

در پست قبلی با تکنولوژی MEF آشنا شدید. در این پست قصد داریم روش استفاده از MEF رو در Asp.Net MVC نمایش بدم. برای شروع یک پروژه پروژه MVC ایجاد کنید. در قسمت Model کلاس Book رو ایجاد کنید و کدهای زیر رو در اون قرار بدید.

```
public class Book
{
    public int Id { get; set; }
    public string Title { get; set; }
    public string ISBN { get; set; }
}
```

یک فولدر به نام Repositories ایجاد کنید و یک اینترفیس به نام IBookRepository رو به صورت زیر ایجاد کنید.

```
public interface IBookRepository
{
    IList<Book> GetBooks();
}
```

حالا نوبت به کلاس BookRepository می‌رسه که باید به صورت زیر ایجاد بشه.

```
[Export( typeof( IBookRepository ) )]
public class BookRepository
{
    public IList<Book> GetBooks()
    {
        List<Book> listOfBooks = new List<Book>( 3 );
        listOfBooks.AddRange( new Book[]
        {
            new Book(){Id=1 , Title="Book1"},
            new Book(){Id=2 , Title="Book2"},
            new Book(){Id=3 , Title="Book3"},
        } );
        return listOfBooks;
    }
}
```

بر روی پوشه کنترلر کلیک راست کرده و یک کنترلر به نام BookController ایجاد کنید و کدهای زیر رو در اون کپی کنید.

```
[Export]
[PartCreationPolicy( CreationPolicy.NonShared )]
public class BookController : Controller
{
    [Import( typeof( IBookRepository ) )]
    BookRepository bookRepository;

    public BookController()
    {
    }

    public ActionResult Index()
    {
        return View( this.bookRepository.GetBooks() );
    }
}
```

PartCreationPolicy که شامل 3 نوع می‌باشد.

Shared: یعنی در نهایت فقط یک نمونه از این کلاس در هر Container وجود دارد.

NonShared: یعنی به ازای هر درخواستی که از نمونه‌ی Export شده می‌شود یک نمونه جدید ساخته می‌شود.

Any: هر 2 حالت فوق Support می‌شود.

حالا قصد داریم یک ControllerFactory با استفاده از MEF ایجاد کنیم. (Controller Factory برای ایجاد نمونه ای از کلاس Controller مورد نظر استفاده می‌شود) برای بیشتر پروژه‌ها استفاده از DefaultControllerFactory کاملاً مناسبه.

```
public class MEFControllerFactory : DefaultControllerFactory
{
    private readonly CompositionContainer _compositionContainer;

    public MEFControllerFactory( CompositionContainer compositionContainer )
    {
        _compositionContainer = compositionContainer;
    }

    protected override IController GetControllerInstance( RequestContext requestContext, Type controllerType )
    {
        var export = _compositionContainer.GetExports( controllerType, null, null ).SingleOrDefault();

        IController result;

        if ( export != null )
        {
            result = export.Value as IController;
        }
        else
        {
            result = base.GetControllerInstance( requestContext, controllerType );
            _compositionContainer.ComposeParts( result );
        }
    }
}
```

اگر با مفاهیمی نظیر CompositionContainer آشنایی ندارید می‌تونید [پست قبلی](#) رو مطالعه کنید.

حالا قصد داریم یک DependencyResolver رو با استفاده از MEF به صورت زیر ایجاد کنیم. (DependencyResolver برای ایجاد نمونه ای از کلاس مورد نظر برای کلاس هایی است که به یکدیگر نیاز دارند و برای ارتباط بین آن از Dependency Injection استفاده شده است).

```
public class MefDependencyResolver : IDependencyResolver
{
    private readonly CompositionContainer _container;

    public MefDependencyResolver( CompositionContainer container )
    {
        _container = container;
    }

    public IDependencyScope BeginScope()
    {
        return this;
    }

    public object GetService( Type serviceType )
    {
        var export = _container.GetExports( serviceType, null, null ).SingleOrDefault();

        return null != export ? export.Value : null;
    }

    public IEnumerable<object> GetServices( Type serviceType )
    {
        var exports = _container.GetExports( serviceType, null, null );
        var createdObjects = new List<object>();

        if ( exports.Any() )
        {
            // ...
        }
    }
}
```

```

        {
            foreach ( var export in exports )
            {
                createdObjects.Add( export.Value );
            }
        }
        return createdObjects;
    }
    public void Dispose()
    {
    }
}

```

حال یک کلاس Plugin ایجاد می‌کنیم.

```

public class Plugin
{
    public void Setup()
    {
        var container = new CompositionContainer( new DirectoryCatalog( HostingEnvironment.MapPath(
            "~/bin" ) ) );

        CompositionBatch batch = new CompositionBatch();
        batch.AddPart( this );

        ControllerBuilder.Current.SetControllerFactory( new MEFControllerFactory( container ) );

        System.Web.Http.GlobalConfiguration.Configuration.DependencyResolver = new
        MefDependencyResolver( container );

        container.Compose( batch );
    }
}

```

همانطور که در این کلاس می‌بینید ابتدا یک CompositionContainer ایجاد کردیم که یک ComposablePartCatalog از نوع DirectoryCatalog به اون پاس دادیم.

DirectoryCatalog یک مسیر رو دریافت کرده و Assembly‌های موجود در مسیر مورد نظر رو به عنوان Catalog در Container اضافه می‌کنه. می‌تونستید از یک AssemblyCatalog هم به صورت زیر استفاده کنید.

```
var container = new CompositionContainer( new AssemblyCatalog( Assembly.GetExecutingAssembly() ) );
```

در تکه کد زیر ControllerFactory پروژه رو از نوع MEFControllerFactory قرار دادیم.

```
ControllerBuilder.Current.SetControllerFactory( new MEFControllerFactory( container ) );
```

و در تکه کد زیر هم DependencyResolver پروژه از نوع MefDependencyResolver قرار دادیم.

```
System.Web.Http.GlobalConfiguration.Configuration.DependencyResolver = new MefDependencyResolver(
    container );
```

کافیست در فایل Global نیز تغییرات زیر را اعمال کنیم.

```

protected void Application_Start()
{
    Plugin myPlugin = new Plugin();
    myPlugin.Setup();

    AreaRegistration.RegisterAllAreas();
}

```

```
        RegisterRoutes( RouteTable.Routes );
    }

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

        routes.MapRoute(
            "Default", // Route name
            "{controller}/{action}/{id}", // URL with parameters
            defaults: new { controller = "Book", action = "Index", id = UrlParameter.Optional } // Parameter
        );
    }
}
```

در انتها View متناظر با BookController رو با سلیقه خودتون ایجاد کنید و بعد پروژه رو اجرا و نتیجه رو مشاهده کنید.

نظرات خوانندگان

نویسنده: حسینی
تاریخ: ۱۳۹۱/۱۲/۱۷ ۱:۹

سلام
تشکر از مطلب خوبتون
اگر بخوایم الگوی Unitof Work با MEF پیاده سازی کنیم دراون صورت به چه صورته؟

نویسنده: مسعود م. پاکدل
تاریخ: ۱۳۹۱/۱۲/۱۹ ۸:۲۶

با توجه به این که پیاده سازی الگوی UnitOfWork در ORM های مختلف متفاوت است یک مثال کلی در این زمینه پیاه سازی می کنم.
فرض کنیم بک اینترفیس به صورت زیر داریم:

```
public interface IUnitOfWork
{
    ISession CurrentSession { get; }
    void BeginTransaction();
    void Commit();
    void RollBack();
}
```

می تونید به جای استفاده از ISession از DbSet در EF CodeFirst هم استفاده کنید.
حالا نیاز به کلاس UnitOfWork برای پیاده سازی Interface بالا داریم. به صورت زیر:

```
[Export(typeof(IUnitOfWork))]
public class UnitOfWork : IUnitOfWork
{
    public ISession CurrentSession
    {
        get { throw new NotImplementedException(); }
    }

    public void BeginTransaction()
    {
        throw new NotImplementedException();
    }

    public void Commit()
    {
        throw new NotImplementedException();
    }

    public void RollBack()
    {
        throw new NotImplementedException();
    }
}
```

پیاده سازی متدها رو به عهده خودتون. فقط از Export Attribute برای تعیین نوع وابستگی کلاس UnitOfWork استفاده کردم.
و در آخر کلاس Repository مربوطه هم به شکل زیر است.

```
public class Respository
{
    [Import]
    private IUnitOfWork uow;

    public Respository()
    {
    }
}
```

در کلاس Repository فیلد uow به دلیل داشتن Import Attribute همیشه توسط Composition Container

مقدار دهی می‌شه.

نویسنده: سینا نادى حق
تاریخ: ۱۵:۳۱۳۹۲/۰۹/۲۰

سلام؛ من MEF رو تو MVC اعمال کردم الان به یک مشکلی بر خوردم. ممنون میشم اگه کمک کنید.
من Pluginها رو توی پروژه جدا با Class Library می‌نویسم، پروژه MVC هم razor engine هست. ولی وقتی می‌خواهم viewها رو به Modelها نسبت بدم همیشه (Model توی همون پروژه Class Library پلاگین هست)
کد view:

```
@model Plugin1.Models.Post
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
@Model.Title
```

ارورشم اینه:

The type or namespace name 'Plugin1' could not be found (are you missing a using directive or an assembly reference?)

در کل چطوری می‌تونم Modelها رو به Viewها در این حالت نسبت بدم.
ممنون

نویسنده: مسعود پاکدل
تاریخ: ۱۷:۱۳۱۳۹۲/۰۹/۲۰

در ابتدا باید عنوان کنم که از آن جا مدل باید بین پلاگین‌ها و سایر ماژول‌ها به اشتراک گذاشته شود در نتیجه مدل برنامه نباید در Class Library پلاگین‌ها باشد. از طرفی شما نیازی به Export کردن مدل‌های برنامه توسط MEF ندارید. دلیل خطای برنامه شما این است که Class Library پلاگین شما به پروژه MVC رفرنس داده نشده است (که البته درست هم عمل نموده اید، فقط مدل برنامه را در یک پروژه دیگر برای مثال DomainModel قرار داده و سپس پروژه DomainModel را به صورت مستقیم به پروژه MVC رفرنس دهید).

فقط به این نکته نیز توجه داشته باشید که بعد از تعریف پلاگین‌ها در class library دیگر حتما اسمبلی‌های مربوطه را از طریق کاتالوگ‌های موجود (نظیر AssemblyCatalog و DirectoryCatalog و ...) به CompositionContainer پروژه MVC خود اضافه کنید.

نویسنده: سینوس
تاریخ: ۲۳:۴۰۱۳۹۲/۱۱/۰۵

با سلام و خسته نباشید. من پروژم رو با MEF راه اندازی کردم مشکل خاصی هم ندارم

با توجه به فرمایش شما باید Modelها توی یک پروژه جدا و عمومی استفاده بشه. مشکلی که من دارم اینه که وقتی پروژه (core) آماده شد و آپلود کردم دیگه نمی‌خواهم وقتی کاربری از یک ماژول استفاده می‌کند خود پروژه یا پروژه مدل‌ها update بشه. دلیلشم این هست که هر کاربری ماژول‌های جداگانه دارد و ماژول‌ها رو runtime اضافه، حذف و یا بروز می‌کند.
ممنون از لطفتون

یک Model چیست؟

• قسمتی از Application است که Domain Logic را پیاده سازی می‌کند.

• همچنین با عنوان Business Logic نیز شناخته می‌شود.

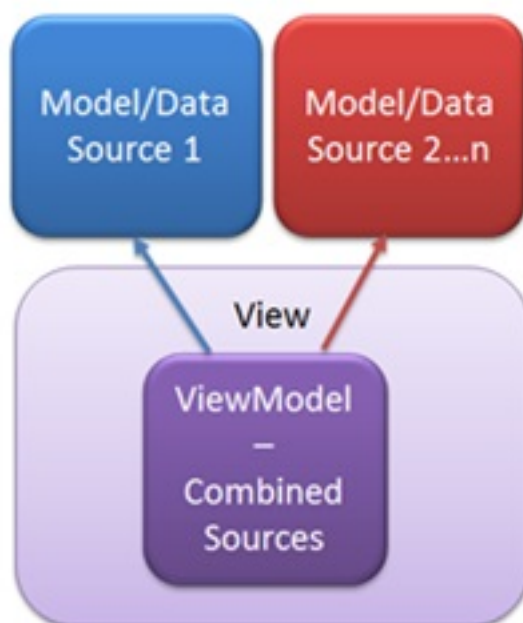
• Domain Logic داده‌هایی را که بین UI و دیتابیس پاس داده می‌شود، مدیریت می‌کند.

