عنوان: **قابل ویرایش کننده‌ی فوق العاده x-editable ؛ قسمت اول**
نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۳/۱۱ ۱۷:۵
آدرس: www.dotnettips.info
گروه‌ها: jQuery, Twitter Bootstrap, Kendo UI

من در یکی از پروژه‌ها از Kendo UI Treeview استفاده کردم و قصد داشتم قابلیت تغییر نام را به گره‌ها بدهم. به همین جهت پس از جستجو به x-editable برخوردم. این کتابخانه‌ی جاواسکریپتی در ابتدا برای قالب‌های بوت استراپ طراحی شده بود که در حال حاضر اینگونه نیست و به راحتی در هر پروژه‌ای که فقط جی کوئری صدا زده شده باشد، قابل اجرا است و نسخه‌ی مخصوص Angular آن هم در این آدرس قرار دارد. همچنین این قابلیت اختیاری و پیش فرض را دارد که به طور خودکار اطلاعات تغییر یافته را به سمت url ایی که شما تعیین می‌کنید، ارسال کند. برای همین نیازی به استفاده جداگانه از jQuery Ajax برای ارسال اطلاعات نیست و البته در انتهای مقاله هم هنگام استفاده از درخت کندو به مشکلی برخوردم که آن را هم بررسی می‌کنیم.

وارد کردن کتابخانه ها

این کتابخانه شامل دو فایل css و JS می‌باشد که بسته به محیطی که در آن کار می‌کنید متفاوت هستند. در این صفحه شما می‌توانید برای 4 محیط Bootstrap2 , JqueryUI, Jquery و Bootsrap3 بسته‌ی مخصوصش را یا به صورت دانلود فایل‌ها یا از طریق CDN دریافت نمایید. در اینجا هم یک دمو از قابلیت‌های آن قابل مشاهده است.

برای شروع، کتابخانه‌ی مورد نظر خود را دریافت و آن‌ها را به صفحه‌ی خود اضافه نمایید. در صورتیکه از Bootstrap استفاده می‌کنید، ابتدا فایل‌های زیر را اضافه کنید:

```
<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="http://code.jquery.com/jquery-2.0.3.min.js"></script>
    <script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>
```

سپس فایل‌های x-editable را صدا بزنید.

اولین حالتیکه میتوانید با این کتابخانه کار کنید، استفاده از خاصیت *-data است. نمونه زیر را در نظر بگیرید:

```
<a href="#" id="favsite" data-type="text" data-pk="1" data-
url="@Url.Action(MVC.Categories.EditCategory())"
data-title="Enter your favorite site">dotnettips.info</a>
```

به تعداد هر عنصری که نیاز است، اینکار را انجام دهید و به هر کدام یک id انتساب دهید. بعد از آن کد زیر را در قسمت script بنویسید:

```
$(document).ready(function () {
        $('#favsite').editable();
    });
```

در صورتیکه قصد دارید خصوصیاتی از اشیاء را که برای همه‌ی عنصرهای معرفی شده یکسان است، معرفی کنید می‌توانید از کد زیر بهره ببرید:

```
$(document).ready(function () {
        $.fn.editable.defaults.mode = 'inline';
        $('#favsite').editable();
    });
```

کد بالا حالت ویرایش تمام عناصر معرفی شده را به inline تغییر می‌دهد.

حالت بعدی که می‌توان استفاده کرد به شکل زیر است:

```
<a href="#" id="favsite" >dotnettips.info</a>
```

```
$.fn.editable.defaults.mode = 'inline';
    $(document).ready(function () {
        $('#favsite').editable({
            type: 'text',
            pk: 1,
            url: '@Url.Action(MVC.Categories.EditCategory())',
            title: 'Enter your favorite site'
        });
    });
```

خوبی این روش این است که می‌توان اطلاعات بیشتری چون رویدادها را به آن پاس داد. تا الان با نحوه‌ی انتساب آن به اشیاء آشنا شدیم. اجازه دهید تا با خصوصیات آن آشنا شویم.

