

همانطور که از نمونه مثال‌های خود Kendo UI مشاهده میشود ، نحوه استفاده از TreeView آن به صورت زیر است :

```
<div>
@(Html.Kendo().TreeView()
    .Name("treeview")
    .TemplateId("treeview-template")
    .HtmlAttributes(new { @class = "demo-section" })
    .DragAndDrop(true)
    .BindTo(Model.Where(e=>e.ParentFolderID==null).OrderBy(e=>e.Order), mappings =>
    {
        mappings.For<DAL.Folder>(binding => binding
            .ItemDataBound((item, folder) =>
            {
                item.Text = folder.FolderName;
                item.SpriteCssClasses = "folder";
                item.Expanded=true;
                item.Id = folder.FolderID.ToString();
            })
            .Children(folder => folder.Folder1));
        mappings.For<DAL.Folder>(binding => binding
            .ItemDataBound((item, folder) =>
            {
                item.Text = folder.FolderName;
                item.SpriteCssClasses = " folder";
                item.Expanded = true;
                item.Id = folder.FolderID.ToString();
            }));
    }
    ))
</div>
```

```
<style type="text/css" scoped>
.demo-section {
    width: 200px;
}

#treeview .k-sprite ,#treeview2 .k-sprite {
    background-image: url("@Url.Content("/Content/kendo/images/coloricons-sprite.png")");
}
.rootfolder { background-position: 0 0; }
.folder { background-position: 0 -16px; }
.pdf { background-position: 0 -32px; }
.html { background-position: 0 -48px; }
.image { background-position: 0 -64px; }
.delete-link,.edit-link {
    width: 12px;
    height: 12px;
    overflow: hidden;
    display: inline-block;
    vertical-align: top;
    margin: 2px 0 0 3px;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    border-radius: 5px;
}
.delete-link{
    background: transparent url("@Url.Content("/Content/kendo/images/close.png")") no-repeat 50%
50%;
}
.edit-link{
    background: transparent url("@Url.Content("/Content/kendo/images/edit.png")") no-repeat 50%
50%;
}
</style>
```

استفاده از این TreeView ساده است ولی اگر احتیاج داشته باشیم که پس از drag&drop کردن گره‌ها آن را ذخیره کنیم چگونه باید عمل کنیم؟ این ریالسمت در خود Kendo تعبیه نشده است ، پس به صورت زیر عمل میکنیم :

پس از ساختن TreeView و اصلاح آن به شکل دلخواه لینک زیر را در ادامه اش می‌آوریم :

```
<a href="#" id="serialize">ذخیره</a>
```

سپس در تگ اسکریپت‌های خود این کد جاوا اسکریپت را برای serialize کردن تمام گره‌های TreeView مینویسیم :

```
$('#serialize').click(function () {
    serialized = serialize();
    window.location.href = "Folder/SaveMenu?serial=" + serialized + "!";
});

function serialize() {
    var tree = $("#treeview").data("kendoTreeView");
    var json = treeToJson(tree.dataSource.view());
    return JSON.stringify(json);
}

function treeToJson(nodes) {
    return $.map(nodes, function (n, i) {
        var result = { id: n.id };
        //var result = { text: n.text, id: n.id, expanded: n.expanded, checked: n.checked };
        if (n.hasChildren)
            result.items = treeToJson(n.children.view());
        return result;
    });
}
```

حال به تشریح کدها می‌پردازیم :

تابع serialize در خط اول تمام عناصر داخلی treeview را گرفته ، به صورت یک datasource قابل انقیاد درآورده و سپس datasource آن را به یک نمایش قابل تجزیه تبدیل میکند و به متد treeToJson می‌فرستد.

```
var tree = $("#treeview").data("kendoTreeView");
var json = treeToJson(tree.dataSource.view());
```

