عنوان: **اعتبارسنجی سایتهای چند زبانه در ASP.NET MVC - قسمت اول**
نویسنده: محسن اسماعیل پور
تاریخ: ۱۳۹۲/۰۳/۰۸ ۰:۵
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net MVC, jQuery, Validation, Localization, Globalization, MVC

اگر در حال تهیه یک سایت چند زبانه هستید و همچنین سری مقالات Globalization در ASP.NET MVC رو دنبال کرده باشید میدانید که با تغییر Culture فایلهای Resource مورد نظر بارگذاری و نوشته‌های سایت تغییر میابند ولی با تغییر Culture رفتار اعتبارسنجی در سمت سرور نیز تغییر و اعتبارسنجی بر اساس Culture فعلی سایت انجام میگیرد. بررسی این موضوع را با یک مثال شروع میکنیم.

یک پروژه وب بسازید سپس به پوشه Models یک کلاس با نام ValueModel اضافه کنید. تعریف کلاس به شکل زیر هست:

```
public class ValueModel
{
    [Required]
    [Display(Name = "Decimal Value")]
    public decimal DecimalValue { get; set; }

    [Required]
    [Display(Name = "Double Value")]
    public double DoubleValue { get; set; }

    [Required]
    [Display(Name = "Integer Value")]
    public int IntegerValue { get; set; }

    [Required]
    [Display(Name = "Date Value")]
    public DateTime DateValue { get; set; }
}
```

به سراغ کلاس HomeController بروید و کدهای زیر را اضافه کنید:

```
[HttpPost]
public ActionResult Index(ValueModel valueModel)
{
    if (ModelState.IsValid)
    {
        return Redirect("Index");
    }

    return View(valueModel);
}
```

Culture را به fa-IR تغییر میدهیم، برای اینکار در فایل web.config در بخش system.web کد زیر اضافه نمایید:

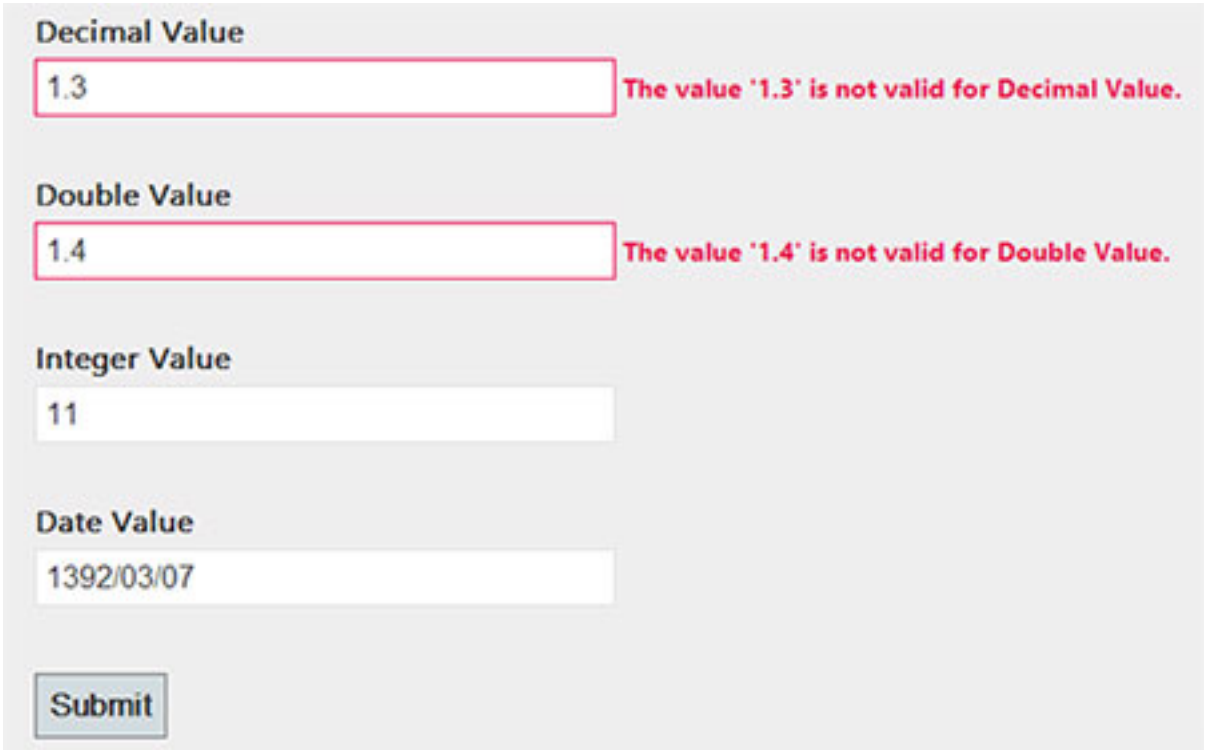
```
<globalization culture="fa-IR" uiCulture="fa-IR" />
```

