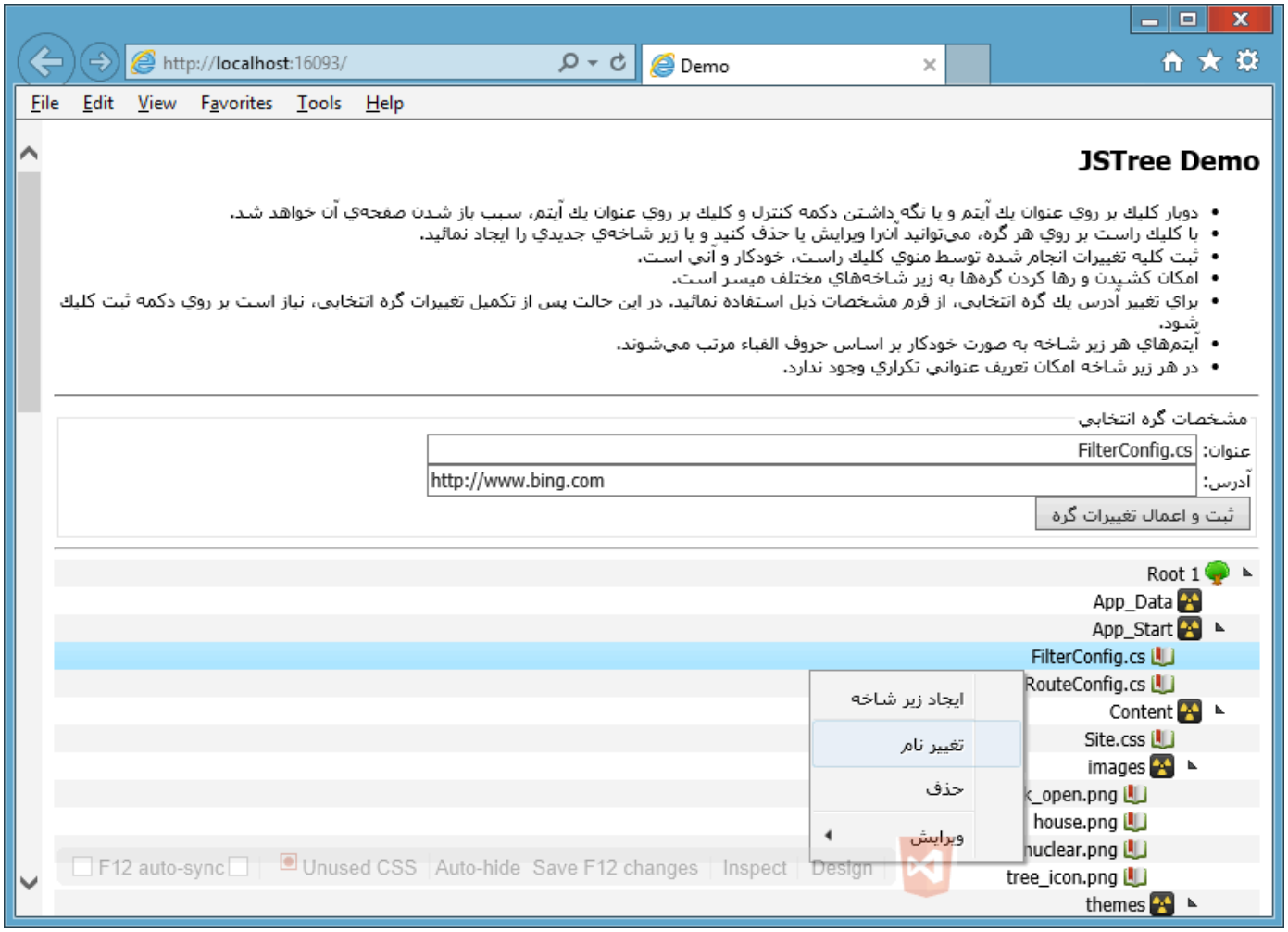
عنوان: **استفاده از افزونه‌ی jsTree در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۱۹ ۷:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jQuery

jsTree یکی از افزونه‌های بسیار محبوب jQuery جهت نمایش ساختارهای سلسله مراتبی، خود ارجاع دهنده و تو در تو است. روش ابتدایی استفاده از آن تعریف یک سری ul و li ثابت در صفحه و سپس فراخوانی این افزونه بر روی آن‌ها است که سبب نمایش درخت‌واره‌ای این اطلاعات خواهد شد. روش پیشرفته‌تر آن به همراه کار با داده‌های JSON و دریافت پویای اطلاعات از سرور است که در ادامه به بررسی آن خواهیم پرداخت.



