

ابتدا در پروژه‌ی mvc خود یک پوشه با نامی دلخواه (مثلا MyHelpers) بسازید و سپس کلاسی با محتویات زیر را به آن اضافه کنید(نام کلاس به دلخواه Helpers گذاشته شده است) :

```
public static class Helpers
{
    // اینجا متدهای کمکی قرار میگیرند
}
```

1- تبدیل تاریخ میلادی به شمسی با استفاده از کتابخانه ی [Persia](#) :

```
public static MvcHtmlString FarsiDate(this HtmlHelper html, DateTime dateTime)
{
    var tag = new TagBuilder("span");
    tag.MergeAttribute("dir", "ltr");
    tag.AddCssClass("farsi-date");
    tag.SetInnerText(Calendar.ConvertToPersian(dateTime).ToString("W"));
    return MvcHtmlString.Create(tag.ToString(TagRenderMode.Normal));
}
```

مثال استفاده :

```
@Html.FarsiDate(news.DateTimeCreated)
```

2- زمان فارسی :

```
public static MvcHtmlString FarsiTime(this HtmlHelper html, DateTime dateTime)
{
    var tag = new TagBuilder("span");
    tag.MergeAttribute("dir", "ltr");
    tag.AddCssClass("farsi-time");
    tag.SetInnerText(Calendar.ConvertToPersian(dateTime).ToString("R"));
    return MvcHtmlString.Create(tag.ToString(TagRenderMode.Normal));
}
```

مثال استفاده :

```
@Html.FarsiTime(news.DateTimeCreated)
```

3- تاریخ و زمان فارسی :

```
public static MvcHtmlString FarsiDateAndTime(this HtmlHelper html, DateTime dateTime)
{
    return MvcHtmlString.Create(FarsiTime(html, dateTime).ToHtmlString() + " ، " +
    FarsiDate(html, dateTime).ToHtmlString());
}
```

مثال استفاده :

```
@Html.FarsiDateAndTime(news.DateTimeCreated)
```

4- زمان گذشته :

```
public static MvcHtmlString FarsiRemaining(this HtmlHelper html, DateTime dateTime)
{
    var tag = new TagBuilder("span");
    tag.MergeAttribute("dir", "rtl");
    tag.AddCssClass("farsi-remaining");
    tag.SetInnerText(Calendar.ConvertToPersian(dateTime).ToRelativeDateString("TY"));
    return MvcHtmlString.Create(tag.ToString(TagRenderMode.Normal));
}
```

مثال استفاده :

```
@Html.FarsiRemaining(news.DateTimeCreated)
```

5- خلاصه‌ی مطلب با استفاده از کتابخانه ی [Html Agility Pack](#) (تعداد کلمات از کلمه‌ی اول یک متن به اندازه‌ی max)

```
public static string GetSummary(this HtmlHelper html, string text, int max)
{
    string summaryHtml = string.Empty;
    // load our html document
    var htmlDoc = new HtmlDocument();
    htmlDoc.LoadHtml(text);
    int wordCount = 0;
    foreach (var element in htmlDoc.DocumentNode.ChildNodes)
    {
        // inner text will strip out all html, and give us plain text
        string elementText = element.InnerText;

        // we split by space to get all the words in this element
        string[] elementWords = elementText.Split(new char[] { ' ' });

        // and if we haven't used too many words ...
        if (wordCount <= max)
        {
            // add the *outer* HTML (which will have proper
            // html formatting for this fragment) to the summary
            summaryHtml += element.OuterHtml;
            wordCount += elementWords.Count() + 1;
        }
        else
        {
            break;
        }
    }
    return summaryHtml;
}
```

مثال استفاده :

```
@Html.Raw(Html.GetSummary(news.Content, 60))
```

6- گرفتن لیست Validation Error ها در ModelState :

```
public static List<string> GetListOfErrors(this ModelStateDictionary modelState)
{
    var list = modelState.ToList();
    var listErrors = new List<string>();
    foreach (var keyValuePair in list)
    {
        listErrors.AddRange(keyValuePair.Value.Errors.Select(error => error.ErrorMessage));
    }
    return listErrors;
}
```

مثال استفاده (در کنترلر):

```
var listErrors = ModelState.GetListOfErrors();
```

از دوستان عزیز خواهمشدم متدهای کمکی مورد استفاده در پروژه های خود را در قسمت نظرات قرار دهند.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۶:۳۹ ۱۳۹۱/۰۴/۰۹