AjaxOptions	همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید: `ajaxOptions: { type: 'put', dataType: 'json' }`
Anim	این ویژگی که تنها در حالت inline پاسخ می‌دهد، می‌تواند زمان بسته شدن x-editable را تغییر دهد که به طور پیش فرض با false مقداردهی شده است. جهت تغییر زمان بسته شدن، کد زیر را وارد نمایید: `anim:'false' //or anim: { duration: 2000 }`
autotext	در انتهای جدول آمده است.
defaultValue	در صورتیکه عنصر مورد نظر محتوایی نداشته باشد و این خصوصیت را مقداردهی کنید، موقع ویرایش، این عبارت تعیین شده نمایش می‌یابد. در مثال بالا باید متن تگ a را حذف کرده تا نتیجه را ببینید: (البته فیلد value نباید مقداری داشته باشد) `defaultValue: 'default val' //or defaultValue: undefined //or defaultValue: null` در بقیه‌ی حالات، ویرایشگر خالی از متن خواهد بود و مقدار پیش فرض آن نال است.
disabled	false کردن این ویژگی باعث غیرفعال شدن x-editable بر

AjaxOptions	همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید: `ajaxOptions: { type: 'put', dataType: 'json' }`
	روی کنترل جاری میگردد.
display	خاصیت display یا مقدار بولین false را دریافت می‌کند، یا نال، یا یک تابع callback را می‌توان به آن پاس داد. این خصوصت زمانی صدا زده می‌شود که اطلاعات به سمت آدرس سرور رفته و با موفقیت بازگشت داده می‌شوند (در صورتی که این ویژگی غیرفعال باشد، بلافاصله بعد از تایید کاربر، از اطلاعات وارد شده صدا زده می‌شود) و سپس متن جدید عنصر تغییر می‌یابد. حال اگر این خاصیت نال که مقدار پیش فرض آن است باشد، متن تغییر می‌یابد. ولی اگر false باشد، متن سابق باقی خواهد ماند و در صورتیکه تابعی به آن پاس داده باشید، طبق تابع شما عمل خواهد کرد. پارامترهایی که تابع شما می‌تواند داشته باشد به شرح زیر است: value : مقدار جدید response : پاسخ سرور (در صورتی که ارسال از طریق Ajax صورت گرفته باشد) و در صورتیکه عنصر شما checlklist یا select باشد که حاوی منبعی از مقادیر هست، مقادیرشان در قالب یک آرایه با نام sourceData بازگشت خواهد خورد. برای دسترسی به آیتم‌های انتخابی هم از کد زیر استفاده می‌کنیم: `$.fn.editableutils.itemsByValue(value, sourceData)`
emptyclass	معرفی یک کلاس css برای موقعیکه عنصر خالی است.
emptytext	در صورتی خالی بودن عنصر، این متن را برای عنصر نمایش بده.
highlight	بعد از به روز رسانی متن عنصر، آن را با این رنگ highlight خواهد کرد و کد رنگی باید در مبنای هگز باشد. مقدار پیش فرض آن false است.
mode	دو حالت نمایشی دارد که پیش فرض آن popup است و با باز کردن یک پنجره، مقدار جدید را دریافت می‌کند. مورد بعدی inline است که به جای باز کردن پنجره، متن عنصر را به حالت ویرایش تغییر میدهد.
name	نام فیلدی که مقدارش تغییر می‌کند.
onblur	زمانی که کاربر فوکوس را از ویرایشگر می‌گیرد، ویرایشگر چه