و در نهایت به سراغ فایل Index.cshtml بروید کدهای زیر رو اضافه کنید:

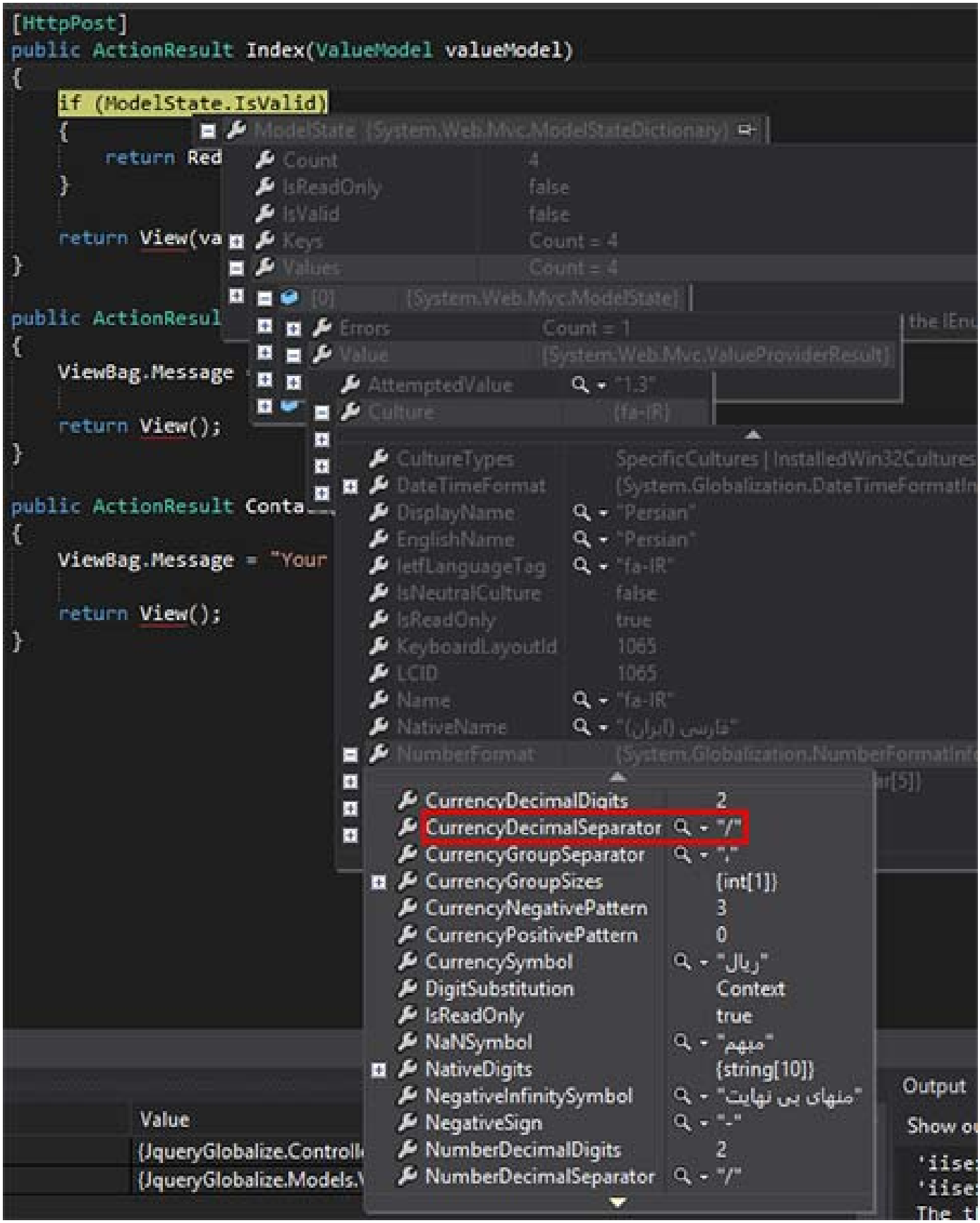
```
@using (Html.BeginForm())
{
    <ol>
        <li>
            @Html.LabelFor(m => m.DecimalValue)
            @Html.TextBoxFor(m => m.DecimalValue)
            @Html.ValidationMessageFor(m => m.DecimalValue)
        </li>
```

```
        <li>
            @Html.LabelFor(m => m.DoubleValue)
            @Html.TextBoxFor(m => m.DoubleValue)
            @Html.ValidationMessageFor(m => m.DoubleValue)
        </li>
        <li>
            @Html.LabelFor(m => m.IntegerValue)
            @Html.TextBoxFor(m => m.IntegerValue)
            @Html.ValidationMessageFor(m => m.IntegerValue)
        </li>
        <li>
            @Html.LabelFor(m => m.DateValue)
            @Html.TextBoxFor(m => m.DateValue)
            @Html.ValidationMessageFor(m => m.DateValue)
        </li>
        <li>
            <input type="submit" value="Submit"/>
        </li>
    </ol>
}
```