دریافت افزونه‌ی jsTree

برای دریافت افزونه‌ی jsTree می‌توان به مخزن کد آن در Github مراجعه کرد و همچنین مستندات آن‌را در سایت jstree.com قابل مطالعه هستند.

تنظیمات مقدماتی jsTree

در این مطلب فرض شده‌است که فایل jstree.min.js، در پوشه‌ی Scripts و فایل‌های CSS آن در پوشه‌ی Content\themes\default کپی شده‌اند.

به این ترتیب layout برنامه چنین شکلی را خواهد یافت:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>

    <link href="~/Content/Site.css" rel="stylesheet" />
    <link href="~/Content/themes/default/style.min.css" rel="stylesheet" />
    <script src="~/Scripts/jquery.min.js"></script>
    <script src="~/Scripts/jstree.min.js"></script>
</head>
<body dir="rtl">
    @RenderBody()

    @RenderSection("scripts", required: false)
</body>
</html>
```

نمایش راست به چپ اطلاعات

در کدهای این افزونه به تگ body و ویژگی dir آن برای تشخیص راست به چپ بودن محیط دقت می‌شود. به همین جهت این تعریف را در layout فوق ملاحظه می‌کنید. برای مثال اگر به فایل jstree.contextmenu.js (موجود در مجموعه سورس‌های این افزونه) مراجعه کنید، یک چنین تعریفی قابل مشاهده است:

```
right_to_left = $("body").css("direction") === "rtl";
```

تهیه ساختاری جهت ارائه‌ی خروجی JSON

با توجه به اینکه قصد داریم به صورت پویا با این افزونه کار کنیم، نیاز است بتوانیم ساختار سلسله مراتبی مدنظر را با فرمت JSON ارائه دهیم. در ادامه کلاس‌هایی که معادل فرمت JSON قابل قبول توسط این افزونه را تولید می‌کنند، ملاحظه می‌کنید:

```
using System.Collections.Generic;

namespace MvcJSTree.Models
{
    public class JsTreeNode
    {
        public string id { set; get; } // نام این خواص باید با مستندات هماهنگ باشد
        public string text { set; get; }
        public string icon { set; get; }
        public JsTreeNodeState state { set; get; }
        public List<JsTreeNode> children { set; get; }
        public JsTreeNodeLiAttributes li_attr { set; get; }
        public JsTreeNodeAAttributes a_attr { set; get; }

        public JsTreeNode()
        {
            state = new JsTreeNodeState();
            children = new List<JsTreeNode>();
            li_attr = new JsTreeNodeLiAttributes();
            a_attr = new JsTreeNodeAAttributes();
        }
    }

    public class JsTreeNodeAAttributes
    {
        // به هر تعداد و نام اختیاری می‌توان خاصیت تعریف کرد
        public string href { set; get; }
    }

    public class JsTreeNodeLiAttributes
    {
        // به هر تعداد و نام اختیاری می‌توان خاصیت تعریف کرد
        public string data { set; get; }
    }
```