تابع treeToJson درخت را به عنوان یکسری گره گرفته و تمام عناصر آن را به فرمت json می‌برد . (قسمتی که به صورت توضیحی درآمده میتواند برای بدست آوردن تمام اطلاعات گره از جمله متن و انتخاب شدن checkbox آن و غیره مورد استفاده قرار بگیرد) در ادامه این تابع اگر گره درحال استفاده فرزندی داشته باشد به صورت بازگشتی همین تابع برای آن فراخوانده میشود.

```
var result = { id: n.id };
//var result = { text: n.text, id: n.id, expanded: n.expanded, checked: n.checked };
if (n.hasChildren)
    result.items = treeToJson(n.children.view());
```

سپس اطلاعات برگشتی که در فرمت json هستند در خط آخر serialaize به رشته تبدیل میشوند و به رویداد کلیک که از آن فراخوانده شده بود بازمیگردند.

```
return JSON.stringify(json);
```

خط آخر رویداد نیز یک Action در Controller مورد نظر را هدف قرار میدهد و رشته بدست آمده را به آن ارسال میکند و صفحه redirect میشود.

```
window.location.href = "Folder/SaveMenu?serial=" + serialized + "!";
```

رشته ای که با عنوان serialize به کنترلر ارسال میشود مانند زیر است :

```
"[{\"id\": \"2\"}, {\"id\": \"5\", \"items\": [{\"id\": \"3\"}, {\"id\": \"6\"}, {\"id\": \"7\"}]}]!"
```

این رشته مربوط به درختی به شکل زیر است :



همانطور که میبینید گره دوم که "پوشه چهارم 45" نام دارد شامل سه فرزند است که در رشته داده شده با عنوان item شناخته شده است. حال باید این رشته با برنامه نویسی سی شارپ جداسازی کرد :

```
string serialized;
Dictionary<int, int> numbers = new Dictionary<int, int>();
```

```
public ActionResult SaveMenu(string serial)
{
    var newfolders = new List<Folder>();
    serialized = serial;
    calculte_serialized(0);
    return RedirectToAction("Index");
}
```

```
void calculte_serialized(int parent)
{
    while (serialized.Length > 0)
    {
        var id_index=serialized.IndexOf("id");
        if (id_index == -1)
        {
            return;
        }
        serialized = serialized.Substring(id_index + 5);
        var quote_index = serialized.IndexOf("\"");
        var id=serialized.Substring(0, quote_index);
        numbers.Add(int.Parse(id), parent);
        serialized = serialized.Substring(quote_index);
        var condition = serialized.Substring(0,3);
        switch (condition)
        {
            case "\\",":":
                break;
            case "\\",\"":
                calculte_serialized(int.Parse(id));
                break;
            case "\\","]":
                return;
                break;
            default:
                break;
        }
    }
}
```

با گرفتن 0 کار خود را شروع میکند یعنی از گره هایی که پدر ندارند و تمام idها را همراه با پدرشان در یک دیکشنری میریزد و هرکجا که به فرزندى برخورد به صورت بازگشتی فراخوانی میشود. پس از اجرای کامل آن ما درخت را در یک دیکشنری به صورت عنصرهای مجزا در اختیار داریم که میتوانیم در پایگاه داده ذخیره کنیم.

## نظرات خوانندگان

نویسنده: امیر بختیاری  
تاریخ: ۱۳۹۲/۱۱/۲۵ ۸:۳۱

با سلام و تشکر از شما  
اگر tree دارای چک باکس باشد و بخواهیم نود هایی که چک خورده است را ذخیره کنیم چگونه عمل کنیم؟  
قابلیت drag هم نداشت مهم نیست . یک tree ثابت از دیتا بیس پر می شود و بعد گزینه هایی که تیک دارد را در جدولی دیگر  
ذخیره می کنیم

نویسنده: اسراء  
تاریخ: ۱۳۹۴/۰۷/۲۱ ۱۱:۲۶

اگر تگ name چکباکس ها یکی باشه در هنگام Post شدن یک آرایه از انتخاب شده ها با نام مشترک آنها ارسال می شود