پرژه را اجرا نمایید و در ۲ تکست باکس اول ۲ عدد اعشاری را و در ۲ تکست باکس آخر یک عدد صحیح و یک تاریخ وارد نمایید و سپس دکمه Submit را بزنید. پس از بازگشت صفحه از سمت سرور در در ۲ تکست باکس اول با این پیامها روبرو میشوید که مقادیر وارد شده نامعتبر میباشند.



اگر پروژه رو در حالت دیباگ اجرا کنیم و نگاهی به داخل ModelState بیاندازیم، میبینیم که کاراکتر جدا کننده قسمت اعشاری برای fa-IR '/' میباشد که در اینجا برای اعداد مورد نظر کاراکتر '.' وارد شده است.



برای فایق شدن بر این مشکل یا باید سمت سرور اقدام کرد یا در سمت کلاینت. در بخش اول راه حل سمت کلاینت را بررسی مینماییم.

در سمت کلاینت برای اینکه کاربر را مجبور به وارد کردن کاراکترهای مربوط به Culture فعلی سایت نماییم باید مقادیر وارد شده را اعتبارسنجی و در صورت معتبر نبودن مقادیر پیام مناسب نشان داده شود. برای اینکار از کتابخانه jQuery Globalize استفاده میکنیم. برای اضافه کردن jQuery Globalize از طریق کنسول nuget فرمان زیر اجرا نمایید:

```
PM> Install-Package jquery-globalize
```