• برای مثال، در یک سیستم انبار، Model کارش ذخیره سازی اقلام در حافظه و Logic تعیین کننده موجود بودن یک آیتم در انبار می‌باشد.

یک ViewModel چیست؟

• ViewModel به ما این امکان را می‌دهد تا از چندین Entity، یک شیء واحد بسازیم.

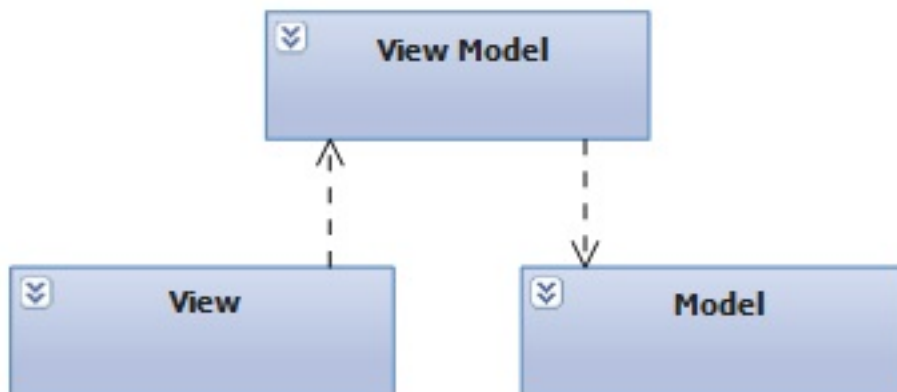
• بهینه شده برای استفاده و رندر توسط یک View



چرا باید از ViewModel استفاده کنیم؟

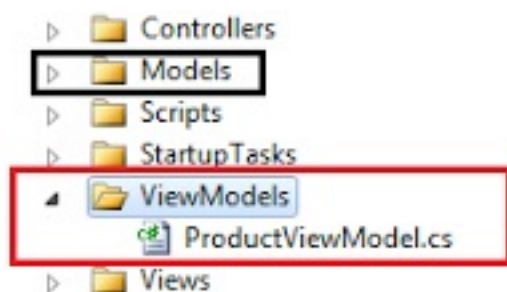
- اگر شما بخواهید چندین Model را به یک View پاس دهید، استفاده از ViewModel ایده بسیار خوبی است.
- همچنین امکان Validate کردن ViewModel را به روشی متفاوت‌تر از Domain Model برای سناریوهای اعتبارسنجی attribute-based را در اختیارمان قرار می‌دهد.
- برای قالببندی داده‌ها نیز استفاده می‌شود.

- برای مثال به یک داده یا مقدار پولی قالببندی شده به یک روش خاص نیاز داریم؟
- ViewModel بهترین مکان برای انجام این کار است.
- استفاده از ViewModel تعامل بین Model و View را ساده‌تر می‌کند.



مکان قرارگیری ViewModel در پروژه بهتر است کجا باشد؟

می‌تواند در یک پوشه با نام ViewModel در ریشه پروژه باشد.



به صورت یک dll ارجاع داده شده به پروژه.
در یک پروژه جدا به عنوان یک Service Layer.

Best Practice هایی در زمان استفاده از ViewModelها

داده‌هایی را که قرار است در View رندر شوند، داخل ViewModel قرار دهید.
View باید خصوصیتی را که نیاز دارد، مستقیماً از ViewModel بدست بیاورد.
وقتی که ViewModel شما پیچیده‌تر می‌شود، بهتر است از یک Mapper استفاده کنید.

اجازه دهید مثالی را در این رابطه براتون بیارم :

در یک View می‌خواهیم هم لیست اخبار سایت و هم لیست سخنرانان سایت (مثال مربوط به یک پروژه برای همایش است) را نمایش دهیم. خوب طبق مطالب فوق استفاده از ViewModel بهترین راه حل است. ViewModel مربوطه به صورت زیر تعریف شده

است :

```
public class SpeakerAndNewsViewModel
{
    public IEnumerable<News> News { get; set; }
    public IEnumerable<Speaker> Speakers { get; set; }
}
```

و در اکشن متد Index هم بدین صورت Model ها را به ViewModel ساخته شده پاس می‌دهیم :

```
public ActionResult Index()
{
    var news = db.News.OrderBy(r=>r.Date)
        .Take(3);
    var speakers = db.Speakers.Take(4);
    var model = new SpeakerAndNewsViewModel {
        News=news,
        Speakers=speakers
    };
    return View(model);
}
```

و یک View را به صورت Strongly Typed از نوع ViewModel خود که در اینجا SpeakerAndNewsViewModel است ایجاد می‌کنیم :

و View مربوطه را به شکل زیر تغییر می‌دهیم :

```
@model Project.Models.SpeakerAndNewsViewModel
@{
    ViewBag.Title = "Home Page";
}
@section news
{
    @foreach (var item in Model.News)
    {
        //..
    }
}
@section speakers
{
    @foreach (var item in Model.Speakers)
    {
        //...
    }
}
```

نظرات خوانندگان

نویسنده: شاهین کیاست
تاریخ: ۱۴:۴۶ ۱۳۹۱/۱۲/۱۳

• ViewModel به ما این امکان را میدهد تا از چندین Entity یک شیء واحد بسازیم.

من فکر می‌کنم ViewModel، مدل مورد نیاز برای یک View را فراهم می‌کند و لزوماً برای این نیست از چند موجودیت دیتابیس یک شیء بسازیم و به View پاس دهیم.

البته در ادامه‌ی مطلب به این مورد اشاره شده. اما به غلط در برخی آموزش‌ها ViewModel اینگونه ایجاد می‌شود که شامل چند Entity هست.

نویسنده: محمد آزاد
تاریخ: ۱۰:۵ ۱۳۹۱/۱۲/۱۴

• قسمتی از Application است که Domain Logic را پیاده سازی می‌کند.

• همچنین با عنوان Business Logic نیز شناخته می‌شود.

برای مثال، در یک سیستم انبار، Model کارش ذخیره سازی اقلام در حافظه و Logic تعیین کننده موجود بودن یک آیتم در انبار میباشد.

مگه مدل Entity‌های ما نیستند که توی سیستم استفاده میشن؟ یا دید من اشتباه بوده نسبت به مدل.
مثلا ما یه مدل به نام Person داشته باشیم من اونو به این صورت می‌بینم و هیچ منطقی توش پیاده سازی نمی‌کنم

```
Public Class PersonModel
{
    long Id{get;set;}
    string Name{get;set;}
}
```

اما با حرف شما مدل Person به این صورت هستش

```
Public Class PersonModel
{
    long Id{get;set;}
    string Name{get;set;}

    int SavePerson(person p);
    Person Finde(long personId);
}
```

من حرف شما رو بد برداشت کردم یا اینکه دیدم نسبت به مدل اشتباه بوده؟

نویسنده: سیروان عقیفی
تاریخ: ۱۰:۴۲ ۱۳۹۱/۱۲/۱۴

بنبید Model به گفته آقای Scott Allen ، در توسعه نرم افزار میتونه صدها مفهوم رو پوشش بده، در کل Model رو میتونیم به این صورت تعریف کنیم : مجموعه ایی از کلاسها که تعریف کننده داده ها و همچنین Business Rule ها هستند Business Rule هم همون قواعد یا Business Logic مربوطه هستند که در بالا تعریف شده که مشخص میکنه دیتا چگونه ذخیره بشه و یا تغییر پیدا کنه.

نویسنده: محمد آزاد
تاریخ: ۱۱:۳۱ ۱۳۹۱/۱۲/۱۴

برای جدا سازی کد بهتر نیست که مثلا Business نرم افزار تو به لایه جدا پیاده سازی بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۱۶ ۱۳۹۱/۱۲/۱۴

البته عنوان کردند که در این زمینه بین علما اختلاف هست.
برای مثال الگویی وجود دارد به نام [Active record](#) که دقیقا با تعریف عنوان شده مطابقت دارد (یک شیء می تواند متدهای مثلا افزودن و به روز رسانی و غیره را هم داشته باشد). مزایا و معایب خاص خودش را هم دارد. مطالعه در مورد این ها جهت آشنایی با سلايق و دیدگاه های مختلف طراحی خوب است. اما وجود آن ها به معنای الزام استفاده از روش های مطرح شده نیست.

نویسنده: امیر بختیاری
تاریخ: ۱۷:۳۲ ۱۳۹۱/۱۲/۱۴

با تشکر از مطلب خوبتان

فقط یه سوال اینکه View Model فقط در جاهایی مثل این که مثال زدید کاربرد دارد یا در جایی که الان در برنامه های وب انجام میدیم و چند تا جدول که با هم ارتباط دارند و یک view از روی آنها ایجاد می کنیم و در پروژه از این View ساخته شده استفاده می کنیم برای اعمال مختلف نیز می توان استفاده کرد . برای مثال دانشجو با نمرات و درس ها و رشته که با هم در ارتباط دارند را یک View Model کرد و در هر قسمت پروژه که نیاز داشتیم از این استفاده کنیم؟
یه سوال دیگه هم اینکه خیلی جاها دیدم که نوشته از Model به صورت مستقیم استفاده نکنید و یک View Model این وسط برای ارتباط داشته باشیم می خواستم ببینم آیا درسته و دلیلش چیه ؟

نویسنده: شاهین کیاست
تاریخ: ۱۷:۴۰ ۱۳۹۱/۱۲/۱۴

سلام ؛

برای درک بهتر ViewModel و علت استفاده از آن [این مطلب](#) را مطالعه بفرمایید .

نویسنده: یزدان
تاریخ: ۲:۱۶ ۱۳۹۲/۰۱/۱۲

لطف میکنید نحوه ارتباط viewmodel و view زمانی که بین model و view قرار میگیره توضیح بدین .
اینکه در آخر ViewModel رو چگونه به Model مرتبط میکنیم ؟
آیا امکان این که هم Model و هم ViewModel اعتبار سنجی کنیم کار درسته ؟
بیشتر قسمت ربط ViewModel به Model برام اهمیت داره که متوجهش نمیشم .
با تشکر

نویسنده: علي
تاریخ: ۱۰:۳۴ ۱۳۹۲/۰۱/۱۲

برای اتصال ViewModel به مدل می نویسیم: یک خاصیت viewmodel، خودت رو به یک خاصیت model انتساب بده پدرجان!
همین! (البته حالت پیشرفته تر استفاده از AutoMapper هست که نخوای این ها رو اگه سخت هست دستی بنویسی)

خواص مدل رابطه یک به یک دارد با جدول شما در بانک اطلاعاتی. اون هم اطلاعاتش نیاز به اعتبار سنجی دارد. ViewModel هم نیاز دارد چون یک سری خاصیت اضافی عموماً دارد که در بانک اطلاعاتی نیست و فقط در View تعریف شده.

نویسنده: یزدان
تاریخ: ۱۱:۴۰ ۱۳۹۲/۰۱/۱۲

«یک خاصیت viewModel، خودت رو به یک خاصیت model انتساب بده پدرجان! همین!»
این رو با مثال و یه مدل و viewModel ساده نشون بده. اینو که شما گفتی میدونم، شما زحمت بکش نحوه انتساب رو بگو. با کد

نویسنده: علی
تاریخ: ۱۲:۲۳ ۱۳۹۲/۰۱/۱۲

انتساب از طریق عملگر مساوی انجام میشه. مثلاً `user.Name = userViewModel.Name`

نویسنده: یزدان
تاریخ: ۱۳:۸ ۱۳۹۲/۰۱/۱۲

این انتساب کجا انجام میشه؟ توی اکشن؟
اگه مقدوره یک مثال ساده با ViewModel و Model و مکانی که این انتساب صورت میگیره بیان کنید.

نویسنده: علی
تاریخ: ۱۳:۱۳ ۱۳۹۲/۰۱/۱۲

در همین مقاله بالا از قسمت «اجازه دهید مثالی را در این رابطه براتون بیارم:» به بعد رو مطالعه کنید. این مثال در مورد HttpGet هست. در حالت HttpPost هم همینیه. فرقی نمی‌کنه.