ممنون. مطلب مفیدی بود. برای حالت GetSummary کاری که من انجام می‌دم، حذف تگ‌های html است و بعد بررسی کلمات آن. به این ترتیب نیازی به کتابخانه‌های جانبی برای بررسی کلمات موجود در html نخواهد بود؛ چون هدف اصلی، متن‌های واقعی در دسترس است منهای تگ‌های موجود:

```
public static string RemoveAllHtmlTags(string text)
{
    return string.IsNullOrEmpty(text) ?
        string.Empty :
        Regex.Replace(text, @"<(.|\n)*?>", string.Empty);
}
```

نویسنده: شهروز جعفری
تاریخ: ۱۹:۴۵ ۱۳۹۱/۰۴/۱۱

من هم برای seo friendly کردن URL از کد زیر استفاده میکنم

```
public static class StringHelpers
{
    public static string ToSeoUrl(this string url)
    {
        // make the url lowercase
        string encodedUrl = (url ?? "").ToLower();

        // replace & with and
        encodedUrl = Regex.Replace(encodedUrl, @"\&+", "and");

        // remove characters
        encodedUrl = encodedUrl.Replace("'", "");

        // remove invalid characters
        encodedUrl = Regex.Replace(encodedUrl, @"^[a-z0-9-\u0600-\u06FF]", "-");

        // remove duplicates
        encodedUrl = Regex.Replace(encodedUrl, @"-+", "-");

        // trim leading & trailing characters
        encodedUrl = encodedUrl.Trim('-');

        return encodedUrl;
    }
}
```

نویسنده: naser
تاریخ: ۲۲:۲۷ ۱۳۹۱/۰۴/۱۷

ببخشید این کتابخانه [Html Agility Pack](#) که معرفی کردید آیا امکان انتخاب تمامی radio هایی که تیک خورده رو میده. در حالت عادی با دستور زیر همه رو برمیگردونه

```
var radio = doc.DocumentNode.SelectNodes("//input[@class='radio']");
```

اما من میخام فقط radio هایی رو برگردونه که تیک خورده. ممنون میشم راهنمایی کنید؟

نویسنده: وحید نصیری

تاریخ: ۲۳:۱۱ ۱۳۹۱/۰۴/۱۷

بهتره این مباحث رو در انجمن یا mailing list خودش بررسی کنید.
زمانیکه شما نودهای radio رو دارید بقیه‌اش چیزی شبیه به این خواهد بود

```
string isChecked = element.GetAttributeValue("checked", "unchecked");
```

نویسنده: میثم شریفی
تاریخ: ۲۰:۱۱ ۱۳۹۱/۰۴/۲۵

```
public static class HtmlHelperExtensions
{
    private const string Nbsp = "&nbsp;";
    private const string SelectedAttribute = " selected='selected'";

    public static MvcHtmlString NbspIfEmpty(this HtmlHelper helper, string value)
    {
        return new MvcHtmlString(string.IsNullOrEmpty(value) ? Nbsp : value);
    }

    public static MvcHtmlString SelectedIfMatch(this HtmlHelper helper, object expected, object actual)
    {
        return new MvcHtmlString(Equals(expected, actual) ? SelectedAttribute : string.Empty);
    }
}
```

و یک مثال جهت استفاده

```
<select>
    @foreach (var item in ViewBag.Items)
    {
        <option@Html.SelectedIfMatch((string)ViewBag.SelectedItem,

(string)item.ItemName)>@item.ItemName</option>
    }
</select>
```

نویسنده: hasani
تاریخ: ۲۱:۵۴ ۱۳۹۱/۰۵/۱۶

با تشکر از مبحث مهمی که ارائه کردید.

در صورتی که جهت فراخوانی اطلاعات از JQuery Ajax استفاده شود، چگونه میتوان از Helper تاریخ شمسی استفاده کرد؟
به بیان دیگر وقتی توسط Ajax اطلاعات از سمت سرور دریافت میشود، اطلاعات بازگشتی توسط متغیری مثل Result که از نوع Json است برگشت داده میشود. که در سمت کلاینت نمیتوان از (@Html.FarsiDate(news.DateTimeCreated) استفاده کرد.
در سمت سرور هم فرض کنید از کد زیر استفاده شود،

```
var data = context.Tabell1.Select(
    p =>
    new
    {
        p.Date1,
        p.Comment,
        p.Cost,
        p.UserId,
    }).FirstOrDefault(p.UserId == UserId);
```

data.Date1 در سمت سرور قابل تغییر نیست و با خطای Readonly متوقف میشود.
ممنون میشوم راهنمایی بفرمائید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۶ ۲۲:۴۹