پس از نصب کتابخانه اگر به پوشه Scripts نگاهی بیاندازید میبینید که پوشای با نام jquery.globalize اضافه شده است. درداخل پوشه زیر پوشی دیگری با نام cultures وجود دارد که در آن Culture های مختلف وجود دارد و بسته به نیاز میتوان از آنها استفاده کرد. دوباره به سراغ فایل Index.cshtm بروید و فایلهای جاوا اسکریپتی زیر را به صفحه اضافه کنید:

```
<script src="~/Scripts/jquery.validate.js"> </script>
<script src="~/Scripts/jquery.validate.unobtrusive.js"> </script>
<script src="~/Scripts/jquery.globalize/globalize.js"> </script>
<script src="~/Scripts/jquery.globalize/cultures/globalize.culture.fa-IR.js"> </script>
```

در فایل globalize.culture.fa-IR.js کاراکتر جدا کننده اعشاری '.' در نظر گرفته شده است که مجبور به تغییر آن هسیتم. برای اینکار فایل را باز کرده و numberFormat را پیدا کنید و آن را به شکل زیر تغییر دهید:

```
numberFormat: {
    pattern: ["n-"],
    ".": "/",
    currency: {
        pattern: ["$n-", "$ n"],
        ".": "/",
        symbol: "ریال"
    }
},
```

و در نهایت کدهای زیر را به فایل Index.cshtml اضافه کنید و برنامه را دوباره اجرا نمایید :

```
Globalize.culture('fa-IR');
$.validator.methods.number = function(value, element) {
    if (value.indexOf('.') > 0) {
        return false;
    }
    var splitedValue = value.split('/');
    if (splitedValue.length === 1) {
        return !isNaN(Globalize.parseInt(value));
    } else if (splitedValue.length === 2 && $.trim(splitedValue[1]).length === 0) {
        return false;
    }
    return !isNaN(Globalize.parseFloat(value));
};
};
```

در خط اول Culture را ست مینمایم و در ادامه نحوه اعتبارسنجی را در unobtrusive validation تغییر میدهیم . از آنجایی که برای اعتبارسنجی عدد وارد شده از تابع parseFloat استفاده میشود، کاراکتر جدا کننده قسمت اعشاری قابل قبول برای این تابع '.' است پس در داخل تابع دوباره '/' به '.' تبدیل میشود و سپس اعتبارسنجی انجام میشود از اینرو اگر کاربر '.' را نیز وارد نماید قابل قبول است به همین دلیل با این خط کد if (value.indexOf('.') > 0) وجود نقطه را بررسی میکنیم تا در صورت وجود '.' پیغام خطا نشان داده شود.در خط بعدی بررسی مینماییم که اگر عدد وارد شده اعشاری نباشد از تابع parseInt استفاده نماییم. در خط بعدی این حالت را بررسی مینماییم که اگر کاربر عددی همچون ۱۲/ وارد کرد پیغام خطا صادر شود.