نویسنده: علیرضا پایدار
تاریخ: ۱۳:۴۱ ۱۳۹۲/۰۱/۱۲

شما اگر که از AutoMapper استفاده می‌کنی باید ابتدا یک Map ایجاد کنی و بهتر است که این کار در متد `Application_Start` انجام بگیرد. به طور مثال:

```
protected void Application_Start()
{
    //مپ کردن مدل به ویومدل
    Mapper.CreateMap<DomainClasses.Product, ProductListVM>();
    //مپ کردن ویومدل به مدل
    Mapper.CreateMap<ProductCreateVM, DomainClasses.Product>();
}
```

حال فرض بگیر دیتایی داری که می‌خواهی انتصاب بدی به ViewModel به این شکل عمل میکنی:

```
public ActionResult ShowList()
{
    var productList = _context.Products.ToList();
    //انتصاب
    var viewModel = Mapper.Map<List<domain.Products>, List<ProductListVM>>(productList);
    return View(viewModel);
}
```

حال برعکس می‌خواهی ViewModel انتصاب بدی به Model:

```
[HttpPost]
public virtual ActionResult Create(ProductCreateVM product)
```

```

{
    try
    {
        if (ModelState.IsValid)
        {
            var model = Mapper.Map<ProductCreateVM, domain.Product>(product);
            _cotext.Products.Add(model);
            _cotext.SaveChanges();
            return RedirectToAction(ShowList);
        }
        return View(product);
    }
    catch
    {
    }
}

```

امیدوارم به درد بخوره.قبل از هر چیز مطالب مربوط به AutoMapper سایت را مطالعه کن مثل:

[مقدمه ای بر AutoMapper](#)

[نکته‌ای در استفاده از AutoMapper](#)

و ...

موفق باشی

اگرم خواستی میتونی دستی این انتصاب‌ها را انجام بدی البته من پیشنهاد نمیکنم.

نویسنده: یزدان

تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۴:۵۰

تشکر مهندس پایدار بابت جواب و توضیح کاملی که ارائه دادین

نویسنده: مولانا

تاریخ: ۱۳۹۲/۰۱/۳۰ ۱۶:۴۲

با سلام.

فرض کنید یک مدل با عنوان category داریم.

اگر نیاز به نمایش تعداد محصولات آن category در ویومدل داشته باشیم، به نظر شما بهترین راه کدام است؟

آیا باید یک کوئری جدا برای یافت تعداد محصولات بزنی و به خاصیت موجود در ویومدل پاس دهیم.

یا باید این خاصیت را درون مدل در نظر بگیریم؟

با تشکر.

نویسنده: وحید نصیری

تاریخ: ۱۳۹۲/۰۱/۳۰ ۲۰:۴۲

- هر دو مورد رو می‌تونید انجام بدید.

- برای نمونه در این سایت و در همین صفحه‌ای که ملاحظه می‌کنید، «تعداد نظرات» فیلد محاسباتی است. یا در جایی دیگر، تعداد مطالب یک برچسب هم فیلد محاسباتی است. این کار برای بالا بردن سرعت نمایشی سایت انجام شده.

نویسنده: ضیا

تاریخ: ۱۳۹۲/۰۲/۲۲ ۰:۱۸

سلام ...

من تاجاییکه میتونستم چیزهایی که درباره viemodel و automapper توی سایت بود رو خوندم ولی چیزی که میخواستم را پیدا

نکردم شاید هم درک درستی ازش نداشتم در کل سوال سوال من از شما دوستان اینه :
یک viewmodel دارم که از چندین جدول توی db ایجاد شده که در زیر نوشتمش :

```
public class ResumeViewModel
{
    public ResumeViewModel()
    {
    }
    public ResumeViewModel(IEnumerable<Resume> resum, IEnumerable<Work_Experience_Job_Seeker>
workExperienceJobSeekerOfViewModel, IEnumerable<Job_Expertises> jobExpertisesOfViewModel,
IEnumerable<Degrees_Work_Experience_Required> degreesWorkExperienceRequiredOfViewModel,
IEnumerable<Specialized_Course> specializedCourseOfViewModel, IEnumerable<Book_Published>
bookPublishedOfViewModel, IEnumerable<Basic_Table> basicTable)
    {
        ResumeOfViewModel = resum;
        WorkExperienceJobSeekerOfViewModel = workExperienceJobSeekerOfViewModel;
        JobExpertisesOfViewModel = jobExpertisesOfViewModel;
        DegreesWorkExperienceRequiredOfViewModel = degreesWorkExperienceRequiredOfViewModel;
        SpecializedCourseOfViewModel = specializedCourseOfViewModel;
        BookPublishedOfViewModel = bookPublishedOfViewModel;
        BasicTable = basicTable;
    }
    public IEnumerable<Resume> ResumeOfViewModel { get; set; }
    public IEnumerable<Work_Experience_Job_Seeker> WorkExperienceJobSeekerOfViewModel { get; set; }
    public IEnumerable<Job_Expertises> JobExpertisesOfViewModel { get; set; }
    public IEnumerable<Degrees_Work_Experience_Required> DegreesWorkExperienceRequiredOfViewModel {
get; set; }
    public IEnumerable<Specialized_Course> SpecializedCourseOfViewModel { get; set; }
    public IEnumerable<Book_Published> BookPublishedOfViewModel { get; set; }
    public IEnumerable<Basic_Table> BasicTable { get; set; }

    public int NumberForm { get; set; } // EditResumes.ctlml & ShowResumes.ctlml ==> baraye
select kardan formha :D
}
```

و این viewmodel رو توی متد Edit Resume استفاده کردم که متد get بصورت زیر ::

```
[HttpGet]
public ActionResult EditResumes(int id)
{
    var contex = new Final_My_ProjectEntities2();
    var res1 = contex.Resumes.Where(rec => rec.Resume_ID == id);
    var res2 = contex.Work_Experience_Job_Seeker.Where(rec => rec.Resume_ID == id).ToList();
    var res3 = contex.Job_Expertises.Where(rec => rec.Resume_ID == id).ToList();
    var res4 = contex.Degrees_Work_Experience_Required.Where(rec => rec.Resume_ID ==
id).ToList();
    var res5 = contex.Specialized_Course.Where(rec => rec.Resume_ID == id).ToList();
    var res6 = contex.Book_Published.Where(rec => rec.Resume_ID == id).ToList();
    var res12 = contex.Basic_Table.ToList();
    var viewModel = new ResumeViewModel(res1, res2, res3, res4, res5, res6, res12);
    var items = new SelectList(
        new[]
        {
            new {Value = "1", Text = "فرم مهارت ها"},
            new {Value = "2", Text = "فرم کتاب/مقالات منتشر شده"},
            new {Value = "3", Text = "فرم سابقه کاری"},
            new {Value = "4", Text = "فرم دوره های تخصصی گذرانده"},
            new {Value = "5", Text = "فرم تخصص های شغلی"},
            new {Value = "6", Text = "فرم مدارک تحصیلی"}
        },
        "Value", "Text");

    ViewBag.Form = new SelectList(items, "Value", "Text");

    var res7 = contex.Basic_Table.Where(rec => rec.Domain == "MilitaryStatus").ToList();
    ViewBag.MilitaryStatus = new SelectList(res7, "Value", "Meaning", res1);

    var res8 = contex.Basic_Table.Where(rec => rec.Domain == "Sex");
    ViewBag.Sex = new SelectList(res8, "Value", "Meaning", res1);

    var res9 = contex.Basic_Table.Where(rec => rec.Domain == "MartialStatus").ToList();
    ViewBag.MartialStatus = new SelectList(res9, "Value", "Meaning", res1);

    var res10 = contex.Basic_Table.Where(rec => rec.Domain == "Degree").ToList();
    ViewBag.Degree = new SelectList(res10, "Value", "Meaning");
}
```

```

var res11 = contex.Basic_Table.Where(rec => rec.Domain == "Ability").ToList();
ViewBag.Ability = new SelectList(res11, "Value", "Meaning");

return View(viewModel);
}

```

و view این متد بصورت زیر هست البته قسمتی از آن ::

```

@model Final_My_Project.ViewModels.ResumeViewModel

@{
    ViewBag.Title = "ویرایش رزومه";
    ViewBag.PartOne = "فرم مهارت ها";
    ViewBag.PartTwo = "فرم کتاب/مقالات منتشر شده";
    ViewBag.PartThree = "فرم سابقه کاری";
    ViewBag.PartFour = "فرم دوره های تخصصی گذرانده";
    ViewBag.PartFive = "فرم تخصص های شغلی";
    ViewBag.PartSix = "فرم مدارک تحصیلی";
}
<h2 style="font-family: Arial;">@ViewBag.Title</h2><br/>
<script type="text/javascript">
    $(function () {
        $('#Gender').change(function () {
            var selectKind = $(this).find('option:selected').text();
            var divMilitary;
            if (selectKind == "زن") {
                divMilitary = $('#Military');
                divMilitary.hide();
                divMilitary.css('display', 'none');
            }
            else if (selectKind == "مرد") {
                divMilitary = $('#Military');
                divMilitary.show();
                divMilitary.css('display', 'block');
            }
        });
    });
</script>

<script type="text/javascript">
    $(function () {
        $('#SelectForm').change(function () {
            var selectFrom = $(this).find('option:selected').text();

            if (selectFrom == "فرم مهارت ها") {
                $('#PartOne').show();
                $('#PartOne').css('display', 'block');
                $('#PartTwo').hide();
                $('#PartTwo').css('display', 'none');
                $('#PartThree').hide();
                $('#PartThree').css('display', 'none');
                $('#PartFour').hide();
                $('#PartFour').css('display', 'none');
                $('#PartFive').hide();
                $('#PartFive').css('display', 'none');
                $('#PartSix').hide();
                $('#PartSix').css('display', 'none');
            }
            if (selectFrom == "فرم کتاب/مقالات منتشر شده") {
                $('#PartTwo').show();
                $('#PartTwo').css('display', 'block');
                $('#PartOne').show();
                $('#PartOne').css('display', 'none');
                $('#PartThree').hide();
                $('#PartThree').css('display', 'none');
                $('#PartFour').hide();
                $('#PartFour').css('display', 'none');
                $('#PartFive').hide();
                $('#PartFive').css('display', 'none');
                $('#PartSix').hide();
                $('#PartSix').css('display', 'none');
            }
        });
    });
</script>

```



```

    }
    if (selectFrom == "فرم سابقه کاری") {
        $('#Part3').show();
        $('#Part3').css('display', 'block');
        $('#PartTwo').hide();
        $('#PartTwo').css('display', 'none');
        $('#PartOne').show();
        $('#PartOne').css('display', 'none');
        $('#Part4').hide();
        $('#Part4').css('display', 'none');
        $('#Part5').hide();
        $('#Part5').css('display', 'none');
        $('#Part6').hide();
        $('#Part6').css('display', 'none');
    }
    if (selectFrom == "فرم دوره‌های تخصصی گذرانده") {
        $('#Part4').show();
        $('#Part4').css('display', 'block');
        $('#PartTwo').hide();
        $('#PartTwo').css('display', 'none');
        $('#PartOne').show();
        $('#PartOne').css('display', 'none');
        $('#Part3').hide();
        $('#Part3').css('display', 'none');
        $('#Part5').hide();
        $('#Part5').css('display', 'none');
        $('#Part6').hide();
        $('#Part6').css('display', 'none');
    }
    if (selectFrom == "فرم تخصص‌های شغلی") {
        $('#Part5').show();
        $('#Part5').css('display', 'block');
        $('#PartTwo').hide();
        $('#PartTwo').css('display', 'none');
        $('#PartOne').show();
        $('#PartOne').css('display', 'none');
        $('#Part3').hide();
        $('#Part3').css('display', 'none');
        $('#Part4').hide();
        $('#Part4').css('display', 'none');
        $('#Part6').hide();
        $('#Part6').css('display', 'none');
    }
    if (selectFrom == "فرم مدارک تحصیلی") {
        $('#Part6').show();
        $('#Part6').css('display', 'block');
        $('#Part5').show();
        $('#Part5').css('display', 'none');
        $('#PartTwo').hide();
        $('#PartTwo').css('display', 'none');
        $('#PartOne').show();
        $('#PartOne').css('display', 'none');
        $('#Part3').hide();
        $('#Part3').css('display', 'none');
        $('#Part4').hide();
        $('#Part4').css('display', 'none');
    }
    });
});
</script>

@Html.DropDownListFor(m=>m.NumberForm, (SelectList)ViewBag.Form, new { id = "SelectForm" })

@using (Html.BeginForm())
{
    @Html.ValidationSummary(true)

    <div id="PartOne" >
    <h3 style="font-family: Arial; color: #008080; font-weight: bold; ">@ViewBag.PartOne</h3><br/>

