

jqGrid از نمایش دو ساختار درختی Nested Set model و Adjacency model پشتیبانی می‌کند. توضیحات تکمیلی و پایه‌ای را در مورد این دو روش مدل سازی اطلاعات، در مطلب «[SQL Antipattern #2](#)» می‌توانید مطالعه کنید. در اینجا روش Adjacency model را به علت بیشتر مرسوم بودن آن و شباهت بسیار زیاد آن به «[مدل‌های خود ارجاع دهنده](#)» بررسی خواهیم کرد.

مدل داده‌ای Adjacency

در حالت ساختار درختی از نوع مجاورت، علاوه بر خواص اصلی یک کلاس، سه خاصیت دیگر نیز باید تعریف شوند:

```
using System;

namespace jqGrid13.Models
{
    public class BlogComment
    {
        // Other properties
        public int Id { set; get; }
        public string Body { set; get; }
        public DateTime AddDateTime { set; get; }

        // for treeGridModel: 'adjacency'
        public int? ParentId { get; set; }
        public bool IsNotExpandable { get; set; }
        public bool IsExpanded { get; set; }
    }
}
```

ParentId که سبب تولید یک مدل خود ارجاع دهنده می‌شود. IsNotExpandable به این معنا است که نود جاری آیا قرار است باز شود و فرزندی دارد یا خیر؟ اگر فرزندی ندارد باید مساوی True قرار گیرد. IsExpanded حالت پیش فرض باز بودن یا نبودن یک نود را مشخص می‌کند.

نحوه‌ی بازگشت اطلاعات درختی از سمت سرور

در نگارش فعلی jqGrid، در حالت نمایش درختی، مباحث صفحه بندی و مرتب سازی غیرفعال هستند و کدهای مرتبط با آن که در اینجا ذکر شده‌اند، فعلا تاثیری ندارند (البته با کمی تغییر در کدهای آن، می‌توان این قابلیت را هم فعال کرد. [اطلاعات بیشتر](#)). نکته‌ی مهم treeGrid، سه پارامتر دیگر هستند که از سمت کلاینت به سرور ارسال می‌شوند:

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Web.Mvc;
using jqGrid13.Models;
using JqGridHelper.DynamicSearch; // for dynamic OrderBy
using JqGridHelper.Models;
using JqGridHelper.Utils;

namespace jqGrid13.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

```

public ActionResult GetComments(JqGridRequest request, int? nodeid, int? parentid, int?
n_level)
{
    var list = BlogCommentsDataSource.LatestBlogComments;

    // در این حالت خاص فعلا در نگارش جای جی کیو گرید صفحه بندی کار نمی کند و فعال نیست و
    // محاسبات ذیل اهمیتی ندارند
    var pageIndex = request.page - 1;
    var pageSize = request.rows;
    var totalRecords = list.Count;
    var totalPages = (int)Math.Ceiling(totalRecords / (float)pageSize);

    var productsQuery = list.AsQueryable();

    if (nodeid == null)
    {
        productsQuery = productsQuery.Where(x => x.ParentId == null);
    }
    else
    {
        productsQuery = productsQuery.Where(x => x.ParentId == nodeid.Value);
    }

    var products = productsQuery.OrderBy(request.sidx + " " + request.sord)
        .Skip(pageIndex * pageSize)
        .Take(pageSize)
        .ToList();

    var newLevel = n_level == null ? 0 : n_level.Value + 1;
    var productsData = new JqGridData
    {
        Total = totalPages,
        Page = request.page,
        Records = totalRecords,
        Rows = (products.Select(comment => new JqGridRowData
        {
            Id = comment.Id,
            RowCells = new List<object>
            {
                comment.Id,
                comment.Body,
                comment.AddDateTime.ToPersianDate(),
                // اطلاعات خاص نمایش درختی به ترتیب
                newLevel,
                comment.ParentId == null ? "" :
comment.ParentId.Value.ToString(CultureInfo.InvariantCulture),
                comment.IsNotExpandable,
                comment.IsExpanded
            }
        })).ToList()
    };
    return Json(productsData, JsonRequestBehavior.AllowGet);
}
}
}

```

nodeid اگر نال بود، یعنی کل اطلاعات ریشه ها (مواردی که parentId مساوی نال دارند)، باید واکنشی شوند. اگر nodeid مقدار داشت، یعنی فرزند نود جاری قرار است بازگشت داده شود.

n_level مقدار جلو رفتگی نمایش اطلاعات یک نود را مشخص می کند. در اینجا چون با کلیک بر روی هر نود، فرزند آن از سرور واکنشی می شود و lazy loading برقرار است، بازگشت مقدار n_level دریافتی از کلاینت به علاوه یک، کافی است. اگر نیاز است تمام نودها باز شده نمایش داده شوند، این مورد را باید به صورت دستی محاسبه کرده و در مدل BlogComment پیش بینی کنید. در نهایت آرایه ای از خواص مدنظر به همراه 4 خاصیت ساختار درختی باید به ترتیب بازگشت داده شوند.