برای اعتبارسنجی تاریخ شمسی متاسفانه توابع کمکی برای تبدیل تاریخ در فایل globalize.culture.fa-IR.js وجود ندارد ولی اگر نگاهی به فایلهای Culture عربی بیاندازید همه دارای توابع کمکی برای تبدیل تاریخ هجری به میلادی هستند به همین دلیل امکان اعتبارسنجی تاریخ شمسی با استفاده از jQuery Globalize میسر نمیباشد. من خودم تعدادی توابع کمکی را به globalize.culture.fa-IR.js اضافه کردام که از تقویم فارسی آقای علی فرهادی برداشت شده است و با آنها کار اعتبارسنجی را انجام میدهیم. لازم به ذکر است این روش ۱۰۰% تست نشده است و شاید راه کاملا اصولی نباشد ولی به هر حال در اینجا توضیح میدهم. در فایل globalize.culture.fa-IR.js قسمت Gregorian_Localized را پیدا کنید و آن را با کدهای زیر جایگزین کنید:

```
Gregorian_Localized: {
    firstDay: 6,
    days: {
        names: ["یکشنبه", "دوشنبه", "سه شنبه", "چهارشنبه", "پنجشنبه", "جمعه", "شنبه"],
        namesAbbr: ["یکشنبه", "دوشنبه", "سه شنبه", "چهارشنبه", "پنجشنبه", "جمعه", "شنبه"],
        namesShort: ["ی", "د", "س", "چ", "پ", "ج", "ش"]
    },
    months: {
        names: ["ژانویه", "فوریه", "مارس", "آوریل", "می", "ژوئن", "ژوئیه", "اوت", "سپتامبر", "اکتبر", "نوامبر", "دسامبر", ""],
        namesAbbr: ["ژانویه", "فوریه", "مارس", "آوریل", "می", "ژوئن", "ژوئیه", "اوت", "سپتامبر", "اکتبر", "نوامبر", "دسامبر", ""]
    },
    AM: ["ق.ظ", "ق.ظ", "ق.ظ"],
    PM: ["ب.ظ", "ب.ظ", "ب.ظ"],
    patterns: {
        d: "yyyy/MM/dd",
        D: "yyyy/MM/dd",
        t: "hh:mm tt",
        T: "hh:mm:ss tt",
        f: "yyyy/MM/dd hh:mm tt",
        F: "yyyy/MM/dd hh:mm:ss tt",
        M: "dd MMMM"
    },
    JalaliDate: {
        g_days_in_month: [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31],
        j_days_in_month: [31, 31, 31, 31, 31, 31, 30, 30, 30, 30, 30, 29]
    },
    gregorianToJalali: function (gY, gM, gD) {
        gY = parseInt(gY);
        gM = parseInt(gM);
        gD = parseInt(gD);
        var gy = gY - 1600;
        var gm = gM - 1;
        var gd = gD - 1;

        var gDayNo = 365 * gy + parseInt((gy + 3) / 4) - parseInt((gy + 99) / 100) + parseInt((gy + 399) / 400);

        for (var i = 0; i < gm; ++i)
            gDayNo += Globalize.culture().calendars.Gregorian_Localized.JalaliDate.g_days_in_month[i];
        if (gm > 1 && ((gy % 4 == 0 && gy % 100 != 0) || (gy % 400 == 0)))
            /* leap and after Feb */
            ++gDayNo;
        gDayNo += gd;

        var jDayNo = gDayNo - 79;

        var jNp = parseInt(jDayNo / 12053);
        jDayNo %= 12053;

        var jy = 979 + 33 * jNp + 4 * parseInt(jDayNo / 1461);

        jDayNo %= 1461;

        if (jDayNo >= 366) {
            jy += parseInt((jDayNo - 1) / 365);
            jDayNo = (jDayNo - 1) % 365;
        }

        for (var i = 0; i < 11 && jDayNo >= Globalize.culture().calendars.Gregorian_Localized.JalaliDate.j_days_in_month[i]; ++i) {
            jDayNo -= Globalize.culture().calendars.Gregorian_Localized.JalaliDate.j_days_in_month[i];
        }
        var jm = i + 1;
        var jd = jDayNo + 1;

        return [jy, jm, jd];
    },
    jalaliToGregorian: function (jY, jM, jD) {
```

```
        jY = parseInt(jY);
        jM = parseInt(jM);
        jD = parseInt(jD);
        var jy = jY - 979;
        var jm = jM - 1;
        var jd = jD - 1;

        var jDayNo = 365 * jy + parseInt(jy / 33) * 8 + parseInt((jy % 33 + 3) / 4);
        for (var i = 0; i < jm; ++i) jDayNo +=