```
    public class JsTreeNodeState
    {
        public bool opened { set; get; }
        public bool disabled { set; get; }
        public bool selected { set; get; }

        public JsTreeNodeState()
        {
            opened = true;
        }
    }
}
```

در اینجا به چند نکته باید دقت داشت:
- هر چند اسامی مانند a_attr، مطابق اصول نامگذاری دات نت نیستند، ولی این نام‌ها را تغییر ندهید. زیرا این افزونه دقیقا به همین نام‌ها و با همین املاء نیاز دارد .
- id، می‌تواند دقیقا معادل id یک رکورد در بانک اطلاعاتی باشد. Text عنوان گره‌ای (node) است که نمایش داده می‌شود. icon در اینجا مسیر یک فایل png است جهت نمایش در کنار عنوان هر گره. توسط state می‌توان مشخص کرد که زیر شاخه‌ی جاری به صورت باز نمایش داده شود یا بسته. به کمک خاصیت children می‌توان زیر شاخه‌ها را تا هر سطح و تعدادی که نیاز است تعریف نمود.
- خاصیت‌های li_attr و a_attr کاملا دلخواه هستند. برای مثال در اینجا دو خاصیت href و data را در کلاس‌های مرتبط با آن‌ها مشاهده می‌کنید. می‌توانید در اینجا به هر تعداد ویژگی سفارشی دیگری که جهت تعریف یک گره نیاز است، خاصیت اضافه کنید.

ساده‌ترین مثالی که از ساختار فوق می‌تواند استفاده کند، اکشن متد زیر است:

```
[HttpPost]
        public ActionResult GetTreeJson()
        {
            var nodesList = new List<JsTreeNode>();

            var rootNode = new JsTreeNode
            {
                id = "dir",
                text = "Root 1",
                icon = Url.Content("~/Content/images/tree_icon.png"),
                a_attr = { href = "http://www.bing.com" }
            };
            nodesList.Add(rootNode);

            nodesList.Add(new JsTreeNode
            {
                id = "test1",
                text = "Root 2",
                icon = Url.Content("~/Content/images/tree_icon.png"),
                a_attr = { href = "http://www.bing.com" }
            });

            return Json(nodesList, JsonRequestBehavior.AllowGet);
        }
```

در ابتدا لیست گره‌ها تعریف می‌شود و سپس برای نمونه در این مثال، دو گره تعریف شده‌اند و در ادامه با فرمت JSON در اختیار افزونه قرار گرفته‌اند.
بنابراین ساختارهای خود ارجاع دهنده را به خوبی می‌توان با این افزونه وفق داد.

فعال سازی اولیه سمت کلاینت افزونه jsTree

برای استفاده‌ی پویای از این افزونه در سمت کلاینت، فقط نیاز به یک DIV خالی است:

```
<div id="jstree">
</div>
```

سپس jstree را بر روی این DIV فراخوانی می‌کنیم:

```
$('#jstree').jstree({
                "core": {
                    "multiple": false,
                    "check_callback": true,
                    'data': {
                        'url': '@getTreeJsonUrl',
                        "type": "POST",
                        "dataType": "json",
                        "contentType": "application/json; charset=utf8",
                        'data': function (node) {
                            return { 'id': node.id };
                        }
                    },
                    'themes': {
                        'variant': 'small',
                        'stripes': true
                    }
                },
                "types": {
                    "default": {
                        "icon": '@Url.Content("~/Content/images/bookmark_book_open.png")'
                    },
                },
                "plugins": ["contextmenu", "dnd", "state", "types", "wholerow", "sort", "unique"],
                "contextmenu": {
                    "items": function (o, cb) {
                        var items = $.jstree.defaults.contextmenu.items();
                        items["create"].label = "ایجاد زیر شاخه";
                        items["rename"].label = "تغییر نام";
                        items["remove"].label = "حذف";
                        var cpp = items["ccp"];
                        cpp.label = "ویرایش";
                        var subMenu = cpp["submenu"];
                        subMenu["copy"].label = "کپی";
                        subMenu["paste"].label = "پیست";
                        subMenu["cut"].label = "برش";
                        return items;
                    }
                }
            });
```

توضیحات

- multiple : false به این معنا است که نمی‌خواهیم کاربر بتواند چندین گره را با نگه داشتن دکمه‌ی کنترل انتخاب کند.
- check_callback : true کدهای مرتبط با منوی کلیک سمت راست ماوس را فعال می‌کند.
- در قسمت data کار تبادل اطلاعات با سرور جهت دریافت فرمت JSON ایی که به آن اشاره شد، انجام می‌شود. متغیر getTreeJsonUrl یک چنین شکلی را می‌تواند داشته باشد:

```
@{
ViewBag.Title = "Demo";
var getTreeJsonUrl =  Url.Action(actionName: "GetTreeJson", controllerName: "Home");
}
```

- در قسمت themes مشخص کرده‌ایم که از قالب small آن به همراه نمایش یک درمیان پس زمینه‌ی روشن و خاکستری استفاده شود. قالب large نیز دارد.
- در قسمت types که مرتبط است با افزونه‌ای به همین نام، آیکن پیش فرض یک نود جدید ایجاد شده را مشخص کرده‌ایم.
- گزینه‌ی plugins، لیست افزونه‌های اختیاری این افزونه را مشخص می‌کند. برای مثال contextmenu منوی کلیک سمت راست ماوس را فعال می‌کند، dnd همان کشیدن و رها کردن گره‌ها است در زیر شاخه‌های مختلف. افزونه‌ی state، انتخاب جاری کاربر را در سمت کلاینت ذخیره و در مراجعه‌ی بعدی او بازیابی می‌کند. با ذکر افزونه‌ی wholerow سبب می‌شویم که انتخاب یک گره، معادل انتخاب یک ردیف کامل از صفحه باشد. افزونه‌ی sort کار مرتب سازی خودکار اعضای یک زیر شاخه را انجام می‌دهد. افزونه‌ی unique سبب می‌شود تا در یک زیر شاخه نتوان دو عنوان یکسان را تعریف کرد.
- در قسمت contextmenu نحوه‌ی بومی سازی گزینه‌های منوی کلیک راست ماوس را مشاهده می‌کنید. در حالت پیش فرض، عناوینی مانند create، rename و امثال آن نمایش داده می‌شوند که به نحو فوق می‌توان آن‌را تغییر داد.

با همین حد تنظیم، این افزونه کار نمایش سلسله مراتبی اطلاعات JSON ایی دریافت شده از سرور را انجام می‌دهد.

ذخیره سازی گره‌های جدید و تغییرات سلسله مراتب پویای تعریف شده در سمت سرور

همانطور که عنوان شد، اگر افزونه‌ی اختیاری contextmenu را فعال کنیم، امکان افزودن، ویرایش و حذف گره‌ها و زیر شاخه‌ها را خواهیم یافت. برای انتقال این تغییرات به سمت سرور، باید به نحو ذیل عمل کرد:

```
$('#jstree').jstree({
              تمام تنظیمات مانند قبل //
            }).on('delete_node.jstree', function (e, data) {
                })
                .on('create_node.jstree', function (e, data) {
                })
                .on('rename_node.jstree', function (e, data) {
                })
                .on('move_node.jstree', function (e, data) {
                })
                .on('copy_node.jstree', function (e, data) {
                })
                .on('changed.jstree', function (e, data) {
                })
                .on('dblclick.jstree', function (e) {
                })
                .on('select_node.jstree', function (e, data) {
                });
```

در اینجا نحوه‌ی تحت کنترل قرار دادن رخدادهای مختلف این افزونه را مشاهده می‌کنید. برای مثال در callback مرتبط با delete_node کار حذف یک گره اطلاع رسانی می‌شود. create_node مربوط است به ایجاد یک گره یا زیر شاخه‌ی جدید. rename_node پس از تغییر نام یک گره فراخوانی خواهد شد. move_node مربوط است به کشیدن و رها کردن یک گره در یک زیر شاخه‌ی دیگر. copy_node برای copy/paste یک گره تعریف شده‌است. Changed یک callback عمومی است. dblclick برای عکس العمل نشان دادن به رخداد دوبار کلیک کردن بر روی یک گره می‌تواند بکار گرفته شود. select_node با انتخاب یک گره فعال می‌شود.
در تمام این حالات، جایی که data در اختیار ما است، می‌توان یک چنین ساختار جاوا اسکریپتی را برای ارسال به سرور طراحی کرد:

