

روش پیش فرض اعتبارسنجی برنامه‌های ASP.NET MVC، استفاده از دو افزونه‌ی `jquery.validate` و `jquery.validate.unobtrusive` است.

```
<script src="~/Scripts/jquery.validate.min.js" type="text/javascript"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js" type="text/javascript"></script>
```

کار اصلی اعتبارسنجی، توسط افزونه‌ی `jquery.validate` انجام می‌شود و فایل `jquery.validate.unobtrusive` صرفاً یک وفق دهنده و مترجم ویژگی‌های خاص ASP.NET MVC به `jquery.validate` است.

عدم سازگاری پیش فرض `jquery.validate` با بعضی از ویجت‌های Kendo UI

در حالت استفاده از Kendo UI، این سیستم هنوز هم کار می‌کند؛ اما با یک مشکل. اگر برای مثال از `kendoComboBox` استفاده کنید، اعتبارسنجی‌های تعریف شده در برنامه، توسط `jquery.validate` دیده نخواهند شد. برای مثال فرض کنید یک چنین مدلی در اختیار View برنامه قرار گرفته است:

```
public class OrderDetailViewModel
{
    [StringLength(15)]
    [Required]
    public string Destination { get; set; }
}
```

با این View که در آن به فیلد `Destination`، یک `kendoComboBox` متصل شده است:

```
@model Mvc4TestViewModel.Models.OrderDetailViewModel

@using (Ajax.BeginForm(actionName: "Index", controllerName: "Home",
    ajaxOptions: new AjaxOptions(),
    htmlAttributes: new { id = "Form1", name = "Form1" }, routeValues: new { }
))
{
    @Html.AntiForgeryToken()
    @Html.ValidationSummary(true)

    <fieldset>
        <legend>OrderDetail</legend>
        <div class="editor-label">
            @Html.LabelFor(model => model.Destination)
        </div>
        <div class="editor-field">
            @Html.TextBoxFor(model => model.Destination, new { @class = "k-textbox" })
            @Html.ValidationMessageFor(model => model.Destination)
        </div>

        <p>
            <button class="k-button" type="submit" title="Submit">
                Submit
            </button>
        </p>
    </fieldset>
}

@section JavaScript
{
    <script type="text/javascript">
        $(function () {
            $("#Destination").kendoComboBox({
                dataSource: [
                    "loc 1",
                    "loc 2"
                ]
            });
        });
    </script>
}
```

```
});
</script>
}
```

اگر برنامه را اجرا کنید و بر روی دکمه‌ی submit کلیک نمائید، ویژگی Required عمل نخواهد کرد و عملاً در سمت کاربر اعتبارسنجی رخ نمی‌دهد.

```
<div class="editor-field">
  <span class="k-widget k-combobox k-header k-textbox" style="">
    <span tabindex="-1" class="k-dropdown-wrap k-state-default" unselectable="on">
      <input name="Destination_input" tabindex="0" class="k-input k-textbox" role="combobox" aria-
        busy="false" aria-disabled="false" aria-expanded="false" aria-readonly="false" aria-
        activedescendant="Destination_option_selected" aria-owns="Destination_listbox" style="width: 100%;"
        aria-autocomplete="list" type="text" autocomplete="off"></input>
      <span tabindex="-1" class="k-select" unselectable="on">...</span>
    </span>
    <input name="Destination" class="k-textbox input-validation-error" id="Destination" aria-
      disabled="false" aria-invalid="true" aria-readonly="false" aria-required="true" aria-
      describedby="Destination-error" style="display: none;" type="text" data-val-required="The Destination
      field is required." data-val="true" data-val-length-max="15" data-val-length="The field Destination must
      be a string with a maximum length of 15." data-role="combobox" value=""></input>
    </span>
    <span class="field-validation-error" data-valmsg-replace="true" data-valmsg-for="Destination">...</span>
  </div>
```

همانطور که در تصویر مشاهده می‌کنید، با اتصال kendoComboBox به یک فیلد، این فیلد در حالت مخفی قرار می‌گیرد و ویجت کندو یو آی بجای آن نمایش داده خواهد شد. در این حالت چون در فایل jquery.validate.js چنین تنظیمی وجود دارد:

```
$.extend( $.validator, {
  defaults: {
    //...
    ignore: ":hidden",
```

