

زمانی که درخواستی به سمت یک Action پارامتر دار ارسال میشود، قسمت ActionInvoker قبل از فراخوانی اکشن مربوطه، به دنبال Model Binder مناسبی برای داده‌های پارامترها می‌گردد و در صورت یافت نشدن، از ModelBinder پیش فرض ASP.NET MVC استفاده می‌کند.

### اما وظیفه‌ی ModelBinder چیست ؟

ModelBinder داده‌های ارسال شده از مرورگر را که توسط درخواست‌های HTTP (کوئری استرینگ‌ها و یا داده‌های همراه با فرم‌ها) ارسال شده است، تبدیل به داده‌های قابل فهم برای پارامترها میکند. به عبارتی ModelBinder وظیفه تبدیل داده‌های ارسال شده از سمت مرورگر به اشیاء .NET را دارد. فرض کنید ما مدلی به شکل زیر داریم :

```
public class CustomerInfo
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public DateTime BirthDate { get; set; }
}
```

فیلد آخر برای ذخیره‌ی تاریخ تولد مشتری استفاده میشود. که View مربوط به آن به شکل زیر خواهد بود :

CustomerInfo

FirstName

LastName

BirthDate

همانطور که می‌بینید تایپ کردن تاریخ به این صورت (AM 12:00:00 2009/1/1) ، هم زیاد جالب نیست و هم کمی مشکل است. به همین دلیل برخی سایت‌ها از سه قسمت جدا برای گرفتن روز ، ماه و سال استفاده می‌کنند و در نهایت آنها را با یکدیگر ترکیب میکنند.

در این مثال ما نیز می‌خواهیم تاریخ را به صورت زیر دریافت و پس از تبدیل آن به تاریخ میلادی، آن را به کاربر نمایش دهیم :

تاریخ تولد

1368

/

مرداد

/

11

اما هنگام ارسال فرم به صورت بالا ، ModelBinder توانایی تبدیل این سه ورودی (روز ، ماه و سال) به فیلد BirthDate موجود در کلاس CustomerInfo را ندارد. به همین خاطر ما باید یک ModelBinder متناسب با نیاز خود را طراحی کنیم. برای ایجاد یک ModelBinder سفارشی نیاز است که از کلاس IModelBinder ارثبری و متد BindModel آن را پیاده سازی کنیم. ساختار این اینترفیس به شکل زیر است :

```
public interface IModelBinder
{
    object BindModel(ControllerContext controllerContext,
        ModelBindingContext bindingContext);
}
```

متد BindModel حاوی 2 پارامتر است :

ControllerContext : حاوی اطلاعاتی در مورد درخواست http جاری

ModelBindingContext : این کلاس حاوی یک property به نام Model است که حاوی ارجاعی به مدلی که همکنون قصد پردازش آن را دارد.

با توجه به موارد بالا کلاس ما به شکل زیر خواهد بود :

```
using System;
using System.Web;
using System.Web.Mvc;
using ModelBinderExample.Models;
using Persia;

namespace ModelBinderExample.CustomModelBinder
{
    // Article written for www.dotnettips.info
    public class CustomerInfoModelBinder : IModelBinder
    {
        public object BindModel(ControllerContext controllerContext, ModelBindingContext bindingContext)
        {
            HttpRequestBase request = controllerContext.HttpContext.Request;

            string firstName = request.Form.Get("FirstName");
            string lastName = request.Form.Get("LastName");

            DateTime birthDate = this.GetMiladiDate(request);

            return new CustomerInfo()
            {
                FirstName = firstName,
                LastName = lastName,
                BirthDate = birthDate
            };
        }

        private DateTime GetMiladiDate(HttpRequestBase request)
        {
            int day = int.Parse(request.Form.Get("Day"));
            int month = int.Parse(request.Form.Get("Month"));
            int years = int.Parse(request.Form.Get("Years"));

            //Convert shamsi to miladi
            return Persia.Calendar.ConvertToGregorian(years, month, day, DateType.Gerigorian);
        }
    }
}
```

در کد بالا ابتدا موارد ارسال شده را دریافت میکنیم و توسط متد GetMiladiDate() تاریخ دریافتی از کاربر که به صورت روز، ماه

و سال میباید را تبدیل به میلادی میکنیم و سپس در قالب یک شی customerInfo آنها را برگشت می‌دهیم.  
 نکته : جهت تبدیل تاریخ شمسی به میلادی از کتابخانه‌ی Persia کمک گرفته شده است که در فایل پیوستی قرار داده شده.  
 کار ایجاد یک ModelBinder سفارشی تمام شده و حال نیاز است کلاس را در فایل Global.asax در قسمت Application\_start() ثبت کنیم به شکل زیر :

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

    WebApiConfig.Register(GlobalConfiguration.Configuration);
    FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
    RouteConfig.RegisterRoutes(RouteTable.Routes);
    //Register New ModelBinder
    ModelBinders.Binders.Add(typeof(CustomerInfo), new CustomerInfoModelBinder());
}
```

و برای استفاده از این ModelBinder ، ما باید به کنترلر اطلاع دهیم که میخواهیم از چه نوع Binding استفاده کنیم به همین دلیل از attribute زیر برای انجام این کار استفاده می‌کنیم:

```
[HttpPost]
public ActionResult Create([ModelBinder(typeof (CustomerInfoModelBinder))] CustomerInfo customerInfo)
{
    if (ModelState.IsValid)
    {
        ViewBag.FirstName = customerInfo.FirstName;
        ViewBag.LastName = customerInfo.LastName;
        ViewBag.BirthDate = customerInfo.BirthDate;
    }
    return View();
}
```

پروژه پیوستی : [ModelBinder-Example.zip](#)

## نظرات خوانندگان

نویسنده:

همتی

تاریخ:

۱۷:۱۴ ۱۳۹۲/۰۴/۲۰

ضمن تشکر از مقاله خوبتان

در قسمت آخر فرمودید ، باید به کنترلر اطلاع دهیم که میخواهیم از چه نوع Binding استفاده کنیم ولی اگر اشتباه نکنم این مورد اجباری نیست یعنی اکشن ما می‌تواند به شکل زیر هم باشد

```
public ActionResult Create(CustomerInfo customerInfo)
```

نویسنده:

ناصر طاهری

تاریخ:

۲۰:۱۷ ۱۳۹۲/۰۴/۲۰

ممنون

بله. در صورتی که ModelBinder رو در Application\_start() رجیستر کنیم نیازی به ذکر صریح در اکشن نیست. در غیر این صورت نیاز است.

```
ModelBinders.Binders.Add(typeof(CustomerInfo), new CustomerInfoModelBinder());
```

در صورتی که ModelBinder را رجیستر نکنیم و مستقیماً نوع Binding را در اکشن مشخص کنیم ، این امکان را خواهیم داشت که برای یک Model چندین ModelBinder مختلف ایجاد و استفاده کنیم.

نویسنده:

سام ناصری

تاریخ:

۸:۵۷ ۱۳۹۲/۰۵/۰۸

خب، برای خاطر خوانندگانی که دچار اشتباه میشوند بهتر نمیدونید که مقاله رو ویرایش کنید و این مطلب رو اصلاح کنید؟

نویسنده:

سام ناصری

تاریخ:

۹:۳ ۱۳۹۲/۰۵/۰۸

من میخواهم که مدل بایندر را برای یک پراپرتی از مدلم تنظیم کنم. یعنی پراپرتی از جنس byte[] داریم که میخواهم توسط FileModelBinder بایند شود. FileModelBinder فایل‌های آپلود شده به سرور را میخواند و در پراپرتی میریزد. من نمیخواهم همه‌ی پراپرتی‌های از جنس byte[] را به این بایندر واگذار کنم فقط میخواهم در مدلهای معینی در پراپرتی‌های مورد نظر این بایندر اعمال شود. همچنین من نمیخواهم که از بایندر در پارامترهای ورودی action method بر روی یک پارامتر byte[] استفاده کنم.

آیا اطلاق یک بایندر به یک پراپرتی از یک مدل برای یک تایپ مشخص مانند byte[] قابل انجام هست؟

نویسنده:

وحید نصیری

تاریخ:

۹:۳۴ ۱۳۹۲/۰۵/۰۸

بله. یک کلاس مشتق شده از System.Web.Mvc.DefaultModelBinder ایجاد کنید. در آن متد BindProperty را override کرده و سپس اگر خاصیتی، مثلاً خاصیت X بود (بر اساس مثلاً یک ویژگی خاص که به آن انتساب داده شده)، آنگاه از Model binder سفارشی استفاده گردد.

یک نمونه پیاده سازی کامل آن:

[MVC Property Binder](#)

نویسنده:

محسن خان

تاریخ:

۹:۳۸ ۱۳۹۲/۰۵/۰۸

اشتباه نیست، فقط کمی verbose است.

نویسنده: نازنین حسینی  
تاریخ: ۱۰:۴۷ ۱۳۹۳/۱۰/۰۱

سلام؛ با تشکر از مقاله شما. میخوامم بپرسم override کردن BindModel یا BindProperty برای زمانی که ما به تمام دیتا هامون دسترسی داریم حالا شکل برگرداندنمون فرق میکنه؟ اگر اینطوره سوالم اینه که برای حالتی که مدل ما به شکل زیر هست چگونه Items را Bind کنم چون از هر روشی میرم null هست!

```
public class Model
{
    public Model()
    {
        Items = new List<ItemModel>();
    }
    public Guid Id { get; set; }
    public Guid ProductId { get; set; }
    public List<ItemModel> Items { get; set; }
}

public class ItemModel
{
    public Guid Id
    public string Title { get; set; }
    public int Value { get; set; }
}
```

و من در view مدل زیر را احتیاج دارم.

```
@model List<Model>
```

در اکشن HttpPost مربوط به این مدل ItemsProperty Is Null.

نویسنده: وحید نصیری  
تاریخ: ۱۱:۰۰ ۱۳۹۳/۱۰/۰۱

- مراجعه کنید به نکته مطرح شده در مطلب «[ساخت یک Form Generator ساده در MVC](#)» زمانیکه قرار است یک آرایه از عناصر از کاربر دریافت شود. قسمت‌های «ویوی نمایش فرم تولید شده برای کاربر نهایی» و ShowForm آن برای دریافت اطلاعات از کاربر، دقیقاً یک لیست از شیء Value را دریافت می‌کنند.
- توضیحات تکمیلی آن در اینجا «[ASP.NET Wire Format for Model Binding to Arrays, Lists, Collections, Dictionaries](#)»