```
function postJsTreeOperation(operation, data, onDone, onFail) {
            $.post('@doJsTreeOperationUrl',
                {
                    'operation': operation,
                    'id': data.node.id,
                    'parentId': data.node.parent,
                    'position': data.position,
                    'text': data.node.text,
                    'originalId': data.original ? data.original.id : data.node.original.id,
                    'href': data.node.a_attr.href
                })
                .done(function (result) {
                    onDone(result);
                })
                .fail(function (result) {
                    alert('failed.....');
                    onFail(result);
                });
        }
```

به این ترتیب در سمت سرور می‌توان id یک گره، متن تغییر یافته آن، والد گره و بسیاری از مشخصات دیگر را دریافت و ثبت کرد. پس از تعریف متد postJsTreeOperation فوق، آن‌را باید به callbackهایی که پیشتر معرفی شدند، اضافه کرد؛ برای مثال:

```
.on('create_node.jstree', function (e, data) {
                    postJsTreeOperation('CreateNode', data,
                        function (result) {
                            data.instance.set_id(data.node, result.id);
                        },
```

```
                        function (result) {
                            data.instance.refresh();
                        });
                })
```

در اینجا متد postJsTreeOperation، یک Operation خاص را مانند CreateNode (تعریف شده در enum ایی به نام JsTreeOperation در سمت سرور) به همراه data، به سرور post می‌کند.
و معادل سمت سرور دریافت کننده‌ی این اطلاعات، اکشن متد ذیل می‌تواند باشد:

```
[HttpPost]
        public ActionResult DoJsTreeOperation(JsTreeOperationData data)
        {
            switch (data.Operation)
            {
                case JsTreeOperation.CopyNode:
                case JsTreeOperation.CreateNode:
                    //todo: save data
                    var rnd = new Random(); // آی دی رکورد پس از ثبت در بانک اطلاعاتی دریافت و بازگشت داده شود
                    return Json(new { id = rnd.Next() }, JsonRequestBehavior.AllowGet);

                case JsTreeOperation.DeleteNode:
                    //todo: save data
                    return Json(new { result = "ok" }, JsonRequestBehavior.AllowGet);

                case JsTreeOperation.MoveNode:
                    //todo: save data
                    return Json(new { result = "ok" }, JsonRequestBehavior.AllowGet);

                case JsTreeOperation.RenameNode:
                    //todo: save data
                    return Json(new { result = "ok" }, JsonRequestBehavior.AllowGet);

                default:
                    throw new InvalidOperationException(string.Format("{0} is not supported.", data.Operation));
            }
        }
```

که در آن ساختار JsTreeOperationData به نحو ذیل تعریف شده‌است:

```
namespace MvcJSTree.Models
{
    public enum JsTreeOperation
    {
        DeleteNode,
        CreateNode,
        RenameNode,
        MoveNode,
        CopyNode
    }

    public class JsTreeOperationData
    {
        public JsTreeOperation Operation { set; get; }
        public string Id { set; get; }
        public string ParentId { set; get; }
        public string OriginalId { set; get; }
        public string Text { set; get; }
        public string Position { set; get; }
        public string Href { set; get; }
    }
}
```

این ساختار دقیقا با اعضای شیء جاوا اسکریپتی که متد postJsTreeOperation به سمت سرور ارسال می‌کند، تطابق دارد.
در اینجا Href را نیز مشاهده می‌کنید. همانطور که عنوان شد، اعضای JsTreeNodeAAttributes اختیاری هستند. بنابراین اگر این اعضاء را تغییر دادید، باید خواص JsTreeOperationData و همچنین اعضای شیء تعریف شده در postJsTreeOperation را نیز تغییر دهید تا با هم تطابق پیدا کنند.

چند نکته‌ی تکمیلی

اگر می‌خواهید که با دوبار کلیک بر روی یک گره، کاربر به href آن هدایت شود، می‌توان از کد ذیل استفاده کرد:

```
var selectedData;
                // ...
                 .on('dblclick.jstree', function (e) {
                     var href = selectedData.node.a_attr.href;
                     alert('selected node: ' + selectedData.node.text + ', href:' + href);