```

```

@foreach (var item in Model.ResumeOfViewModel)
{
    <table dir="rtl">
    <tr>
        <td>
            @Html.Label("عنوان رزومه")
        </td>
        <td>
            @Html.TextBoxFor(model => item.Title_Of_Resume , new {@class = "text", style =
"width:100 px"})
            @Html.ValidationMessageFor(model => item.Title_Of_Resume)
        </td>
    </tr>
    <tr>
        <td>
            <div id="Gender" >
                @Html.Label("نوع جنسیت")
                @Html.DropDownList("نوع جنسیت", new SelectList(ViewBag.Sex,
                    "Value", "Text", item.Sex_ID == 0 ? 0 : item.Sex_ID))
                @Html.ValidationMessageFor(model => item.Sex_ID)
            </div>
        </td>
        <td>
            <div >
                @Html.Label("وضعیت تاهل")
                @Html.DropDownList("وضعیت تاهل", new SelectList(ViewBag.MaritalStatus,
                    "Value", "Text", item.Marital_Status_ID == 0 ? 0 : item.Marital_Status_ID))
                @Html.ValidationMessageFor(model => item.Marital_Status_ID)
            </div>
        </td>
    </tr>
    <tr id="Military" style="display: none;">
        <td>
            @Html.Label("وضعیت نظام وظیفه")
        </td>
        <td>
            @Html.DropDownList("وضعیت نظام وظیفه", new SelectList(ViewBag.MilitaryStatus,
                "Value", "Text", item.Military_Status_ID == 0 ? 0 :
item.Military_Status_ID), new { id = "Gender" })
            @Html.ValidationMessageFor(model => item.Military_Status_ID)
        </td>
    </tr>
    <tr>
        <td>
            @Html.Label("آشنایی با رایانه")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Knowledge_Of_Computers_ID,
(SelectList)ViewBag.Ability)
            @Html.ValidationMessageFor(model => item.Knowledge_Of_Computers_ID)
        </td>
    </tr>
    <tr>
        <td>
            @Html.Label("آشنایی با امور اداری و دفتری")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Knowledge_Administrative_and_Clerical_ID,
(SelectList) ViewBag.Ability)
            @Html.ValidationMessageFor(model => item.Knowledge_Administrative_and_Clerical_ID)
        </td>
    </tr>
    <tr>
        <td>
            @Html.Label("آشنایی با زبان انگلیسی")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Knowledge_Of_English_ID, (SelectList)
ViewBag.Ability)
            @Html.ValidationMessageFor(model => item.Knowledge_Of_English_ID)
        </td>
    </tr>
    <tr>

```

```

        <td>
            @Html.Label("آشنایی با زبان عربی")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Knowledge_Of_Arabic_ID, (SelectList)
ViewBag.Ability)
            @Html.ValidationMessageFor(model => item.Knowledge_Of_Arabic_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("آشنایی با ماکروسافت آفیس")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Knowledge_Of_Office_ID, (SelectList)
ViewBag.Ability)
            @Html.ValidationMessageFor(model => item.Knowledge_Of_Office_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("آشنایی با امور مالی و حسابداری")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Knowledge_Of_Finance_ID, (SelectList)
ViewBag.Ability)
            @Html.ValidationMessageFor(model => item.Knowledge_Of_Finance_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("آشنایی با مدیریت")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Knowledge_Of_Manage_ID, (SelectList)
ViewBag.Ability)
            @Html.ValidationMessageFor(model => item.Knowledge_Of_Manage_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("گواهینامه رانندگی پایه یک")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Driving_license_One_ID, (SelectList)
ViewBag.Ability)
            @Html.ValidationMessageFor(model => item.Driving_license_One_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("گواهینامه رانندگی پایه دو")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Driving_license_Two_ID, (SelectList)
ViewBag.Ability)
            @Html.ValidationMessageFor(model => item.Driving_license_Two_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("گواهینامه رانندگی پایه موتورسیکلت")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Certificate_Motor_ID, (SelectList)
ViewBag.Ability)
            @Html.ValidationMessageFor(model => item.Certificate_Motor_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("ماشین شخصی")
        </td>

```

```

        <td>
            @Html.DropDownListFor(model =>item.Personal_Vehicle_ID, (SelectList)
ViewBag.Ability)
            @Html.ValidationMessageFor(model =>item.Personal_Vehicle_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("روابط عمومی")
        </td>
        <td>
            @Html.DropDownListFor(model =>item.Public_Relationship_ID, (SelectList)
ViewBag.Ability)
            @Html.ValidationMessageFor(model =>item.Public_Relationship_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("دیگر توانایی ها")
        </td>
        <td>
            @Html.EditorFor(model =>item.Etc_Ability)
            @Html.ValidationMessageFor(model =>item.Etc_Ability)
        </td>
    </tr>
</table>
}
</div>

<p>
    <input type="submit" value="Save" onclick="return confirm('آیا از ثبت اطلاعات مطمئن هستید؟')>
</p>
}

<div>
    @Html.ActionLink("بازگشت به مدیریت رزومه ها", "ManageOfResumes")
</div>

@section scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

و مشکل اینجاست که بعد از ثبت اطلاعات وقتی به متد post میره مقدارش null هستش ... درحالیکه فقط در صورت edit اینجوریه وقتی از همین viewmodel برای مشاهده رزومه که فقط گزارشگیری استفاده میکنم نتیجه را میبینم ولی اینجا نه !
متد پست بصورت زیر هستش ... اگر میدونید چطور و چی کار کنم که این درست شه ممنون میشم ... چون دیگه نمیدونم تو متد پست چی بنویسم ... منتظر جوابم ... که چرا null میده و اینکه تو متد پست چطور اینارو ذخیره کنم توی db ... ؟

```

[HttpPost]
public ActionResult EditResumes(ResumeViewModel model) // model null mishe ! CHERAA??!
{
    var contex = new Final_My_ProjectEntities2();
    try
    {
        if (ModelState.IsValid)
        {
            // Code... che cody?
            contex.SaveChanges();
        }
    }
    catch (Exception)
    {
        ViewBag.wrong = "لطفا داده های ورودی را بررسی نمایید";
    }
    return View(model);
}

```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۲ ۰:۳۶

- شما هم از ViewModel استفاده کردی و هم از ViewBag با هم در یک View. از یکی بهتره استفاده کنید. ترجیح هم با ViewModel است تا به خواص آن به صورت strongly typed در یک View دسترسی داشته باشید و تحت کنترل کامپایلر هم باشد (تمام ViewBag ها رو حذف و تبدیل به خواص ViewModel کنید).

- کلاس ViewModel ایی که تدارک دیدید، دارای سازنده‌ای است که باید مقدار دهی شود پیش از استفاده. خوب ... MVC از کجا باید تشخیص بده که در زمان Post اطلاعات به سرور چطور باید این‌رو به صورت خودکار و هله سازی کنه؟ به عبارتی ViewModel شما باید بدون سازنده، قابل و هله سازی باشد. سازنده رو حذف و صرفاً از خواص استفاده کنید.

- برای ثبت نهایی هم باید اطلاعات خواص ViewModel رو (اون‌هایی رو که در Model یا جدول خاص شما وجود دارند) به Model [انتساب بدی](#). بعد این و هله Add شود به Context و دست آخر هم ذخیره.

(البته کار شما به نظر لایه بندی صحیحی نداره وگرنه جای این مسایل در لایه سرویس برنامه است؛ چه بازگشت ViewModel و چه ذخیره سازی اطلاعات آن)

نویسنده: ضیا
تاریخ: ۱۳۹۲/۰۲/۲۲ ۱:۲۸

من چندتایی سوال پیش اومد برام ::: چرا از نظر لایه بندی مورد داره ؟
جای کدوم مسائل ؟ viewModel ؟
یعنی الان ذخیره سازی اطلاعات کجا باشه ؟ مگه نباید تو قسمت کنترلر اینکار انجام شه ؟
من سازنده‌ها رو حذف کردم و viewModel رو صدا زدم ولی بعد از تغییرات وقتی به متد پست فرستاده میشه همچنان مقدارش null هست ... چرا ؟ :

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۲ ۸:۵۰

- در مورد اصول اولیه لایه بندی [یک دوره 11 قسمتی](#) در سایت هست. بحث مفصلی است و نیاز به پایه تئوری داره.

- مشکل بعدی کدهای شما به [Html.BeginForm](#) بر می‌گرده. زمانیکه پارامتری رو عنوان نکنید به اکشن متد پیش فرض مشخص شده در سیستم مسیریابی اشاره می‌کنه. بهتره که این موارد رو صریحاً ذکر کنید تا مشکل نداشته باشید؛ مانند کد زیر. به علاوه شما نمی‌تونید کل یک حلقه رو داخل یک فرم قرار بدید. الان در صفحه، چندین و چند کنترل دریافت اطلاعات دارای یک Id پیدا کردید. اعضای داخل حلقه باید یک فرم را تشکیل بدن؛ در این حالت شیء ارسالی به سرور مثلاً Resume خواهد بود و نه ResumeOfViewModel شما که در عمل یک لیست است. بهتر است داخل حلقه این View شلوغ رو Refactor کنید به یک Partial view با مدلی از جنس Resume تا بهتر بتونید مسایل رو مدیریت کنید.

```
@using (Html.BeginForm(actionName: "EditResume", controllerName: "SomeName"))
{
}
```

نویسنده: ضیا
تاریخ: ۱۳۹۲/۰۲/۲۲ ۱۳:۵۸

مرسی از جوابتون ...
اینجا یه حلقه هست با اینکه باید 6 تا حلقه باشه چون اطلاعات از 6 جدول جدا میاد برای ویرایش شدن با اینکه اینجا فقط مقادیر یه جدول نشون میده 5 تا جدول دیگه مونده ... در اینصورت هم از partialview استفاده کنم ؟
اگر آره ... لینکی که در اینبار به بتونه من رو کمک کنه دارید ؟ (خوندن اطلاعات و ...)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۲ ۱۴:۴

استفاده از PartialView یکی از روش‌های Refactoring یک View شلوغ هست. [اطلاعات بیشتر](#)

پ.ن.

برای پیگیری مسایل شخصی و تجاری خودتون، یا نیاز به مشاور خصوصی دارید یا از انجمن‌های مرتبط خارج از این سایت استفاده کنید.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۲۳ ۲۰:۲۵

با سلام.

ویو مدل من دارای چند لیست است که هرکدام از یک جدول خاص از دیتابیس اطلاعاتشان را بدست می‌آورند. حالا اگر کاربر اطلاعات را اشتباه وارد کرد و ما دوباره قرار است فرم را به او با همان اطلاعات قبلی نمایش دهیم، آیا منطقی است که برای پرکردن لیست‌های مورد نیاز دوباره به دیتابیس کوئری بزنیم یا روش بهتری وجود دارد؟ با تشکر.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۴/۲۴ ۹:۳۸

روش با سربار کم استفاده از Ajax است و به روز رسانی فقط قسمتی از صفحه بجای به روز رسانی کل یک صفحه شلوغ.

نویسنده: محمد آزاد
تاریخ: ۱۳۹۲/۰۴/۲۴ ۱۳:۵۵

من می‌خواستم بدونم ViewModel با DTO فرقی داره؟

الان اگه من یک ویو دارم که با اطلاعات چند Dto کار میکنه آیا تعریف یک viewModel بصورت زیر صحیح هستش؟