آزمایش سیزدهم			
شماره	نظر	تاریخ	
1 ▼	نظر ریشه 1	1393/05/09	1
2 ▼	پاسخ 1 به ریشه 1	1393/05/09	1
3 ○	پاسخ 1 به پاسخ 2	1393/05/09	1
4 ▼	پاسخ 2 به ریشه 1	1393/05/09	2
5 ○	پاسخ 1 به پاسخ 4	1393/05/09	1
6 ▼	پاسخ 3 به ریشه 1	1393/05/09	3
7 ○	پاسخ 1 به پاسخ 6	1393/05/09	1
8 ▼	پاسخ 4 به ریشه 1	1393/05/09	4
9 ○	پاسخ 1 به پاسخ 8	1393/05/09	1
10 ▼	پاسخ 5 به ریشه 1	1393/05/09	5
11 ○	پاسخ 1 به پاسخ 10	1393/05/09	1
12 ◀	نظر ریشه 12	1393/05/09	2
23 ◀	نظر ریشه 23	1393/05/09	3
34 ◀	نظر ریشه 34	1393/05/09	4
45 ◀	نظر ریشه 45	1393/05/09	5
56 ◀	نظر ریشه 56	1393/05/09	6
67 ◀	نظر ریشه 67	1393/05/09	7

فعال سازی سمت کاربر treeGrid

برای فعال سازی سمت کاربر نمایش درختی اطلاعات، باید سه خاصیت ذیل تنظیم شوند:

```
$('#list').jqGrid({
  caption: "آزمایش سیزدهم",
  // مانند قبل ....
  treeGrid: true,
  treeGridModel: 'adjacency',
  ExpandColumn: '@(StronglyTyped.PropertyName<BlogComment>(x => x.Body))',
}).jqGrid('gridResize', { minWidth: 400 });
```

تنظیم treeGrid: true سبب فعال سازی treeGrid می‌شود. توسط treeGridModel حالات Adjacency و Nested Set model قابل تنظیم هستند و ExpandColumn نام ستونی را مشخص می‌کند که قرار است فرزندان آن نمایش داده شوند.

یک نکته‌ی تکمیلی

اگر می‌خواهید دقیقاً به شکل زیر برسید:

نظر
▼ نظر ریشه 1
▼ پاسخ 1 به ریشه 1
● پاسخ 1 به پاسخ 2
▼ پاسخ 2 به ریشه 1
● پاسخ 1 به پاسخ 4
▼ پاسخ 3 به ریشه 1
● پاسخ 1 به پاسخ 6
▼ پاسخ 4 به ریشه 1
● پاسخ 1 به پاسخ 8
▼ پاسخ 5 به ریشه 1
● پاسخ 1 به پاسخ 10

تنظیم rownumbers: true را گرید را حذف کنید. همچنین ستون Id را نیز با تنظیم‌های hidden:true, key: true مخفی نمائید (در تعاریف colModel).

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[jqGrid13.zip](#)

برای مطالعه بیشتر

[Tree Grid](#)

[Nested Set Model](#)

[Adjacency Model](#)

نظرات خوانندگان

نویسنده: ابوالفضل رجب پور
تاریخ: ۱۳۹۳/۰۵/۲۲ ۱۲:۳۲

- آیا در این ساختار سلسله مراتبی adjacency می‌شود inline edit انجام داد؟
در صفحه‌ی ویکی پلاگین، این خط در قسمت محدودیت‌های adjacency نوشته شده و سپس خط خورده.