                     // auto redirect
                     if (href) {
                         window.location = href;
                     }

                     // activate edit mode
                     //var inst = $.jstree.reference(selectedData.node);
                     //inst.edit(selectedData.node);
                 })
                 .on('select_node.jstree', function (e, data) {
                     //alert('selected node: ' + data.node.text);
                     selectedData = data;
                 });
```

در callback مرتبط با select_node می‌توان به گره انتخابی درستی یافت. سپس می‌توان این گره را در callback متناظر با dblclick برای یافتن href و مقدار دهی window.location که معادل redirect سمت کاربر است، بکار برد.
حتی اگر خواستید که با دوبار کلیک بر روی یک گره، گزینه‌ی ویرایش آن فعال شود، کدهای آن را به صورت کامنت مشاهده می‌کنید.

مثال کامل این بحث را از اینجا می‌توانید دریافت کنید:

MvcJSTree.zip

نظرات خوانندگان

نویسنده: ناصر پورعلی
تاریخ: ۱۳۹۳/۰۵/۰۶ ۱۵:۵

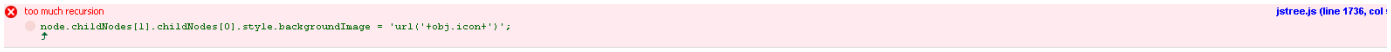
با سلام

من دارم از همین jstree مثال شما استفاده میکنم.

و json زیر رو هم به سمت کلاینت برمیگردونم به صورت صحیح :

```
[{"id":"OrganizationTree","text":"ساختار
سازمانی","icon":"/Content/images/tree_icon.png","state":{"opened":true,"disabled":false,"selected":fals
e},"children":[{"id":"2","text":"آنات","icon":"/Content/images/nuclear.png","state":{"opened":true,"dis
abled":false,"selected":false},"children":[{"id":"4","text":"آموزش","icon":"/Content/images/nuclear.pn
g","state":{"opened":true,"disabled":false,"selected":false},"children":[],"li_attr":{"data":null},"a_a
ttr":{"href":null}},{"id":"5","text":"هیات
مدیره","icon":"/Content/images/nuclear.png","state":{"opened":true,"disabled":false,"selected":false},"
children":[],"li_attr":{"data":null},"a_attr":{"href":null}}],"li_attr":{"data":null},"a_attr":{"href":
null}},{"id":"1","text":"پرنیان","icon":"/Content/images/nuclear.png","state":{"opened":true,"disabled":
false,"selected":false},"children":[{"id":"1","text":"BPM","icon":"/Content/images/nuclear.png","state"
:{"opened":true,"disabled":false,"selected":false},"children":[],"li_attr":{"data":null},"a_attr":{"hre
f":null}},{"id":"3","text":"پشتیبانی","icon":"/Content/images/nuclear.png","state":{"opened":true,"disab
led":false,"selected":false},"children":[],"li_attr":{"data":null},"a_attr":{"href":null}},{"id":"2","t
ext":"فروش","icon":"/Content/images/nuclear.png","state":{"opened":true,"disabled":false,"selected":fa
lse},"children":[],"li_attr":{"data":null},"a_attr":{"href":null}}],"li_attr":{"data":null},"a_attr":{"
href":null}}],"li_attr":{"data":null},"a_attr":{"href":null}}]
```

منتهی همش خطای too much recursion میگیرم، در صورتی که حلقه ای رو هم نمی‌بینم که ایجاد شده باشه داخل رشته json تولید شده.



اینهم تنظیمات فراخوانی

```
<script>
            $(function () {
                $('#jstree').jstree({
                    "core": {
                        "multiple": true,
                        "check_callback": true,
                        'data': {
                            'url': '@getTreeJsonUrl',
                            "type": "POST",
                            "dataType": "json",
                            "contentType": "application/json; charset=utf8",
                            'data': function (node) {
                                return { 'id': node.id };
                            }
                        },
                        'themes': {
                            'variant': 'large',
                            'stripes': false
```

