

یک Model چیست؟

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

• همچنین با عنوان Business Logic نیز شناخته می‌شود.

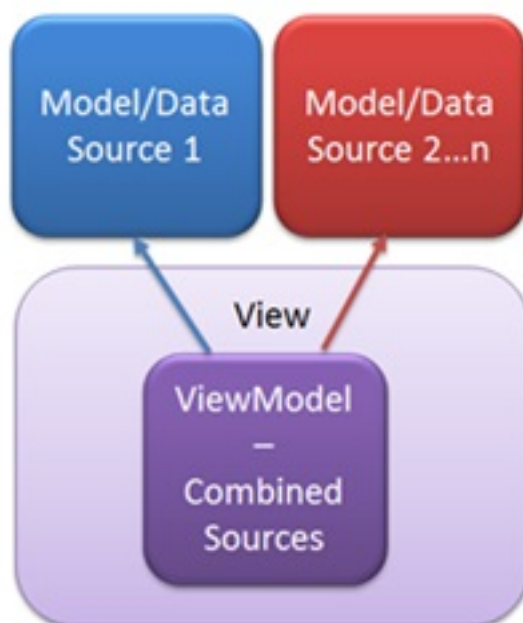
• Domain Logic داده‌هایی را که بین UI و دیتابیس پاس داده می‌شود، مدیریت می‌کند.

• 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.

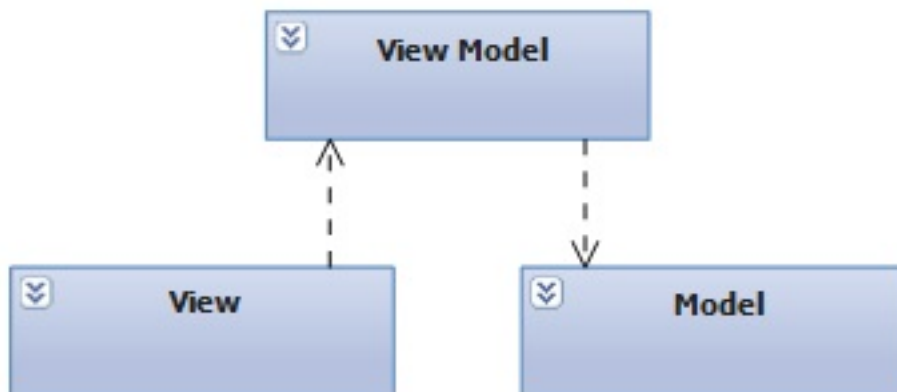
• بهینه شده برای استفاده و رندر توسط یک View



چرا باید از ViewModel استفاده کنیم؟

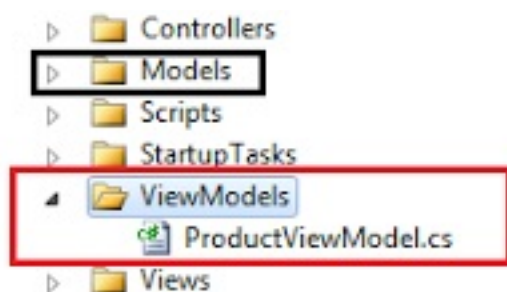
- اگر شما بخواهید چندین Model را به یک View پاس دهید، استفاده از ViewModel ایده بسیار خوبی است.
- همچنین امکان Validate کردن ViewModel را به روشی متفاوت‌تر از Domain Model برای سناریوهای اعتبارسنجی attribute-based را در اختیارمان قرار می‌دهد.
- برای قالب‌بندی داده‌ها نیز استفاده می‌شود.

- 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 ساده تر می کند.



مکان قرارگیری ViewModel در پروژه بهتر است کجا باشد؟

می تواند 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 ViewModel ها

داده هایی 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 پیچیده تر می شود، بهتر است از یک Mapper استفاده کنید.

اجازه دهید مثالی را در این رابطه براتون بیارم :

در یک View می خواهیم 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 :

Add View

View name:
Index

View engine:
Razor (CSHTML)

☒ Create a strongly-typed view

Model class:
SpeakerAndNewsViewModel (Project.Models)

Scaffold template:
Empty

☒ Reference script libraries

☐ Create as a partial view

☒ Use a layout or master page:

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

ContentPlaceHolder ID:
MainContent

Add Cancel

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 درستیست ؟
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: ۱۳۹۲/۰۱/۱۲ ۱۱:۴۰

«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: ۱۳۹۲/۰۱/۱۲ ۱۲:۲۳

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

نویسنده: یزدان
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۳:۸

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: ۱۳۹۲/۰۱/۱۲ ۱۳:۱۳

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: ۱۳۹۲/۰۱/۱۲ ۱۳:۴۱

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 بزنینم 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. این‌کار 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.Part3 = "فرم سابقه کاری";
    ViewBag.Part4 = "فرم دوره های تخصصی گذرانده";
    ViewBag.Part5 = "فرم تخصص های شغلی";
    ViewBag.Part6 = "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>
    </tr>

```

```

        <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 کودوم 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 می‌گردد. 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 اینباره 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 مازول) 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 « [سفرشی 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؟