```
public class ProjectViewModel{
    public ProjectDto Project{ get; set; }
    public ProjectProgramDto ProjectProgram { get; set;}
    public IEnumerable<SelectListItem> PriorityListItems { get; set; }
}
```

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۴/۲۴ ۱۴:۴

یک رای‌گیری در این مورد در حال انجام است

« در ASP.NET MVC آیا ViewModel همان DTO است؟ »

نویسنده: محمد آزاد
تاریخ: ۱۳۹۲/۰۴/۲۴ ۱۴:۱۵

یعنی جواب قطعی برایش نیست؟ چون رای‌گیری نمیتونه ملاک خوبی باشه بنظرم

نویسنده: ح مراداف
تاریخ: ۱۳۹۲/۱۱/۲۴ ۱۳:۴۴

با سلام؛ لطفا درباره @section speakers یکم توضیح بدین. این @Section به چه کار آید ؟
حدس من اینه که با این دستور بشه توی یک مدل کلی بخش (مثل مائول) رو تعریف کرد و توی یک ویو بتونیم در جاهای مختلف این Section‌ها رو نمایش بدیم ؟ (عدم نیاز به چند Partial View و نوشتن همه توی یک ویو)

نویسنده: وحید نصیری

تاریخ: ۱۳:۵۳ ۱۳۹۲/۱۱/۲۴

در قسمت «[تزریق نواحی ویژه یک View در فایل layout](#)» به این موضوع پرداخته شده.

نویسنده: مجتبی فخاری

تاریخ: ۰:۲ ۱۳۹۲/۱۲/۱۵

با سلام؛ برای استفاده از ViewModel در زمان اجرا بعد از این که روی دکمه جدید کلیک می‌کنم این خطا را می‌دهد:

Either ErrorMessageString or ErrorMessageResourceName must be set, but not both.

میشه راهنمایی کنید؟

نویسنده: وحید نصیری

تاریخ: ۰:۱۱ ۱۳۹۲/۱۲/۱۵

[این خطا](#) را [در گوگل](#) جستجو کنید. [در اولین لینک](#) پاسخ را پیدا می‌کنید.

نویسنده: منصور جعفری

تاریخ: ۲۰:۵۷ ۱۳۹۲/۱۲/۲۶

سلام؛ وقتی رو صفحه اصلی سایت کاربر روی ادامه مطلب کلیک میکنه و وارد صفحه مربوط به اون مطلب میشه، صفحه جدید از یک پارشال ویوو تشکیل شده که دارای یک تگ فرم هست برای ارسال نظرات کاربران و هم چنین یک پارشال ویوو دیگه برای نمایش مطلب کامل اون لینک میباشد. برای استفاده هردو مدل یک view model تعریف شده که شامل هر دو موجودیت‌ها است. وقتی توی ویوو اصلی دکمه ادامه مطلب زده میشه این ارور داده میشه:

The model item passed into the dictionary is of type 'Class.Domain.Entities.Product', but this dictionary requires a model item of type 'WebAli.ViewModel.ProductCommentListViewModel'.

نویسنده: وحید نصیری

تاریخ: ۲۳:۱۲ ۱۳۹۲/۱۲/۲۶

این خطا به این معنا است که در اکشن متد کنترلر متناظر، خروجی نهایی و مدل ارسال شده به View، از جنس تک شیء Product است؛ اما این View نیاز به دریافت مدلی از کنترلر، از جنس ProductCommentListViewModel دارد (نوع خروجی اکشن متد را بررسی کنید).

حتی در حالت RenderPartial هم ممکن است این خطا را دریافت کنید. کلا در هر جایی که شیءایی به یک View برای رندر ارسال می‌شود، نوع آن و نوع مدل تعریف شده View را تطابق دهید.

نویسنده: اسدی

تاریخ: ۸:۳۳ ۱۳۹۳/۰۲/۰۲

سلام، من توی مدل پراپرتی از نوع DateTime؟ دارم که توی view اون رو با کنترلر datepicker نمایش میدم. مشکل اینجاست که مقداری که به مدل پاس میشه، تاریخ شمسی هستش و مدل انتظار datetime رو داره. نهایتاً توی uow.SaveChanges با این خطا مواجه میشم:

the conversion of a datetime2 data type to a datetime data type resulted in an out-of-range value

نویسنده: وحید نصیری

تاریخ: ۹:۳۶ ۱۳۹۳/۰۲/۰۲

مراجعه کنید به قسمت « [سفری سازی model binder پیش فرض ASP.NET MVC](#) » انتهای مطلب

نویسنده: جمشیدی فر
تاریخ: ۱۸:۱۶ ۱۳۹۳/۰۵/۲۹

من یک پروژه چندلایه دارم که شامل لایه‌های Domain که همه Entityها درون آن تعریف شده اند و لایه سرویس و لایه Presentation. مشکلی در مورد Validation سمت کلاینت دارم. هنگامی که برای یک View از مدلی که در لایه Domain تعریف شده استفاده میکنم، Validation سمت کلاینت و بدون رفرش صفحه انجام میشه اما هنگامی که از ViewModel تعریف شده در لایه Presentation استفاده می‌کنم، Validation با رفرش صفحه انجام میشه. دلیلش چیه؟

نویسنده: سروش
تاریخ: ۱۹:۵۸ ۱۳۹۳/۰۷/۲۱

با سلام و درود به شما عزیزان
من به مشکلی دارم و اون هم اینکه من می‌خواهم یک select new روی مدلم بزنم ولی چون نام فیلدها جدید به همیم خاطر در Partial View یا View اون ها رو نمی‌شناسه آیا راهکاری دارید؟

```
var models = from x in db.tblFinalProjects
              select x;
              {
                PTitle=x.xProjectTitle,
                ZTitle=x.tblZone.xCaption,
                PLastProgress=x.xLastProgress,
                PStatus=x.tblCurrentStatu.xCaption,
                PMasool="",
              };
```

این متغیر من هستش ولی مدل من tblFinalProject هست که بصورت Strongly Type به View پاس میدم اما نمی‌تونم روی فیلدهای جدید کار کنم . چون مدل اصلی رو میذاره ولی سلکت جدید من تلفیقی از جداولی است که به هم ارتباط دارن
مرسی

نویسنده: محسن خان
تاریخ: ۲۱:۵۳ ۱۳۹۳/۰۷/۲۱

معادل فیلدهایی که select کردی، یک کلاس جدید ViewModel درست کن. بعدی این select رو روی اون انجام بده (بجای اینکه مثل الان anonymous type باشه). نوع View رو هم از نوع لیستی از همین ViewModel جدید تعیین کن. به این نکته [LINQ](#) [projection](#) می‌گن.

```
public class MyViewModel
{
    public string PTitle {set;get;}
    // مابقی خواص در اینجا
}

var list = from x in db.tblFinalProjects
            select new MyViewModel
            {
                PTitle=x.xProjectTitle,
                ZTitle=x.tblZone.xCaption,
                PLastProgress=x.xLastProgress,
                PStatus=x.tblCurrentStatu.xCaption,
                PMasool="",
            };
```


به صورت پیش فرض دسترسی به تمامی اکشن‌ها مجاز است مگر اینکه آن اکشن به تگ Authorize مزین شود. حال [Best Practice](#) این است که حتی اگر شما یک یا دو اکشنی دارید که نیاز است کاربرای خاصی به آن‌ها دسترسی داشته باشند بهتر است که دسترسی به تمام اکشن‌ها محدود شود و بعد آن اکشن‌هایی که نیاز است دسترسی عمومی داشته باشند، بهشون دسترسی داده بشه. در واقع هدف از این Best Practice جلوگیری از قلم افتادن یک اکشن به اشتباه، هنگام دادن تگ Authorize هست.

خوب، برای انجام اینکار از فیلترهای سفارشی سراسری (Global Filters) استفاده می‌کنیم. کافیه خط زیر رو به کلاس FilterConfig اضافه کنید.

```
filters.Add(new AuthorizeAttribute());
```

با این کار تمام اکشن‌های شما با تگ Authorize مزین می‌شوند و تمام دسترسی‌ها محدود.

نکته: این روش تا قبل از ASP.NET MVC 4 فقط برای سیستم‌هایی که هیچ اکشنی با دسترسی عمومی نداشتن جوابگو بوده و در صورت داشتن چنین اکشنی این روش جوابگوی شما نیست. ولی از ASP.NET MVC 4 به بعد با اضافه شدن تگ [AllowAnonymous] این مشکل حل شده.

حالا که تمامی اکشن‌ها محدود شدن، حالا نوبت به اکشن‌هایی می‌رسه که دسترسی عمومی به اونها آزاده. برای این کار به راحتی از تگ [AllowAnonymous] استفاده می‌کنیم

```
[AllowAnonymous]
public ActionResult Index()
{
    return View();
}
```

فایل پروژه [BestPracticeForAuthenticatingUsers.rar](#)

نظرات خوانندگان

نویسنده: آیمو

تاریخ: ۱۳۹۲/۰۷/۰۳ ۹:۲۳

سلام . من برای پروژه ام همین کارو کردم . صفحه اول من از توی یک ناحیه (Area) لود میشه . به این صورت که درخواست میره به یک اکشن مثل Home از کنترلر ControlPanel که این اکشن Allow Anonymous هست . بعد از داخل اون Redirect میشه به همون Area و داخل اون هم به کنترلر Home و به اکشن Index هست . اما موقع لود شدن redirect میشم به صفحه لاگین . لطفا راهنمایی کنید.

نویسنده: محسن خان

تاریخ: ۱۳۹۲/۰۷/۰۳ ۹:۴۲

«بعد از داخل اون Redirect میشه» خودکار هست یا دستی؟ اگر خودکار هست، خوب طبیعی هست جایی که AllowAnonymous نداره (یعنی اکشن متد بعدی که به اون Redirect شده)، کاربر رو به صفحه لاگین هدایت کنه. Global Filters یعنی دقیقاً همین. یعنی اعمال سراسری به همه جا، مگر اینکه با AllowAnonymous مستثنی شود.

نویسنده: آیمو

تاریخ: ۱۳۹۲/۰۷/۰۳ ۱۰:۲

اره خب حرفه شما درسته . اما هر دوتا اکشن من AllowAnonymous هست . در واقع از یک اکشن AllowAnonymous میخوام برم به یه اکشن AllowAnonymous داخل یه ناحیه . مشکل اینجاست که نمیره و میره تو صفحه لاگین .

نویسنده: محسن خان

تاریخ: ۱۳۹۲/۰۷/۰۳ ۱۰:۱۹

ممکن هست در اون صفحه دوم داری چیزی رو نمایش می‌دی که ارجاعی داره به یک اکشن متد محافظت شده. مثلاً صفحات چند قسمتی هست. یک قسمت داره از یک اکشن متد غیر عمومی شده اطلاعات دریافت می‌کنه.

نویسنده: ایمان

تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۴:۳۴

سوال من اینه که اگه بخواهیم این محدودیت رو محدودتر کنیم [مثلاً فقط نقش abc] دسترسی داشته باشه، اونوقت باید ["Authorize(Roles=abc)"] رو روی اکشن بگذاریم؟

نویسنده: محمد صاحب

تاریخ: ۱۳۹۳/۰۴/۰۳ ۲۲:۰۶

اگه قراره روی فقط رو یه اکشن اعمال بشه درسته میتونی حتی میتونی روی کنترلر بزاری که تمام اکشن‌های اون کنترلر رو محدود کنه و اگه میخوای با روشی که تو این پست اومده عمل کنی باید خودت یک Authorize Attribute سفارشی بنویسی و به [فیلترهای سراسری](#) ت ادد کنی.

در پروژه فروشگاهی تحت Asp.Net MVC 4 بهترین روش برای ایجاد یک سبد خرید استفاده از یک Api Controller می‌باشد.

من در پروژه‌ای که در MVC 3 داشتم این مورد را بدین شکل انجام داده بودم که با ایجاد یک کلاس و درج چند سطر کد در Global.asax این مورد حل میشد و در Api Controller ای که میخواستیم اطلاعات را در آن به کمک Session دریافت یا ویرایش کنیم، امکان دسترسی به Session را داشتم:

یک کلاس ایجاد کرده و کدهای زیر را در داخل آن درج میکنیم:

```
//using System.Web.Http.WebHost;
//using System.Web.Routing;
//using System.Web.SessionState;

public class MyHttpControllerHandler : HttpControllerHandler, IRequiresSessionState
{
    public MyHttpControllerHandler(RouteData routeData): base(routeData)
    {
    }
}

public class MyHttpControllerRouteHandler : HttpControllerRouteHandler
{
    protected override IHttpHandler GetHttpHandler(RequestContext requestContext)
    {
        return new MyHttpControllerHandler(requestContext.RouteData);
    }
}
```

و برای اینکه این مورد، در برنامه‌ای که ساختیم کار کند باید در Global.asax این کدها را درج کنیم (دوباره تاکید می‌کنم این نکته فقط در MVC 3 کار می‌کند)

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

    var apiroute = routes.MapHttpRoute(
        name: "DefaultApi",
        routeTemplate: "api/{controller}/{id}",
        defaults: new { id = RouteParameter.Optional }
    );
    apiroute.RouteHandler = new MyHttpControllerRouteHandler();

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );
}
```

با تنظیمات فوق در برنامه‌های Asp.NET MVC 3 امکان استفاده از Session در API Controller ها در دسترس قرار می‌گیرد. اما این کار در ASP.NET MVC 4 قابل استفاده نیست و خیلی بهتر و راحتتر انجام می‌شود (میتوانید به WebApiConfig.cs در فولدر App_Start پروژه مراجعه نمایید)

حال چه تنظیماتی نیاز است تا دوباره این امکان برقرار شود؟

تنها کدهای ذیل را در Global.asax درج نمایید و امکان استفاده از Session را در API Controller های MVC4، ایجاد نمایید.

```
public class MvcApplication : System.Web.HttpApplication
{
    public override void Init()
    {
        this.PostAuthenticateRequest += MvcApplication_PostAuthenticateRequest;
    }
}
```

```
        base.Init();
    }

    void MvcApplication_PostAuthenticateRequest(object sender, EventArgs e)
    {
        System.Web.HttpContext.Current.SetSessionStateBehavior(
            SessionStateBehavior.Required);
    }

    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();

        WebApiConfig.Register(GlobalConfiguration.Configuration);
        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
        RouteConfig.RegisterRoutes(RouteTable.Routes);
        BundleConfig.RegisterBundles(BundleTable.Bundles);
        AuthConfig.RegisterAuth();
    }
}
```

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۲۱:۳۷ ۱۳۹۲/۰۷/۱۷

لطفا این قسمت « بهترین روش برای ایجاد یک سبد خرید استفاده از یک Api Controller می باشد.» را بیشتر توضیح بدید. با تشکر.

نویسنده: بهمن خلفی
تاریخ: ۲۱:۴۶ ۱۳۹۲/۰۷/۱۷

در اکثر فروشگاههایی که با Asp.net MVC توسعه پیدا کردند اضافه کردن یک کالا به سبد خرید پروسه ای زمان بر بوده و کاربر پسند نیست در حالی که در فروشگاههای متن باز مشابه این عمل بصورت زیبا و کاملا پرسرعت انجام میشود . API یکی از مباحث خوبی است که در MVC براحتی قابل استفاده بوده و این قدرت را به برنامه نویسی میده که بتواند از مباحثی مانند Ajax یا JSON مثلا در سبد کاربر استفاده کند.
بهترین روش به نظر من است ! بدلیل اینکه بسیار راحت و امن است . راه اندازی آن در حد 10 الی 15 دقیقه بیشتر طول نمیکشد و شما میتوانید تمامی مباحثی مانند احراز هویت و ... را طبق اصول MVC روی همه Actionهای مورد نیاز اعمال کنید.
اگر اغراق شده فقط یک نظر شخصی است. (-)

نویسنده: محسن خان
تاریخ: ۲۱:۴۹ ۱۳۹۲/۰۷/۱۷

ممنون. JSON و Ajax با کنترلرهای معمولی MVC هم قابل دسترسی است. چه مزیت بیشتری رو با Web API به دست آوردید؟

نویسنده: بهمن خلفی
تاریخ: ۲۱:۵۷ ۱۳۹۲/۰۷/۱۷

بله این امکان وجود دارد اما شما این امکان را میتوانید در یک برنامه بصورت کلی استفاده نمائید یعنی در هر جای برنامه باشید میتوانید به این بخش دسترسی داشته باشید .
از طرفی شما با استفاده از API دیگر برخی از اطلاعات یا آیتمهای صفحه را بدلیل Post شدن صفحه (Dispose) از دست نخواهید داد.
استفاده از Web API ها یک عمل عمومی بحساب می آید اما در این مطلب مقصود دسترسی به Session در API Controller ها میباشد.

نویسنده: مهدیار
تاریخ: ۱۹:۲۲ ۱۳۹۲/۰۷/۱۹

استفاده از session ؟
به نظرم یک جدول موقت برای کارت می گزینید بهتر بود . این هم یک [مطلب](#) مفید برای استفاده نکردن از session.

نویسنده: محسن خان
تاریخ: ۲۱:۲۶ ۱۳۹۲/۰۷/۱۹

کوکی هم برای یک طراحی Http friendly خوب هست. خصوصا اینکه یک شخص آنچنان اقلام زیادی را که هربار خرید نمی کند. همچنین [HTML5 local storage](#) نیز مفید است.

نویسنده: بهمن خلفی
تاریخ: ۸:۵۷ ۱۳۹۲/۰۷/۲۰

بله دوست عزیز ، استفاده از Session مزایا و معایب خودش را به همراه دارد.

اما در یک سیستم فروشگاهی حذف و اضافه شدن کالا در سبد خرید بصورت مکرر انجام میشود و برای اینکه شما چه کاربران مهمان و چه کاربران عضو را مدیریت نموده و بتوانید ترافیک روی بانک اطلاعاتی خود را کاهش دهید میتوانید از Session استفاده نمایید . (بطور مثال اگر 1000 نفر در یک سایت فروشگاهی که حداقل 100 کالا ارائه شده است را به سبد خرید خود اضافه یا تعداد و ... آنها را ویرایش نمایند در بهترین حال 100000 درخواست به بانک اطلاعاتی ارسال و دریافت خواهد شد)

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۷/۲۰ ۸:۵۹

بله دوست عزیز ، این هم یک امکان بسیار خوبی است که در سمت کاربر میتوان ازش استفاده نمود.
در مطالب آینده سعی خواهد شد کاربرد این امکان در یک برنامه کاربردی ارائه شود. باتشکر

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۴/۰۸ ۲۰:۴۵

برای طراحی سبد کالا یا هر نوع آیتم مشابه آن بهتر هست که از روش هایی غیر از سشن بهره برد. چون سشن خودش سنگینی زیادی داره و با افزایش کاربران همزمان این افزایش مصرف حافظه بیشتر هم خواهد شد.

مشکلات دیگری هم که داره این هست که در سشن یک شی زمان دار هست و اگر کاربر 15مثلاً دقیقه کار نکند بعد از آن سبد خالی می شود که بهتر هست انتخابها همواره حفظ گردد . پس بهترین راهها استفاده از [کوکی \(+ + \)](#) است که در این مقاله نحوه [ذخیره هر نوع داده](#) (به همراه فشرده سازی) ذکر شده است و مورد دوم استفاده از [indexedDB و local storage](#) است که به زودی قسمت دوم هم منتشر می شود.

مورد بعدی اینکه در صورت به روز رسانی وب سایت به خصوص وب کانفیگ سشنها ریست خواهند شد و سبدی که کاربر پر کرده است در آن لحظه از بین خواهد رفت که با استفاده روش های کلاینتی چنین مشکلی ایجاد نمی شود.

مورد بعدی اینکه با استفاده از جاوااسکریپت و توابع تحت کلاینت سرعت بهتری در اجرا و رسم سبد کالا می توانید داشته باشید بدون اینکه بار زیادی را به سرور تحمیل کنید.

نکته تکمیلی اینکه فقط باید در رسم مجدد سبد کالا تحت کلاینت باید موجودی ها و قیمت ها و تخفیف ها و ... هم مجددا بررسی شوند که اطلاعات قبلی نمایش داده نشود که البته این بر میگرده به طراحی ساختاری که ترتیب داده اید.
استفاده از جدول جداگانه برای نگهداری اطلاعات هم روش مناسبی حساب می شود که انتخاب بین اینها به نظر برنامه نویس مربوط می شود.

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۵/۰۳ ۱:۱۳

یک نکته دیگر که در مورد ذخیره آیتمها مثل یک سبد کالا به ذهنم خورد این هست که روش ذخیره در یک جدول در سمت سرور در بعضی موارد کمک حال بسیار خوبی خواهد بود. در این حالت از مزایای زیر بهره مند خواهیم شد:
اینکه بفهمیم در حال حاضر چه محصولی بیشتر در سبد کالای کاربران قرار گرفته است که هنوز در خرید آن مردد هستند؟ آیا با یک تخفیف مناسب به آن محصول می توان آن ها را به سمت تایید سبد خرید تشویق کرد؟
و دیگر گزارشات مشابه

قبلا مطالبی در سایت راجع به [نوع داده شمارشی یا Enum](#) و همچنین [CheckBoxList](#) و [RadioButtonList](#) وجود دارد. اما در این مطلب قصد دارم تا یک روش متفاوت را برای تولید و بهره گیری از CheckBoxList با استفاده از نوع داده‌های شمارشی برای شما ارائه کنم.

فرض کنید بخواهید به کاربر این امکان را بدهید تا بتواند چندین گزینه را برای یک فیلد انتخاب کند. به عنوان یک مثال ساده فرض کنید گزینه ای از مدل، پارچه‌های مورد علاقه یک نفر هست. کاربر می‌تواند چندین پارچه را انتخاب کند. و این فرض را هم بکنید که به لیست پارچه‌ها گزینه دیگری اضافه نخواهد شد. پارچه (Fabric) را مثلا می‌توانیم به صورت زیر تقسیم بندی کنیم:

پنبه (Cotton)

ابریشم (Silk)

پشم (Wool)

ابریشم مصنوعی (Rayon)

پارچه‌های دیگر (Other)

با توجه به اینکه دیگر قرار نیست به این لیست گزینه دیگری اضافه شود می‌توانیم آنرا به صورت یک نوع داده شمارشی (Enum) تعریف کنیم. مثلا بدین صورت:

```
public enum Fabric
{
    [Description("پنبه")]
    Cotton,

    [Description("ابریشم")]
    Silk,

    [Description("پشم")]
    Wool,

    [Description("ابریشم مصنوعی")]
    Rayon,

    [Description("پارچه‌های دیگر")]
    Other
}
```

حال فرض کنید View Model زیر فیلدی از نوع نوع داده شمارشی Fabric دارد:

```
public class MyViewModel
{
    public Fabric Fabric { get; set; }
}
```

توجه داشته باشید که فیلد Fabric از کلاس MyViewModel باید چند مقدار را در خود نگهداری کند. یعنی می‌تواند هر کدام از گزینه‌های Cotton, Silk, Wool, Rayon, Other به صورت جداگانه یا ترکیبی باشد. اما در حال حاضر با توجه به اینکه یک فیلد Enum معمولی فقط می‌تواند یک مقدار را در خودش ذخیره کند قابلیت ذخیره ترکیبی مقادیر در فیلد Fabric از View Model بالا وجود ندارد.

اما راه حل این مشکل استفاده از پرچم (Flags) در تعریف نوع داده شمارشی هست. با استفاده از پرچم نوع داده شمارشی بالا به صورت زیر باید تعریف شود:

```
[Flags]
public enum Fabric
```

```
{
    [Description("پنبه")]
    Cotton = 1,

    [Description("ابریشم")]
    Silk = 2,

    [Description("پشم")]
    Wool = 4,

    [Description("ابریشم مصنوعی")]
    Rayon = 8,

    [Description("پارچه های دیگر")]
    Other = 128
}
```

همان طور که می بینید از عبارت [Flags] قبل از تعریف enum استفاده کرده ایم. همچنین هر کدام از مقادیر ممکن این نوع داده شماری با توانهایی از 2 تنظیم شده اند. در این صورت یک نمونه از این نوع داده می تواند چندین مقدار را در خودش ذخیره کند.

برای آشنایی بیشتر با این موضوع به کدهای زیر نگاه کنید:

```
Fabric cotWool = Fabric.Cotton | Fabric.Wool;
int cotWoolValue = (int) cotWool;
```

به وسیله عملگر | می توان چندین مقدار را در یک نمونه از نوع Fabric ذخیره کرد. مثلاً متغیر cotWool هم دارای مقدار Fabric.Cotton و هم دارای مقدار Fabric.Wool هست. مقدار عددی معادل متغیر cotWool برابر 5 هست که از جمع مقدار عددی Fabric.Cotton و Fabric.Wool به دست آمده است.