```
                    }
                },
                "types": {
                    "default": {
                        "icon": '@Url.Content("~/Content/images/bookmark_book_open.png")'
                    },
                },
                "plugins": ["contextmenu", "dnd", "state", "types", "checkbox", "wholerow", "sort",
"unique", "real_checkboxes"],
                "contextmenu": {
                    "items": function (o, cb) {
                        var items = $.jstree.defaults.contextmenu.items();
                        items["create"].label = "ایجاد زیر شاخه";
                        items["rename"].label = "تغییر نام";
                        items["remove"].label = "حذف";
                        var cpp = items["ccp"];
                        cpp.label = "ویرایش";
                        var subMenu = cpp["submenu"];
                        subMenu["copy"].label = "کپی";
                        subMenu["paste"].label = "پیست";
                        subMenu["cut"].label = "برش";
                        return items;
                    }
                }
            });
        });
    </script>
```

فکر میکنین مشکل از کجا باشه؟

جالب اینجاست که اگه فقط یه سطح پایین برم مشکلی نیست!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۰۷ ۲:۳۳

گره‌های تعریف شده unique ID ندارند. این unique ID در کل tree معنا پیدا می‌کند و الزاما ارتباطی به ID رکورد شما در یک جدول خاص بانک اطلاعاتی ندارد.

نویسنده: ناصر پورعلی
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۶:۱۳

ممنون، مشکل قبلی حل شد.
مشکل دیگه ام اینه که من مجبورم از jquery 1.6.1 استفاده کنم، آیا نسخه ای از jstree هست که با این نسخه jquery به خوبی کار کنه؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۸:۲۸

از jQuery>=1.9 تعدادی از متدهای قدیمی آن مانند live حذف شدند. راه حلی که برای آن وجود دارد استفاده از پروژه‌ای است به نام jQuery migrate . این پروژه متدهای حذف شده را بر اساس API جدید بازنویسی کرده. بنابراین افزونه‌های به روز نشده قدیمی، بدون مشکل با نگارش‌های جدید jQuery کار خواهند کرد.
استفاده از آن هم ساده‌است. تنها کاری که باید انجام دهید، تعریف آخرین نگارش jQuery و سپس افزودن jQuery migrate است:

```
<script src="jquery.js"></script>
<script src="jquery-migrate-1.2.1.js"></script>
```

نویسنده: سعیده
تاریخ: ۱۳۹۳/۰۷/۲۴ ۱۵:۴۶

با سلام و تشکر از آموزشتون.

من از jstree در یک صفحه‌ی html ای استفاده کردم اما دیتا رو از طریق webmethod و ajaxcall دریافت میکنم.
اما نمیدونم چطور باید دیتا رو به‌تری ویو bind کنم.شما برای درست بودن فرمت دیتا یک کلاس تعریف کردین و با اکشن متد دیتا رو بایند کردین...در حال حاضر دیتای من به صورت Id/Title/ParentId هستش. ممنون میشم اگر من رو راهنمایی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۴ ۱۹:۰

برای دریافت JSON، این روزها دیگر کسی از فایل‌های asmx استفاده نمی‌کند. امکان استفاده از ASP.NET Web API با وب فرم‌ها هم وجود دارد. اطلاعات بیشتر
بعد از آن تنها کاری که باید انجام شود، بازگشت مستقیم خروجی GetTreeJson مثال فوق است و سایر مفاهیم آن یکی هست.

نویسنده: رامبد
تاریخ: ۱۳۹۴/۰۴/۰۷ ۱۱:۲

سلام؛ من میخوام با این کد اطلاعات را از بانکی که بصورت کد فرست ایجاد کردم بخونم چطور باید اینکار رو بکنم و نمی‌خوام اطلاعات رو از روت دریافت نکنه؟ اگر منبعی وجود دارد معرفی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۴/۰۷ ۱۱:۴۰

اگر متن را مطالعه کنید، دوبار به مدل‌های « خود ارجاع دهنده » ارجاع شده‌است؛ یعنی همان بحث EF Code First.
اگر سورس انتهای بحث را دریافت کنید، متد PopulateTree بازگشتی آن، نحوه‌ی پر کردن ساختار تو در توی مورد نیاز را نمایش می‌دهد.
همین متد بازگشتی را در مورد مدل‌های خود ارجاع دهنده هم می‌توان بکار برد. نمونه‌ی این نوع متدها در مطلب « ساخت منوهای چند سطحی در ASP.NET MVC » نیز استفاده شده‌است. متد ShowTree آن هم یک متد بازگشتی است.
بنابراین الان مشخص است که مدل‌ها را باید به چه نحوی طراحی کرد. همچنین نحوه‌ی خواندن بازگشتی آن هم همانند متد ShowTree است. از این دو استفاده کنید برای پر کردن ساختار لیستی JsTreeNode ها.

نویسنده: رحیم شیرخانی
تاریخ: ۱۳۹۴/۰۴/۰۸ ۹:۲۹

سلام من این تایپیک‌ها رو بررسی کردم ولی نتونستم این فرایندها رو ادغام کنم ببینید من داخل همین پروژه شما دو تا مدل با نام‌های Category.cs وDataBaseContext.cs تعریف کردم و کد هاشم بصورت زیر است حالا چه جوری با تابع بازگشتی داخل HomeController بجای populatetree اطلاعات رو دریافت کنم در ضمن فیلدها هم طبق گفته خودتون باید مطابق jstree باشه که در مل‌ها تعریف کردم؟

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Text;

namespace MvcJSTree.Models.Entities
{
    public class Category
    {
        public Category()
        {

        }
        public Category(string id,string parentid,string orginialid,string text,string position,string
href)
        {
            this.Id = id;
            this.ParentId = parentid;
            this.OriginalId = orginialid;
            this.Text = text;
            this.Position = position;
```

```
            this.Href = href;
        }

        public string Id { set; get; }
        public string ParentId { set; get; }
        public string OriginalId { set; get; }
        public string Text { set; get; }
        public string Position { set; get; }
        public string Href { set; get; }

        public override string ToString()
        {
            return
string.Format("{0},{1},{2},{3},{4},{5}",this.Id,this.ParentId,this.OriginalId,this.Text,this.Position,t
his.Href);
        }
    }

}
```

کد مربوط به DataBaseContext

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Text;
using System.Data.Entity;

namespace MvcJSTree.Models
{
    public class DataBaseContext:System.Data.Entity.DbContext
    {
        public DataBaseContext()
        {

        }
         static DataBaseContext ()
        {

            System.Data.Entity.Database.SetInitializer(
                new System.Data.Entity.DropCreateDatabaseIfModelChanges<DataBaseContext>());

        }

         public System.Data.Entity.DbSet<DataBaseContext> Category { get; set; }

    }
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۴/۰۸ ۱۰:۱۱

- هیچ الزامی ندارد که ساختار serialization مورد نیاز jstree، با ساختار جدول بانک اطلاعاتی شما یکی باشد.
- جدولی را که طراحی کردید، صرفا با JsTreeOperationData تطابق دارد.
- این جدول اصول شیءگرایی مدل‌های خود ارجاع دهنده را لحاظ نکرده‌است و صرفا یک ساختار ساده‌ی دریافت اطلاعات از کاربر هست و نه بیشتر.
- اگر قرار است با این نوع جداول و کلاس‌های غیر شیءگرا کار کنید، نیاز است SQL خام بنویسید و از مفاهیم CTE استفاده کنید.

نتیجه گیری؟
مدل خودتان را با مدلی که در مقاله‌ی مدل‌های خود ارجاع دهنده عنوان شده، تطبیق دهید تا بتوانید از قابلیت‌های شیءگرای EF استفاده کنید.

نویسنده: رحیم شیرخانی
تاریخ: ۱۳۹۴/۰۴/۰۸ ۱۵:۷

من از EF مدل‌های خود ارجاع دهنده مثالی که زدید استفاده کردم. حالا چطور باید تو js tree آنرا بخونم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۴/۰۸ ۱۶:۳۸

از این مثال ایده بگیرید: CommentsController.zip