آیا راه حلی هست؟

- در ادامه سوال بالا، فرض بفرمایید یک سری دسته بندی و یک سری محصول داریم. می‌خواهیم این دسته بندی‌ها با عمق نامحدود را باز کنیم و وقتی به یک محصولی رسید، بتواند inline edit کند. این مدل رو با کدام حالت باید پیاده کرد؟ گروه بندی grouping یا سلسله مراتبی treemode؟ نامحدود بودن عمق گروه‌های محصولات را هم در نظر بگیرید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۲ ۱۲:۵۱

چند مثال در این زمینه:

[Functionality -> Add tree node](#) (مثال رسمی)

[Insert a row](#) (روی صفحه کلیک راست کرده و سورس آن را مطالعه کنید)

[Add new rows to jqGrid Treegrid model](#) (در استک اورفلو جستجو کنید، مطالب خوبی در مورد jqGrid دارد. تا امروز 8000

[سؤال مرتبط](#) دارد. امکان ندارد در زمینه‌ی jqGrid مشکلی داشته باشید و در آنجا مطرح نشده باشد)

نویسنده: رسولی
تاریخ: ۱۳۹۳/۰۹/۰۹ ۱۰:۴۸

با سلام؛ در این ساختار درختی از Parent برای تشخیص ریشه و فرزندان استفاده شده است. سوالم اینه که امکانش هست که ساختار درختی بین دو مدل و توسط کلید خارجی صورت گیرد؟ به عنوان مثال میخام گریدی داشته باشم که شامل لیست استان‌ها باشد و با کلیک بر روی هر استان لیست شهرهای آن نشان داده شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۹ ۱۱:۱۴

- مدل توضیح داده شده در اینجا Adjacency model است و شباهت زیادی به «[مدل‌های خود ارجاع دهنده](#)» دارد. طراحی خودتان را بر اساس مطلب یاد شده انجام دهید و یا نداشت نهایی اطلاعات خودتان را تبدیل کنید به این حالت. مهم نیست ساختار اصلی بانک اطلاعاتی شما به چه صورتی است. همینقدر که خروجی کوئری آن Adjacency model باشد (شبیه به ساختار کلاس BlogComment مطلب فوق)، با توضیحات فوق سازگار خواهد بود. اگر مستقیماً SQL می‌نویسید، در مطلب «[SQL Antipattern #2](#)» کوئری‌های آن موجود است. اگر با LINQ و EF کار می‌کنید، توضیحات مطلب «[مباحث تکمیلی مدل‌های خود ارجاع دهنده در EF](#)» را پیگیری کنید.

+ امکان اتصال دو جدول با کلید خارجی نیز در اینجا وجود دارد:

```
public ActionResult GetComments(JqGridRequest request, int? nodeId, int? parentid, int? n_level)
```

در امضای متد فوق که در بحث مطرح شده، node id تعیین کننده‌ی واکنشی از parent id است. اگر node id نال بود، یعنی نمایش بار اول لیست (نمایش لیست استان‌ها):

```
if (nodeid == null)
{
    productsQuery = productsQuery.Where(x => x.ParentId == null);
}
```

اگر نال نبود (درخواست واکشی اطلاعات استان بر اساس node id آن)، یعنی روی یک نود کلیک شده‌است. در اینجا فیلد parent id می‌تواند به عنوان کلید خارجی که به جدولی دیگر اشاره می‌کند نیز تفسیر و جایگزین شود:

```
else
{
    // آی دی یک گره می‌تواند کلید خارجی یک جدول دیگر باشد
    productsQuery = productsQuery.Where(x => x.ParentId == nodeId.Value);
}
```

برای این متد نهایتاً مهم نیست که productsQuery به چه نحوی تهیه می‌شود. مهم نیست که از چند جدول مختلف حاصل می‌شود. فقط مقادیر نهایی آن مهم است.

نویسنده: رسولی
تاریخ: ۱۶:۵۱ ۱۳۹۳/۰۹/۱۰

سلام

امکان افزودن، ویرایش و حذف بصورت پویا در jqGrid درختی وجود دارد؟
من لینک ویرایش و حذف را به هر رکورد اضافه کردم ولی گرید بهم ریخته شد.

نویسنده: وحید نصیری
تاریخ: ۱۸:۲۴ ۱۳۹۳/۰۹/۱۰

- مراجعه کنید به مجموعه مثال‌های آن [در اینجا](#). مواردی مانند «Functionality -> Add tree node» در آن مستند شده‌اند.
- یک مثال دیگر [در اینجا](#)
- مجموعه پرسش و پاسخ‌های [stack overflow](#) هم در این زمینه مفید است.