Globalize.culture().calendars.Gregorian_Localized.JalaliDate.j_days_in_month[i];

        jDayNo += jd;

        var gDayNo = jDayNo + 79;

        var gy = 1600 + 400 * parseInt(gDayNo / 146097); /* 146097 = 365*400 + 400/4 - 400/100 +
400/400 */
        gDayNo = gDayNo % 146097;

        var leap = true;
        if (gDayNo >= 36525) /* 36525 = 365*100 + 100/4 */ {
            gDayNo--;
            gy += 100 * parseInt(gDayNo / 36524); /* 36524 = 365*100 + 100/4 - 100/100 */
            gDayNo = gDayNo % 36524;

            if (gDayNo >= 365)
                gDayNo++;
            else
                leap = false;
        }

        gy += 4 * parseInt(gDayNo / 1461); /* 1461 = 365*4 + 4/4 */
        gDayNo %= 1461;

        if (gDayNo >= 366) {
            leap = false;

            gDayNo--;
            gy += parseInt(gDayNo / 365);
            gDayNo = gDayNo % 365;
        }

        for (var i = 0; gDayNo >=
Globalize.culture().calendars.Gregorian_Localized.JalaliDate.g_days_in_month[i] + (i == 1 && leap) ;
i++)
            gDayNo -= Globalize.culture().calendars.Gregorian_Localized.JalaliDate.g_days_in_month[i] +
(i == 1 && leap);
        var gm = i + 1;
        var gd = gDayNo + 1;

        return [gy, gm, gd];
    },
    checkDate: function (jY, jM, jD) {
        return !(jY < 0 || jY > 32767 || jM < 1 || jM > 12 || jD < 1 || jD >
            (Globalize.culture().calendars.Gregorian_Localized.JalaliDate.j_days_in_month[jM - 1] + (jM
== 12 && !((jY - 979) % 33 % 4))));
    },
    convert: function (value, format) {
        var day, month, year;

        var formatParts = format.split('/');
        var dateParts = value.split('/');
        if (formatParts.length !== 3 || dateParts.length !== 3) {
            return false;
        }

        for (var j = 0; j < formatParts.length; j++) {
            var currentFormat = formatParts[j];
            var currentDate = dateParts[j];
            switch (currentFormat) {
                case 'dd':
                    if (currentDate.length === 2 || currentDate.length === 1) {
                        day = currentDate;
                    } else {
                        year = currentDate;
                    }
                    break;
                case 'MM':
                    month = currentDate;
                    break;
                case 'yyyy':
```

```
                    if (currentDate.length === 4) {
                        year = currentDate;
                    } else {
                        day = currentDate;
                    }
                    break;
                default:
                    return false;
            }
        }

        year = parseInt(year);
        month = parseInt(month);
        day = parseInt(day);
        var isValidDate = Globalize.culture().calendars.Gregorian_Localized.checkDate(year, month,
day);
        if (!isValidDate) {
            return false;
        }

        var grDate = Globalize.culture().calendars.Gregorian_Localized.jalaliToGregorian(year, month,
day);
        var shDate = Globalize.culture().calendars.Gregorian_Localized.gregorianToJalali(grDate[0],
grDate[1], grDate[2]);

        if (year === shDate[0] && month === shDate[1] && day === shDate[2]) {
            return true;
        }

        return false;
    }
},
```