خروجی json در یک View یعنی متغیری از نوع جاوا اسکریپت. این متغیر به صورت مستقیم در کدهای یک htmlHelper قابل استفاده نیست (چون از نوع دات نت و سی شارپ یا وی بی است).
در اینجا تبدیل تاریخ مدنظر را باید همان طرف کدهای کنترلر انجام داد. یا این طرف از توابع جاوا اسکریپتی تبدیل تاریخ میلادی به شمسی باید استفاده کرد. دو context مختلف را نمی‌شود در آن واحد با هم ترکیب کرد.
البته می‌شود داخل کدهای جاوا اسکریپتی یک View از کدهای مثلا Razor استفاده کرد اما این کدها در زمان اجرا به صورت جاوا اسکریپتی پردازش و رندر می‌شوند و نه به فرمت مثلا سی شارپ. [برای نمونه](#) می‌شود در یک View داخل کدهای JavaScript موجود از Url.Action استفاده کرد. این مورد در زمان اجرا فقط تبدیل به یک رشته JavaScript می‌شود. البته این نکته برای بکارگیری Html.FarsiDateAndTime نیز صدق می‌کند و فراخوانی آن باید داخل کدهای Ajax جاوا اسکریپتی یک View باشد (و نه خارج از آن چون context دیگر JavaScript نخواهد بود). همچنین باید در نظر داشت که Html.FarsiDateAndTime در زمان رندر شدن جاوا اسکریپت موجود در یک View، تبدیل به معادل رشته‌ای آن شده و جایگزین می‌شود. یعنی در قسمت نتیجه یک عملیات Ajax قابل استفاده نخواهد بود.

نویسنده: hasani
تاریخ: ۱۳۹۱/۰۵/۱۶ ۲۳:۱۴

ممنون بابت پاسخ سریع و کامل شما.
اگر «تبدیل تاریخ همان طرف کدهای کنترلر» مد نظر باشد امکان دارد راهنمایی بفرمائید؟ اگر به شکل کدهای زیر عمل کنم طبیعی است که خطا میدهد

راه بهتری هست که قبل از Return عمل تبدیل تاریخ را انجام دهم و سپس data را پس
JsonRequestBehavior.AllowGet, را انجام دهم.

ممنون

```
var data = context.Tabell1.Select(
    p =>
    new
    {
        Date1= Persia.Calendar.ConvertToPersian(p.Date1),
        p.Comment,
        p.Cost,
        p.UserId,
    }).FirstOrDefault(p.UserId == UserId);
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۶ ۲۳:۲۵

چه خطایی می‌ده؟ چون من الان استفاده کردم و خطایی نداد.
اگر یک چنین خطایی دریافت کردید:

The conversion of a varchar data type to a datetime data type resulted in an out-of-range value.

مربوط است به نال بودن یکی از مقادیر تاریخ ثبت شده در بانک اطلاعاتی. خاصیت را در سمت کدهای خود nullable تعریف کنید.

نویسنده: hasani

تاریخ: ۲۳:۴۱ ۱۳۹۱/۰۵/۱۶

خطای زیر را میدهد:

LINQ to Entities does not recognize the method 'Persia.SolarDate ConvertToPersian(System.DateTime)' method, and this method cannot be translated into a store expression.

برای ساخت مدل

من اول به روش دیتابیس فرست EDMX را ساختم و سپس از ADO.NET DbContext Generator استفاده کردم.

نویسنده: وحید نصیری

تاریخ: ۲۳:۵۱ ۱۳۹۱/۰۵/۱۶

- من برای تست از EF 4.3.1.0 استفاده کردم و به نظر این محدودیت رو حداقل با select و projection ندارد و برنامه بدون مشکل اجرا شد.

- برای نگارش های قبل از آن این روش جواب می دهد:

```
db.table.Where(...)  
    .Select(p =>  
        new  
        {  
            p.Date1,  
            p.Comment,  
            p.Cost,  
            p.UserId,  
        })  
    .AsEnumerable() // مهم  
    .Select(... استفاده کنید)  
    .ToList();
```

نویسنده: hasani

تاریخ: ۰:۳۵ ۱۳۹۱/۰۵/۱۷

WOW Wonderful

مثل همیشه عالی بود.

حل شد.

نویسنده: احمدعلی شفیعی

تاریخ: ۱۱:۳۹ ۱۳۹۳/۰۸/۰۱

برای تبدیل تاریخ میلادی به شمسی استفاده از [System.Globalization.PersianCalendar](#) راه حل مناسبی نیست؟

نویسنده: محسن خان

تاریخ: ۱۲:۳۱ ۱۳۹۳/۰۸/۰۵

تمام این ها در پشت صحنه از همین فضای نام استفاده می کنند. فقط کمی ساده ترش کردن.