حال فرض کنید فیلد Fabric از View Model ذکر شده (کلاس MyViewModel) را به صورت لیستی از چک باکس ها نمایش دهیم. مثل زیر:

MyViewModel

☐ پنبه
 ☐ ابریشم
 ☐ پشم
 ☐ ابریشم مصنوعی
 ☐ پارچه های دیگر

Create

شکل (الف)

سپس بخواهیم تا کاربر بعد از انتخاب گزینه های مورد نظرش از لیست بالا و پست کردن فرم مورد نظر، بایندر وارد عمل شده و فیلد Fabric را بر اساس گزینه هایی که کاربر انتخاب کرده مقداردهی کند.

برای این کار از [پروژه MVC Enum Flags](#) کمک خواهیم گرفت. این پروژه شامل یک Html Helper برای تبدیل Enum به یک CheckBoxList و همچنین شامل Model Binder مربوطه هست. البته بعضی از کدهای Html Helper آن احتیاج به تغییر داشت که

آنها انجام دادم ولی بایندر آن بسیار خوب کار می‌کند.

خوب html helper مربوط به آن به صورت زیر می‌باشد:

```
public static IHtmlString CheckBoxesForEnumFlagsFor<TModel, TEnum>(this HtmlHelper<TModel> htmlHelper,
Expression<Func<TModel, TEnum>> expression)
{
    ModelMetadata metadata = ModelMetadata.FromLambdaExpression(expression, htmlHelper.ViewData);
    Type enumModelType = metadata.ModelType;

    // Check to make sure this is an enum.
    if (!enumModelType.IsEnum)
    {
        throw new ArgumentException("This helper can only be used with enums. Type used was: " +
enumModelType.FullName.ToString() + ".");
    }

    // Create string for Element.
    var sb = new StringBuilder();

    foreach (Enum item in Enum.GetValues(enumModelType))
    {
        if (Convert.ToInt32(item) != 0)
        {
            var ti = htmlHelper.ViewData.TemplateInfo;
            var id = ti.GetFullHtmlFieldId(item.ToString());

            //Derive property name for checkbox name
            var body = expression.Body as MemberExpression;
            var propertyName = body.Member.Name;
            var name = ti.GetFullHtmlFieldName(propertyName);

            //Get currently select values from the ViewData model
            TEnum selectedValues = expression.Compile().Invoke(htmlHelper.ViewData.Model);

            var label = new TagBuilder("label");
            label.Attributes["for"] = id;
            label.Attributes["style"] = "display: inline-block;";
            var field = item.GetType().GetField(item.ToString());

            // Add checkbox.
            var checkbox = new TagBuilder("input");
            checkbox.Attributes["id"] = id;
            checkbox.Attributes["name"] = name;
            checkbox.Attributes["type"] = "checkbox";
            checkbox.Attributes["value"] = item.ToString();

            if ((selectedValues as Enum != null) && ((selectedValues as Enum).HasFlag(item)))
            {
                checkbox.Attributes["checked"] = "checked";
            }
            sb.AppendLine(checkbox.ToString());

            // Check to see if DisplayName attribute has been set for item.
            var displayName = field.GetCustomAttributes(typeof(DisplayNameAttribute), true)
                .FirstOrDefault() as DisplayNameAttribute;
            if (displayName != null)
            {
                // Display name specified. Use it.
                label.SetInnerText(displayName.DisplayName);
            }
            else
            {
                // Check to see if Display attribute has been set for item.
                var display = field.GetCustomAttributes(typeof(DisplayAttribute), true)
                    .FirstOrDefault() as DisplayAttribute;
                if (display != null)
                {
                    label.SetInnerText(display.Name);
                }
                else
                {
                    label.SetInnerText(item.ToDescription());
                }
            }
            sb.AppendLine(label.ToString());

            // Add line break.
            sb.AppendLine("<br />");
        }
    }
}
```

```

    }
}
return new HtmlString(sb.ToString());
}

```

در کدهای بالا از متد الحاقی ToDescription نیز برای تبدیل معادل انگلیسی به فارسی یک مقدار از نوع داده شمارشی استفاده کرده ایم.

```

public static string ToDescription(this Enum value)
{
    var attributes =
(DescriptionAttribute[])value.GetType().GetField(value.ToString()).GetCustomAttributes(typeof(DescriptionAttribute), false);
    return attributes.Length > 0 ? attributes[0].Description : value.ToString();
}

```

برای استفاده از این Html Helper در View کد زیر را می‌نویسیم:

```
@Html.CheckBoxesForEnumFlagsFor(x => x.Fabric)
```

که باعث تولید خروجی که در تصویر (الف) نشان داده شد می‌شود. و همچنین مدل بایندر مربوط به آن به صورت زیر هست:

```

public class FlagEnumerationModelBinder : DefaultModelBinder
{
    public override object BindModel(ControllerContext controllerContext, ModelBindingContext bindingContext)
    {
        if (bindingContext == null) throw new ArgumentNullException("bindingContext");
        if (bindingContext.ValueProvider.ContainsPrefix(bindingContext.ModelName))
        {
            var values = GetValue<string[]>(bindingContext, bindingContext.ModelName);
            if (values.Length > 1 && (bindingContext.ModelType.IsEnum && bindingContext.ModelType.IsDefined(typeof(FlagsAttribute), false)))
            {
                long byteValue = 0;
                foreach (var value in values.Where(v => Enum.IsDefined(bindingContext.ModelType, v)))
                {
                    byteValue |= (int)Enum.Parse(bindingContext.ModelType, value);
                }
                return Enum.Parse(bindingContext.ModelType, byteValue.ToString());
            }
            else
            {
                return base.BindModel(controllerContext, bindingContext);
            }
        }
        return base.BindModel(controllerContext, bindingContext);
    }

    private static T GetValue<T>(ModelBindingContext bindingContext, string key)
    {
        if (bindingContext.ValueProvider.ContainsPrefix(key))
        {
            ValueProviderResult valueResult = bindingContext.ValueProvider.GetValue(key);
            if (valueResult != null)
            {
                bindingContext.ModelState.SetModelValue(key, valueResult);
                return (T)valueResult.ConvertTo(typeof(T));
            }
        }
        return default(T);
    }
}

```

این مدل بایندر را باید به این صورت در متد Application_Start فایل Global.asax فراخوانی کنیم:

```
ModelBinders.Binders.Add(typeof(Fabric), new FlagEnumerationModelBinder());
```

مشاهده می‌کنید که در اینجا دقیقاً مشخص کرده ایم که این مدل بایندر برای نوع داده شمارشی Fabric هست. اگر نیاز دارید تا این بایندر برای نوع داده‌های شمارشی دیگری نیز به کار رود نیاز هست تا این خط کد را برای هر کدام از آنها تکرار کنید. اما راه حل بهتر این هست که کلاسی به صورت زیر تعریف کنیم و تمامی نوع داده‌های شمارشی که باید از بایندر بالا استفاده کنند را در یک پراپرتی آن برگشت دهیم. مثلاً بدین صورت:

```
public class ModelEnums
{
    public static IEnumerable<Type> Types
    {
        get
        {
            var types = new List<Type> { typeof(Fabric) };
            return types;
        }
    }
}
```

سپس به متد Application_Start رفته و کد زیر را اضافه می‌کنیم:

```
foreach (var type in ModelEnums.Types)
{
    ModelBinders.Binders.Add(type, new FlagEnumerationModelBinder())
}
```

اگر گزینه‌های پشم و ابریشم مصنوعی را از CheckBoxList تولید شده انتخاب کنیم، بدین صورت:

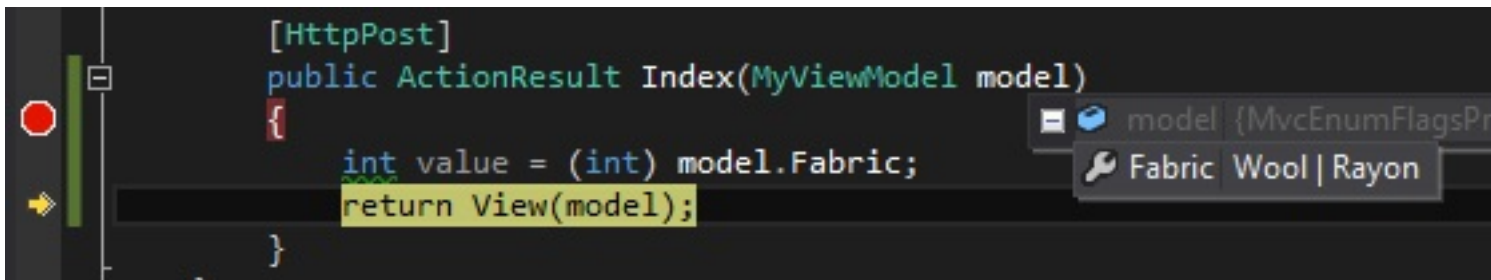
MyViewModel

- ☐ پنبه
- ☐ ابریشم
- ☒ پشم
- ☒ ابریشم مصنوعی
- ☐ پارچه های دیگر

Create

شکل (ب)

و سپس فرم را پست کنید، موردی شبیه زیر مشاهده می‌کنید:



شکل (ج)

همچنین مقدار عددی معادل در این جا برابر 12 می باشد که از جمع دو مقدار Wool و Rayon به دست آمده است. بدین ترتیب در یک فیلد از مدل، گزینه های انتخابی توسط کاربر قرار گرفته شده اند.

پروژه مربوط به این مثال را از لینک زیر دریافت کنید:

[MvcEnumFlagsProjectSample.zip](#)

پی نوشت : پوشه های bin و obj و packages جهت کاهش حجم پروژه از آن حذف شده اند. برای بازسازی پوشه packages لطفاً به مطلب [بازسازی کامل پوشه packages بسته های NuGet به صورت خودکار](#) مراجعه کنید.

نظرات خوانندگان

نویسنده: علی

تاریخ: ۱۸:۵۱ ۱۳۹۲/۰۸/۰۱

من یک گرید تلریک دارم که یک فیلد چکباکس کنارش هست و مثلا لیستی از یوزرها رو این گرید شامل میشه . آیتم‌های انتخابی رو چطوری میتونم به کنترلر ارسال کنم (ورودی کنترلرمو چی بگیرم)...

نویسنده: وحید نصیری

تاریخ: ۱۸:۵۸ ۱۳۹۲/۰۸/۰۱

سؤال شما بیشتر به مطلب « [ASP.NET MVC در CheckBoxList](#) » مرتبط است تا enum ایی که ویژگی flag دارد. ضمنا گرید تلریک [مستندات خوبی دارد](#) که بهتر است به آن مراجعه کنید (مثال Ajax CheckBoxes هست که کدهای View و کنترلر آن نیز پیوست شدند).

