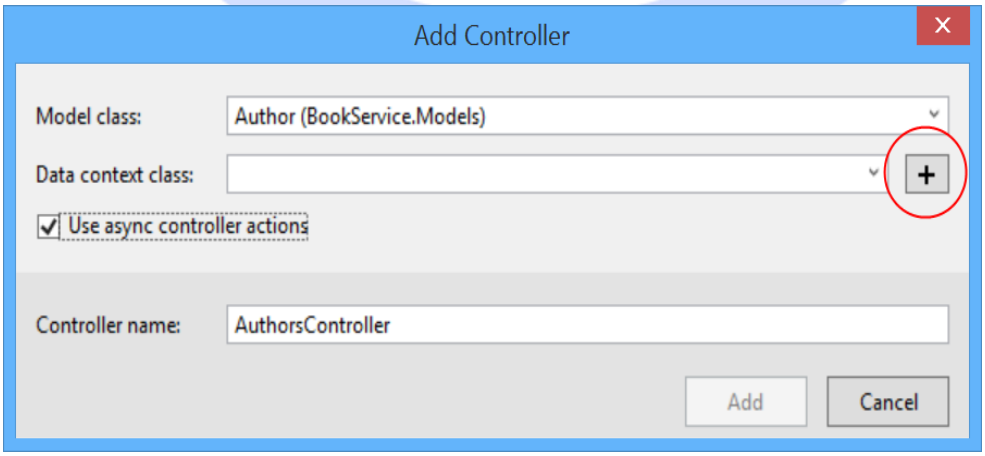
در پنجره Add Controller مراحل زیر را انجام دهید:

1– در لیست Model Class، کلاس Author را انتخاب کنید(اگر این کلاس را پیدا نکردید، دوباره به برنامه خود بازگردید و دوباره Build کنید.)

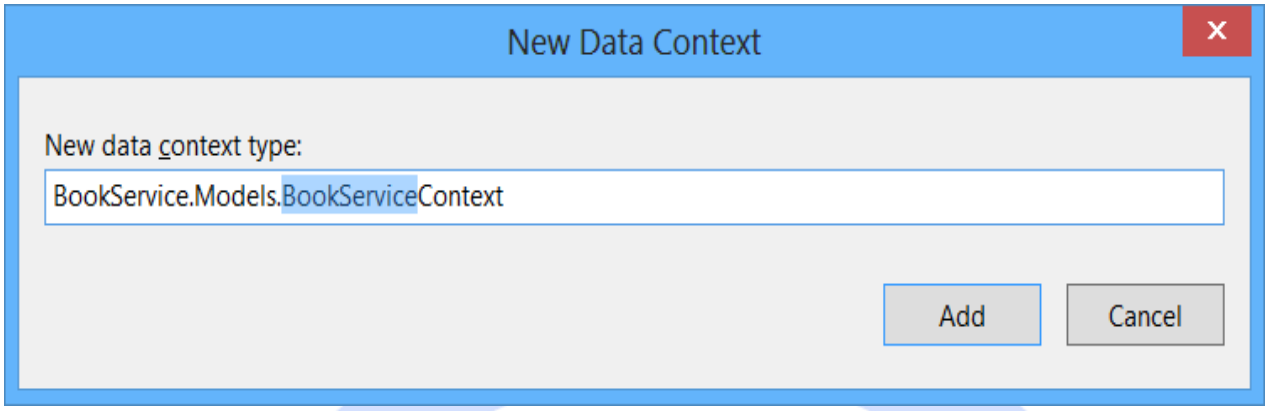
2– گزینه Use async controller actions را انتخاب کنید.

3– نام Controller را AuthorsController انتخاب کنید.

4– دکمه (+) را برای Data Context Class بزنید.

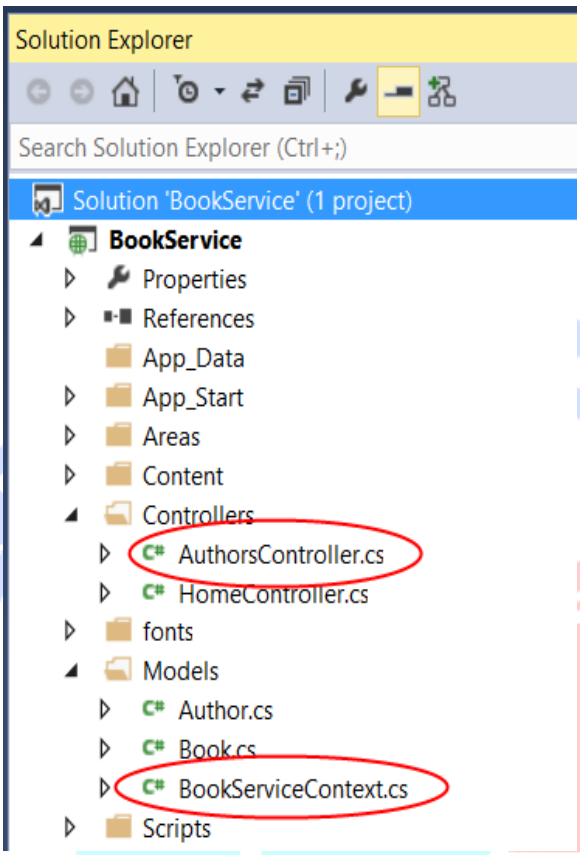


در پنجره New Data Context ، نام پیش فرض را قرار داده و روی Add کلیک کنید.



برای تکمیل فرآیند پنجره Add Controller، Add را کلیک کنید. این پنجره دو کلاس به پروژه شما اضافه می کند:

- AuthorsController یک Web API Controller تعریف می کند. REST API Controller لیست author برای اجرای عملیات CRUD کاربران پیاده سازی می کند.
- BookServiceContext موجودیت object ها را در زمان اجرا مدیریت می کند که شامل پر کردن دیتابیس با object ها، تغییر ردیابی و تداوم داده در دیتابیس می شود. این از DbContex ارث بری می کند.



در این قسمت، پروژه را دوباره build کنید.حالا دوباره این مراحل را برای اضافه کردن Controller API برای موجودیت های Book طی کنید. در اینجا، Book را به عنوان کلاس model انتخاب کنید و کلاس `BookServiceContext` موجود را برای data context انتخاب کنید(data context جدید نسازید) برای اضافه کردن Controller ، Add کنید.