به صورت پیش فرض از اعتبارسنجی فیلدهای مخفی صرفنظر می‌شود. راه حل آن نیز ساده‌است. تنها باید خاصیت ignore را بازنویسی کرد و تغییر داد:

```
<script type="text/javascript">
  $(function () {
    var form = $('#Form1');
    form.data('validator').settings.ignore = ''; // default is ":hidden".
  });
</script>
```

در اینجا صرفاً خاصیت ignore فرم یک، جهت در نظر گرفتن فیلدهای مخفی تغییر کرده‌است. اگر می‌خواهید این تنظیم را به تمام فرم‌ها اعمال کنید، می‌توان از دستور ذیل استفاده کرد:

```
<script type="text/javascript">
  $.validator.defaults({
    ignore: ""
  });
</script>
```

یکپارچه کردن سیستم اعتبارسنجی Kendo UI با سیستم اعتبارسنجی ASP.NET MVC

در مطلب « [اعتبارسنجی ورودی‌های کاربر در Kendo UI](#) » با زیرساخت اعتبارسنجی Kendo UI آشنا شدید. برای اینکه بتوان این سیستم را با ASP.NET MVC یکپارچه کرد، نیاز است دو کار صورت گیرد:
الف) تعریف فایل kendo.aspnetmvc.js به صفحه اضافه شود:

```
<script src="~/Scripts/kendo.aspnetmvc.js" type="text/javascript"></script>
```

ب) همانند قبل، متد kendoValidator بر روی فرم فراخوانی شود تا سیستم اعتبارسنجی Kendo UI در این ناحیه فعال گردد:

```
<script type="text/javascript">
    $(function () {
        $("form").kendoValidator();
    });
</script>
```

پس از آن خواهیم داشت:

OrderDetail

❗ The OrderDetailId field is required.

Origin

لطفا فیلد منبع را وارد کنید

Net Wt

❗ The Net Wt field is required.

Value Date

❗ The Value Date field is required.

Destination

❗ The Destination field is required.

Submit

فایل kendo.aspnetmvc.js در بسته‌ی مخصوص Kendo UI تهیه شده برای ASP.NET MVC موجود است (در پوشه‌ی js آن)، عملکردی مشابه فایل jquery.validate.unobtrusive مایکروسافت دارد. کار آن وفق دادن و ترجمه‌ی اعتبارسنجی unobtrusive به روش Kendo UI است.

این فایل را از اینجا می‌توانید دریافت کنید:

[kendo.mvc.zip](#)

البته باید دقت داشت که در حال حاضر فقط ویژگی‌های ذیل از ASP.NET MVC توسط kendo.aspnetmvc.js پشتیبانی می‌شوند:

Required

StringLength
Range
RegularExpression

برای تکمیل آن می‌توان از یک پروژه‌ی سورس باز به نام [Moon.Validation for KendoUI Validator](#) استفاده کرد. برای مثال remote validation مخصوص Kendo UI را [اضافه کرده‌است](#) .

نظرات خوانندگان

نویسنده: امیر

تاریخ: ۱۳:۰ ۱۳۹۳/۰۸/۲۸

سئوالی برای من پیش اومد امکان استفاده از مدل mvc داخل کنترلرهای جاوااسکریپت کندو هست ؟ مثلا اگه combobox بشه آیدیش در مدل اصلی انتخاب بشه ؟

نویسنده: وحید نصیری

تاریخ: ۱۳:۴۶ ۱۳۹۳/۰۸/۲۸

« استفاده از Expression ها جهت ایجاد Strongly typed view در ASP.NET MVC »

نویسنده: شاکری

تاریخ: ۱۰:۳۷ ۱۳۹۳/۱۰/۰۲

من برای فعال سازی remote validation از پروژه Moon استفاده کردم اما باز هم عمل نمیکند!

```
[Moon.Web.Validation.RemoteValidator("IsUserExist", "Admin", HttpMethod = "POST", ErrorMessage = "نام کاربری قبلا ثبت شده است")
public string Username { get; set; }
```

```
@Html.EditorFor(model => model.Username)
@Html.ValidationMessageFor(model => model.Username)
```

```
[HttpPost]
public ActionResult IsUserExist(string Username)
{
    if (userSerive.isUser(Username)) return Json(true);
    return Json(false);
}
```