نویسنده: علیرضا

تاریخ: ۱۲:۱۵ ۱۳۹۲/۰۸/۰۲

استفاده از Description برای Enum ها جالب ولی به نظر من کمی مشکل سازه و مشکل اصلی اون Code Wired کردن شرح هر جزء Enum هست. اگر با یک پروژه دوزبانه طرف باشیم چی؟ اگر کلا استراتژی ترجمه رشته‌ها در پروژه ما این طور باشه که از منابع خارجی مثل DB یا اسمبلی‌های Resource استفاده کنیم چی؟

نویسنده: وحید نصیری

تاریخ: ۱۲:۳۲ ۱۳۹۲/۰۸/۰۲

مراجعه کنید به پیشنیازهای این مباحث تکمیلی. مانند:

[تهیه سایت‌های چند زبانه و بومی سازی نمایش اطلاعات در ASP.NET MVC](#)

[ASP.NET MVC در Globalization](#)

اولی در مورد کار با ریسورس‌ها است و بومی سازی ویژگی‌ها نیز در آن لحاظ شده و دومی تهیه یک فریم ورک است برای کار با بانک اطلاعاتی و تامین منبع داده از این طریق

چنانچه قبلاً با فایل‌های Less کار کرده باشید، متوجه خواهید شد که به صورت پیش فرض و همانند فایل‌های css و js قابلیت افزوده شدن به Bundle.config را دارا نمی‌باشند. برای انجام این کار باید مراحل کوتاهی را طی نمایید:

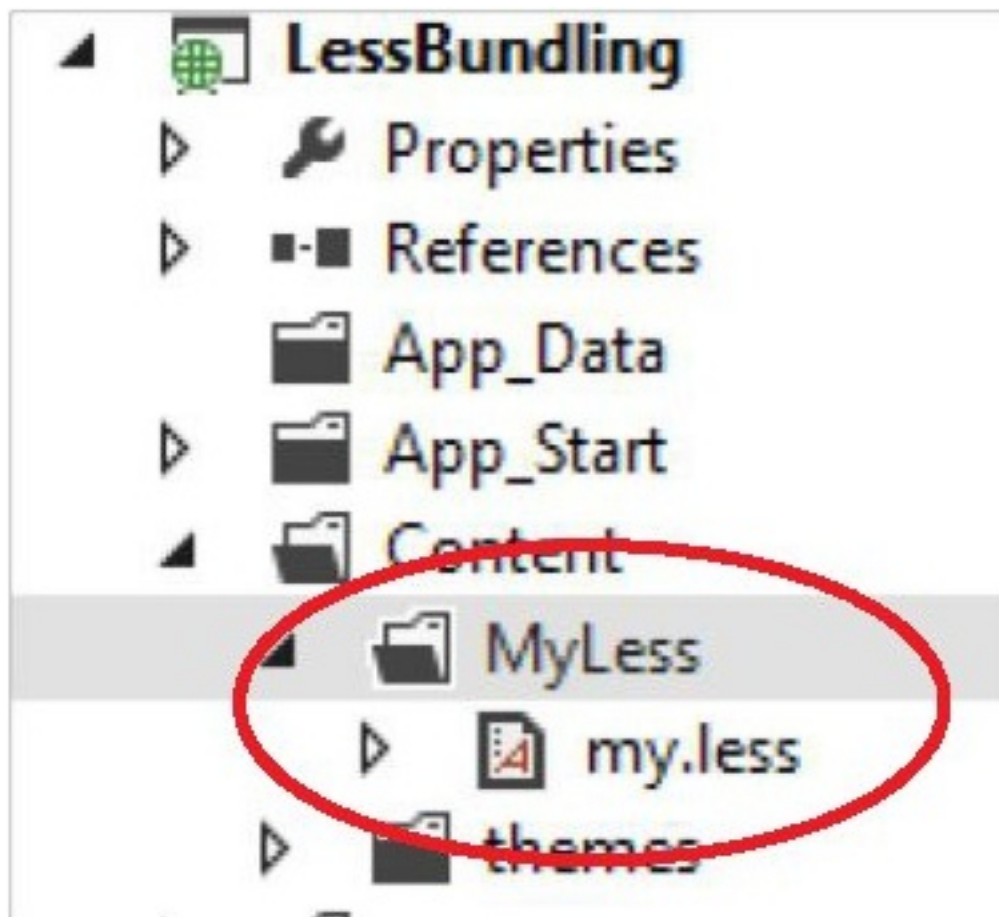
1- به منوی project و بخش Manage NuGet Packages ... رفته و dotless را جستجو و نصب نمایید.

2- کلاسی به نام "LessTransForm" ایجاد کنید که از "IBundleTransform" ارث بری کند، کدهای آن را به صورت ذیل تغییر دهید:

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

namespace LessBundling
{
    public class LessTransform:IBundleTransform
    {
        public void Process(BundleContext context, BundleResponse response)
        {
            response.Content = dotless.Core.Less.Parse(response.Content);
            response.ContentType = "text/css";
        }
    }
}
```

3- فولدري برای فایل‌های Less خود ایجاد کنید:



4- به App_Start/BundleConfig.cs رفته و کدهای ذیل در آن اضافه کنید:

```
var lessBundle = new Bundle("~/Content/Less").IncludeDirectory("~/Content/MyLess", "*.less");
lessBundle.Transforms.Add(new LessTransform());
lessBundle.Transforms.Add(new CssMinify());

bundles.Add(lessBundle);
```

Bundling به اتمام رسید! حال می‌توانید از کد ذیل در view ها و یا در Layout خود، همانند فایل‌های css و js استفاده کنید:

```
@Styles.Render("~/Content/Less")
```

[منبع](#)

[TwitterBootstrapMVC](#) یا به اختصار [BMVC](#) یک کتابخانه از Helper های مفید برای ساده سازی استفاده از Twitter Bootstrap در MVC می‌باشد .

در این کتابخانه امکانات مختلف Bootstrap از طریق Helper های نوشته شده برای MVC براحتی قابل استفاده می‌باشد و فرایند کد نویسی را ساده‌تر و در عین حال خواناتر می‌کند ، Helper های موجود در این کتابخانه به صورت زنجیره ای (fluent syntax) نوشته شده که استفاده از آن را سهولت می‌بخشد .

برای استفاده از آن در MVC 4 کافی است بعد از پیکر بندی Bootstrap ([راهنمایی](#)) به کتابخانه TwitterBootstrapMvc رفرنسی ایجاد کنید و با استفاده از این راهنما [نحوه استفاده](#) را فرا گیرید . همچنین می‌توانید آن را از طریق NuGet بارگذاری نمایید .

نسخه‌ی MVC4 آن را در اینجا برای شما نیز آپلود نمودم در زیر نمونه ای از استفاده از آن را می‌بینید

```
@Html.Bootstrap().LabelFor(x => x.UserName)
@Html.Bootstrap().TextBoxFor(m => m.UserName)
@Html.Bootstrap().PasswordFor(m => m.Password)
@Html.Bootstrap().FileFor(m => m.File)
@Html.Bootstrap().CheckBoxFor(m => m.IsActivated)
@Html.Bootstrap().RadioButtonFor(m => m.Gender, "male")
@Html.Bootstrap().DropDownListFor(m => m.State, Model.UsaStates)
@Html.Bootstrap().ListBoxFor(m => m.State, Model.UsaStates)
@Html.Bootstrap().TextAreaFor(m => m.Description)
```

ایجاد یک فرم

```
@using (Html.Bootstrap().Begin(new Form().Type(FormType.Inline)))
{
    @Html.Bootstrap().TextBoxFor(m => m.Email).Placeholder("Email")
    @Html.Bootstrap().PasswordFor(m => m.Password).Placeholder("Password")
    @Html.Bootstrap().CheckBoxFor(m => m.RememberMe).Label()
    @Html.Bootstrap().SubmitButton().Text("Sign in")
}
```

یک فرم Modal

```
@Html.Bootstrap().Button().Text("Show Modal").IconAppend(Icons.camera).TriggerModal("MyModal")

@using(var modal = Html.Bootstrap().Begin(new Modal() .Id("MyModal") .HtmlAttributes(new { @class
= "custom-class" })).Fade() )){
    using(modal.BeginHeader()){
        {
            <h2>Some header</h2>
        }
    }
    using(modal.BeginBody()){
        {
            <p>Some body<p>
        }
    }
    using(modal.BeginFooter()){
        {
            <p>Footer here.<p>
            @Html.Bootstrap().Button().Text("Close")
        }
    }
}
```


نظرات خوانندگان

نویسنده: Hamid NCH
تاریخ: ۱۳۹۲/۰۹/۱۳ ۱۲:۱۸

این کتابخانه محدودیت استفاده نداره؟ منظورم اینه کافیه این کتابخونه رو به رفرنس اد کنیم و استفاده کنیم! ممنون.

نویسنده: افشار محبی
تاریخ: ۱۳۹۲/۰۹/۱۳ ۱۲:۵۷

تا حالا فرصت نشده که به طور اساسی با Twitter Bootstrap درگیر بشوم. اما وجود این تعداد راهنما و ابزار جانبی آدم را امیدوار می کند که بتواند یک خروجی خیلی خوب از آن بگیرد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۳ ۱۲:۵۸

- داره ([مجاناً نیست](#)). البته فایل هایی که ایشون پیوست کردند به نظر محدودیت ندارند.
- ضمناً [باز هم هستند](#) یک سری Wrapper برای بوت استرپ که می توانند مورد استفاده قرار گیرند:
[TwitterBootstrapMvc](#) (نسخه سورس باز مطلب جاری است)
[Mvc Bootstrap Html Helper Extensions](#)
[Bootstrap Helpers](#) (معرفی در اینجا)
[Twitter Bootstrap Controls for ASP.NET](#)

نویسنده: رضا رضایی
تاریخ: ۱۳۹۲/۰۹/۱۳ ۲۳:۱۰

با سلام
متأسفانه این کتابخانه از نسخه 2 به بعد به صورت غیر رایگان عرضه شده است ، هر چند نسخه آپلود شده آخرین نسخه آن برای MVC 4 می باشد .
این کتابخانه برا اساس امکانات Bootstrap 3 نوشته شده است و هدفش تسهیل استفاده از Bootstrap 3 در MVC می باشد و بسیار کاربردیست .
دارای محدودیت خاصی نمی باشد و بسیاری از امکانات پیچیده Bootstrap را ساده نموده است

نویسنده: رضا رضایی
تاریخ: ۱۳۹۲/۰۹/۱۳ ۲۳:۱۴

twitter Bootstrap کتابخانه جامعی از CSS ها و JScript ها برای طراحی صفحات وب می باشد ک امروزه بسیار مقبول واقع شده است و ظاهری ساده و در عین حال کارآمد را برای وب سایت شما با پشتیبانی از انواع مختلف از نمایشگرها به ارمغان می آورد .

نویسنده: هومن
تاریخ: ۱۳۹۲/۰۹/۱۵ ۱۱:۲۸

سلام
مطلب بسیار مفیدی بود ممنون.
مشکلی که من دارم اینه که اولاً از طریق نیوگت قابل نصب نیست یعنی طبق سایت خودشون اسم بکجشو که مینویسم چیزی پیدا نمیکنه...
منم dll ها رو از سایت خودشون دانلود کردم و دستی add refrence کردم... در webconfig قسمت view ها هم دو تگ مربوط را اضافه کردم.

اما در ریزور تا اینجا **Bootstrap . Html** () را می شناسد اما بعد از پراتنز intelli-sense آن را نمی شناسد. مشکل از کجاست؟ ضمناً نسخه vs من 2013 و از mvc5 استفاده میکنم

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۵ ۱۱:۳۳

- نسخه سایت خودش سی روزه هست. باید یک فایل مجوز هم براش دریافت کنی، اینجا:
<https://www.twitterbootstrapmvc.com/Download> یا نسخه پیوست شده از متن رو دریافت و استفاده کن.
- برای MVC5 در اینجا: <http://www.nuget.org/packages/TwitterBootstrapMVC5>
- برای MVC4 در اینجا: <http://www.nuget.org/packages/TwitterBootstrapMVC>
[سایر موارد مرتبط](#)

نویسنده: رضا رضایی
تاریخ: ۱۳۹۲/۰۹/۱۶ ۱۹:۳۸

سلام ، از NuGet قابل دریافت ولی نیاز به لایسنس داره که باید بخری و نسخه مجانش فقط سی روزست ، فایل های که پیوست کردم این مشکل و نداره و آخرین نسخهست ، راه درستش اینه که اول Bootstrap 3 را کامل تو برنامهت پیکربندی کنی و از جواب دادنش اطمینان حاصل کنی بعد فقط کافیه فایل TwitterBootstrapMVC.dll و T4MVCExtensions.dll رفرنس بزنی ، بعد از اون باید در قسمت معرفی فضاهای نام در مسیر View / web.config این دو مورد و اضافه کنی

```
<add namespace="TwitterBootstrapMVC" />
<add namespace="TwitterBootstrap3" />
```

بعد از اون براحتی جواب میده
یادتون نره که فایل Portable.Licensing.dll هم هنگام اجرا باد در کنار دو فایل بالا وجود داشته باشد

نویسنده: مجتبی فخاری
تاریخ: ۱۳۹۲/۱۲/۲۳ ۱۷:۴۷

با سلام
آیا با Bootstrap.rtl3 هم جواب میده ؟
آخه من همین فایل شما را که برای mvc4 است دانلود کرده و add References هم نمودم و فضاهای نام را هم در View / web.config اضافه نمودم ولی اصلاً کار نمیده و برام Intelisense رو که میاره داخلش Bootstrap نداره.