AjaxOptions	همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید: `ajaxOptions: { type: 'put', dataType: 'json' }`
	پاسخی باید به آن بدهد، باز بماند؟ *ignore* ، بسته شود؟ *cancel* و یا مقدار داده شده را تایید کند؟ *submit*
params	پارامترهای درخواست ایجکسی که کنترل در حالت پیش فرض ارسال می‌کند؛ شامل Pk که آن را با id رکورد پر می‌کنیم. name نام فیلدی که تغییر یافته است و value که مقدار جدید است. در صورتیکه دوست دارید اطلاعات اضافی‌تری نیز ارسال شوند، می‌توانید از این خاصیت استفاده کنید و پارامترها را در قالب Object به آن پاس کنید. ولی اگر بخواهید در کل همه‌ی پارامترها را رونویسی کنید باید یک تابع را به آن پاس کنید: `params: function(params) { //در این حالت پارامترهای پیش فرض ارسال نشده و تنها پارامترهای معرفی شده در این تابع ارسال می‌شوند params.a = 1; return params; }`
pk	کلید اصلی رکورد شما در دیتابیس یا هان id است. در صورتی که از کلیدهای ترکیبی استفاده می‌کنید، نگران نباشید فکر آن را هم کرده اند. `//کلید عدد pk:1, //کلید رشته ای pk:'dp123' //کلید ترکیبی pk:{id: 1, lang: 'en'} //معرفی یک تابع به آن و بازگشت یکی از مقادیر بالا بعد از محاسبات pk:function() { }`
Placement	این ویژگی فقط به درد حالت Popup می‌خورد که پنجره را کجای عنصر نمایش دهد و شامل چهار مقدار left,right,top,bottom می‌شود.
saveonchange	زمانی که مقدار جدید، برابر مقدار فعلی باشد و این خاصیت

AjaxOptions	همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید: `ajaxOptions: { type: 'put', dataType: 'json' }`
	false باشد، هیچ تغییری رخ نخواهد داد. ولی اگر برابر true باشد ،مقدار جدید اسال و جایگزین مقدار فعلی خواهد شد. مقدار پیش فرض آن false است.
selector	با استفاده از این خصوصیت در عنصر انتخابی به دنبال عناصری که در selector تعیین شده می‌گردد و حالت ویرایش را روی آن‌ها فعال می‌کند. در این حالت استفاده از خصوصیات emptytext و autotext در ابتدای امر ممکن نیست و بعد از اولین کلیک قابل استفاده هستند. نکته بعدی اینکه شما باید کلاس‌های زیر را دستی اضافه کنید. کلاس editable-click برای همه کنترل‌ها وکلاس editable-empty به کنترل‌های بدون مقدار و برای مقداردهی کنترلهای بدون مقدار میتوان از خاصیت data-value='' استفاده کرد. `<div id="user"> <!-- empty --> Empty <!-- non-empty --> Operator </div> <script> $('#user').editable({ selector: 'a', url: '/post', pk: 1 }); </script>`
send	سه مقدار auto,always و never را دریافت می‌کند. موقعی که شما آن را روی auto تنظیم کنید؛ در صورتی مقادیر به سمت سرور ارسال می‌شوند که دو خاصیت url و pk تعریف شده باشند. در غیر این صورت ویرایش فقط در حالت محلی و روی سیستم کاربر رخ خواهد داد.
showbuttons	در صورتیکه با false مقداردهی شود، تایید فرم به طور خودکار انجام می‌گیرد و اگر با یکی از مقادیر left یا Bottom پر شود، دکمه‌ها را در آن قسمت نشان می‌دهد.
success	اطلاعات به سمت سرور رفته و با موفقیت با کد 200 بازگشت داده شده‌اند. در مستندات نوشته است، هر کد وضعیتی غیر از