نویسنده: وحید نصیری

تاریخ: ۱۵:۲۸ ۱۳۹۳/۱۰/۰۳

کتابخانه‌ی ذکر شده را حذف و سپس به روش زیر برای فعال سازی remote validation عمل کنید:

```
$.validator.methods.remote = function () { /* disabled */ };
$("form").kendoValidator({
    onfocusout: true,
    onkeyup: true,
    rules: {
        remote: function (input) {
            var remoteAttr = input.attr("data-val-remote-url");
            if (typeof remoteAttr === typeof undefined || remoteAttr === false) {
                return true;
            }

            var isInvalid = true;
            var data = {};
            data[input.attr('name')] = input.val();

            $.ajax({
                url: remoteAttr,
                mode: "abort",
                port: "validate" + input.attr('name'),
                dataType: "json",
                type: input.attr("data-val-remote-type"),
                data: data,
                async: false,
                success: function (response) {
                    isInvalid = response;
                }
            });
        }
    }
});
```

```

    });
    return !isValid;
  }
},
messages: {
  remote: function (input) {
    return input.data('val-remote');
  }
}
});

```

- در اینجا در ابتدا متد remote کتابخانه‌ی jQuery Validator غیرفعال می‌شود. سپس یک rule جدید، به kendoValidator به نام دلخواه remote اضافه شده‌است. چون rule‌های kendoValidator اعمال async را پشتیبانی نمی‌کنند، در درخواست ajax آن async: false تنظیم شده‌است. به این ترتیب سطر پس از ajax، پس از پایان کار عملیات ajax فراخوانی می‌شود و در این حالت kendoValidator بدون مشکل کار خواهد کرد.

- سمت سرور آن هم مانند قبل به همراه استفاده از ویژگی Remote است که از آن صرفاً برای مقدار دهی data-val-remote-url و val-remote که در rule جدید استفاده می‌شوند، کمک گرفته خواهد شد.

نویسنده: شاکری

تاریخ: ۱۱:۲۸ ۱۳۹۳/۱۰/۰۴

خیلی ممنون بابت پاسخگویی

این روش در حالتی جواب می‌دهد که ما اطلاعات رو از طریق فرم ارسال کنیم.

اما من برای درج یا ویرایش از template در گرید کندو به صورت popup استفاده کردم که در این حالت از فرم برای ارسال اطلاعات استفاده نمی‌کند!

نویسنده: وحید نصیری

تاریخ: ۱۱:۳۷ ۱۳۹۳/۱۰/۰۴

مراجعه کنید به مطلب «[اعتبارسنجی ورودی‌های کاربر در Kendo UI](#)». در انتهای مطلب در مورد «اعتبارسنجی سفارشی در DataSource» بحث شده‌است. Kendo UI Grid هم اطلاعات اعتبارسنجی فیلدهای خودش را از data source دریافت می‌کند و ... اصول طراحی اعتبارسنجی آن، هیچ تفاوتی با نکته‌ی عنوان شده ندارد (یک custom rule سفارشی را به نام remote، دقیقاً مانند همین مثال می‌توانید اضافه کنید).

نویسنده: شاکری

تاریخ: ۱۲:۵۴ ۱۳۹۳/۱۰/۰۴

باز هم ممنون بابت پاسخگویی

در حالتی که از popup template داخل گرید استفاده کنیم برای remote validation همیشه از مثال شما با اندکی تغییرات استفاده کرد.

```

(function($, kendo) {
  $.extend(true, kendo.ui.validator, {
    rules: {
      remote: function(input) {
        var remoteAttr = input.attr("data-val-remote-url");
        if (typeof remoteAttr === typeof undefined || remoteAttr === false) {
          return true;
        }

        var isValid = true;
        var data = {};
        data[input.attr('name')] = input.val();

        $.ajax({
          url: remoteAttr,
          mode: "abort",
          port: "validate" + input.attr('name'),
          dataType: "json",
          type: input.attr("data-val-remote-type"),

```

```
        data: data,
        async: false,
        success: function(response) {
            isInvalid = response;
        }
    });
    return !isInvalid;
},
messages: {
    remote: function(input) {
        return input.data('val-remote');
    }
}
});
})(jQuery, kendo);
```