روال کار در تابع convert به اینصورت است که ابتدا تاریخ وارد شده را بررسی مینماید تا معتبر بودن آن معلوم شود به عنوان مثال اگر تاریخی مثل 31/12/1392 وارد شده باشد و در ادامه برای بررسی بیشتر تاریخ یک بار به میلادی و تاریخ میلادی دوباره به شمسی تبدیل میشود و با تاریخ وارد شده مقایسه میشود و در صورت برابری تاریخ معتبر اعلام میشود. در فایل Index.cshtml کدهای زیر اضافی نمایید:

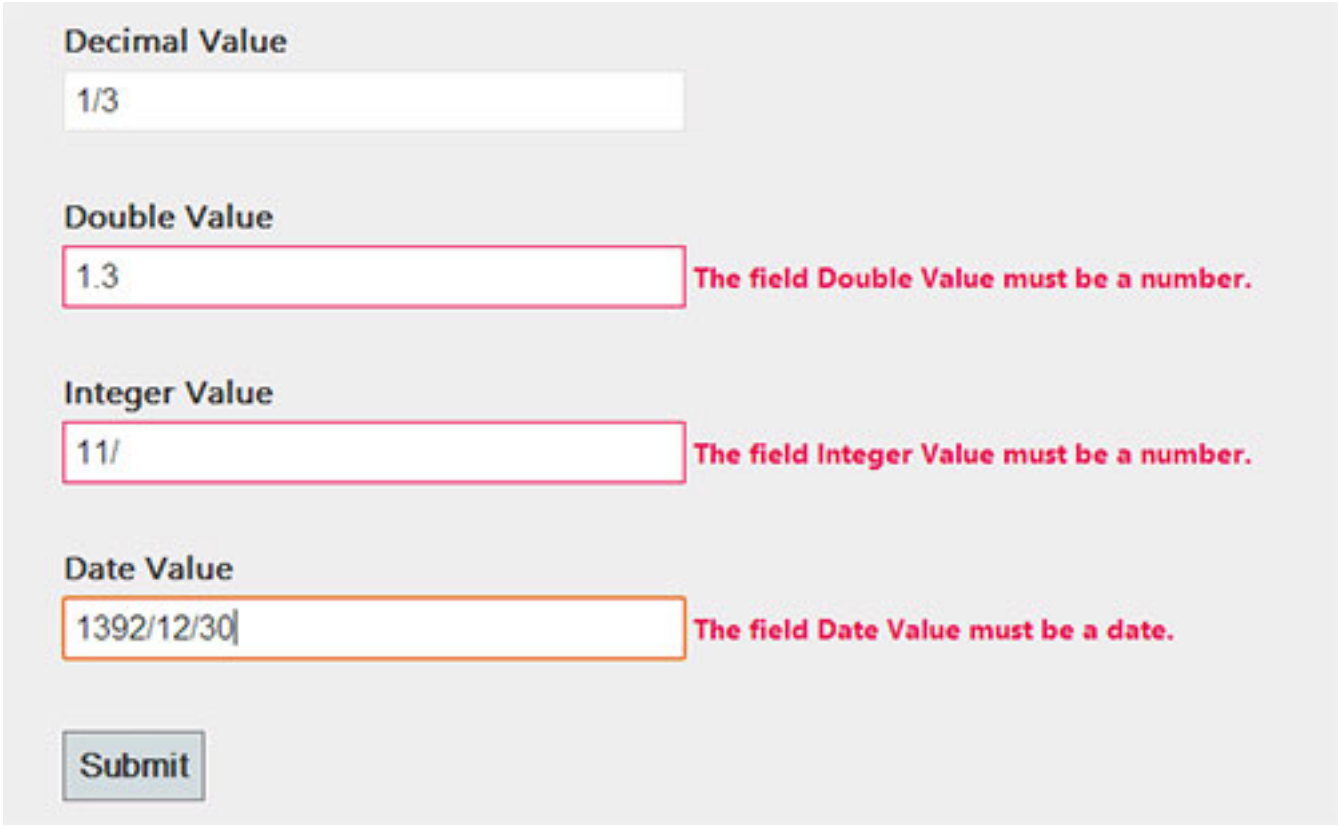
```
$.validator.methods.date = function (value, element) {
    return Globalize.culture().calendars.Gregorian_Localized.convert(value, 'yyyy/MM/dd');
};
```

برای اعتبارسنجی تاریخ میتوانید از ۲ فرمت استفاده کنید:

۱ - yyyy/MM/dd

۲ - dd/MM/yyyy

البته از توابع اعتبارسنجی تاریخ میتوانید به صورت جدا استفاده نمایید و لزومی ندارد آنها را همراه با jQuery Globalize بکار ببرید. در آخر خروجی کار به این شکل است:



در کل استفاده از jQuery Globalize برای اعتبارسنجی در سایتهای چند زبانه به نسبت خوب میباشد و برای هر زبان میتوانید از culture مورد نظر استفاده نمایید. در قسمت دوم این مطلب به بررسی بخش سمت سرور میپردازیم.