AjaxOptions	همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید: `ajaxOptions: { type: 'put', dataType: 'json' }`
	200 بازگشت داده شود، به سمت خاصیت error هدایت می‌شو.د ولی آن طور که من با httpresponsemessage تست کردم، چنین چیزی را مشاهده نکردم و مجددا success صدا زده شد. پس بهتر هست داده‌ای را که به سمت کنترل برگشت می‌دهید، خودتان کنترل کنید. به خصوص اگر انتقال اطلاعات صحیح باشد. ولی اگر در دیتابیس، تغییر با خطا روبرو گردد بهتر است نتیجه‌ی آن ارسال شده و از تغییر مقدار فعلی ممانعت به عمل آورید. `success: function(response, newValue) { if(!response.success) return response.msg; }`
toggle	اگر قصد دارید که باز و بسته کردن ویرایشگر را بر عهده‌ی مثلا یک دکمه‌ی روی صفحه بگذارید، این خصوصیت به شما کمک می‌کند: `$('#edit-button').click(function(e) { e.stopPropagation(); $('#favsite').editable('toggle'); });` به جای toggle نیز می‌توان از show و hide هم استفاده کرد. وجود عبارت e.stopPropagation جهت باز شدن صحیح ویرایشگر الزامی است.
type	نوع ویرایشی را که قرار است انجام گیرد، مشخص می‌کند. text برای متن، date برای تاریخ، textarea جهت متون طولانی‌تر نسبت به text و بسیاری از موارد دیگر
unsavedclass	این کلاس موقعی اعمال می‌گردد که اطلاعاتی را ویرایش کرده‌اید، ولی اطلاعاتی به سمت سرور ارسال نشده است. مثلا pk مقداردهی نشده یا send=never قرار داید و یا اینکه به صورت محلی ذخیره می‌کنید و می‌خواهید در آخر همه‌ی اطلاعات را ارسال کنید. این خاصیت به طور پیش فرض با کلاس editable-unsaved مقداردهی شده که می‌توانید با نال کردن، از شرش خلاص شوید.
url	این خاصیت با آدرس سمت سرور پر می‌شود. ولی میتوان به آن یک تابع هم پاس کرد که این تابع جایگزین درخواست

AjaxOptions	همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید: `ajaxOptions: { type: 'put', dataType: 'json' }`
	ایجکسی خودش خواهد شد و برای بازگشت دادن نتیجه‌ی این درخواست به سمت تابع‌های callback خودش می‌توانید یک deferred object را برگشت دهید: `url: function(params) { var d = new $.Deferred; if(params.value === 'abc') { //returning error via deferred object return d.reject('error message'); } else { //async saving data in js model someModel.asyncSaveMethod({ ..., success: function(){ d.resolve(); } }); return d.promise(); } }`
validate	مقدار پیش فرض آن نال است و می‌توان به آن یک تابع را جهت اعتبارسنجی سمت کلاینت پاس داد. به عنوان آرگومان، مقدار جدیدی را ارسال کرده و در آن به اعتبارسنجی می‌پردازید. در صورتی که مقدار، نامعتبر باشد، می‌توانید یک پیام خطا از نوع رشته‌ای را برگردانید. در صورتی که از نسخه‌ی 1.5.1 به بعد استفاده می‌کنید، دریافت یک object با مقادیر زیر نیز ممکن شده است: newValue: مقدار جدید و جایگزین مقدار غیر معتبر. msg : پیام خطا. به کدهای زیر در سه حالت نگاه کنید: `validate: function (value) { if ($.trim(value) == '') { //در تمامی نسخه‌های یک پیام متنی باز میگردد return 'This field is required'; //1.5.1 //یک مقدار جدید برگشت میدهد که بلافاصله آن را // تایید میکند و متن عنصر به روز می‌شود return { newValue: 'validated' }; //متن جدید ار ارسال کرده و پیام هشدار را نشان میدهد //ولی تایید نمی‌کند و منتظر تایید کاربر است return { newValue: 'validated',`

AjaxOptions	همانطور که متوجه شدید به طور خودکار اطلاعات ویرایش شده، به سمت آدرس داده شده، به شیوه Post ارسال می‌گردند. در صورتیکه قصد دست بردن در نوع درخواست را دارید، می‌توانید از این ویژگی استفاده کنید: `ajaxOptions: { type: 'put', dataType: 'json' }`
	`msg: 'This field is required' }; } }`
value	مقدار پیش فرضی که در ویرایشگر نشان می‌دهد و اگر مقداردهی نشود، از متن عنصر استفاده می‌کند.
autotext	سه مقدار دارد auto (پیش فرض)،always و never. موقعیکه عنصر شما متنی نداشته باشد و روی auto تنظیم شده باشد، مقدار value را به عنوان متن عنصر نمایش می‌دهد. always کاری ندارد که عنصر شما متن دارد یا خالی است؛ مقدار value به آن انتساب داده خواهد شد. never هیچگاه.

در قسمت بعدی که قسمت پایانی است مطالب را ادامه می‌دهیم.