


در این قسمت قصد داریم با نحوه پیاده سازی امتیاز دهی ستاره‌ای به مطالب، که نمونه‌ای از آن‌را در سایت جاری در قسمت‌های مختلف آن مشاهده می‌کنید، آشنا شویم.

مطلب 29

عنوان 29

متن ... متن ... متن 29

امتیاز 3.50 از 5 توسط 30 نفر 

مدل برنامه

در ابتدای کار نیاز است تا ساختاری را جهت ارائه لیستی از مطالب که دارای گزینه امتیاز دهی می‌باشند، تهیه کنیم:

```
namespace jQueryMvcSample03.Models
{
    public class BlogPost
    {
        public int Id { set; get; }
        public string Title { set; get; }
        public string Body { set; get; }

        /// <summary>
        /// اطلاعات رای گیری یک مطلب به صورت یک خاصیت تو در تو یا پیچیده
        /// </summary>
        public Rating Rating { set; get; }

        public BlogPost()
        {
            Rating = new Rating();
        }
    }
}
```

```
namespace jQueryMvcSample03.Models
{
    ///[ComplexType]
    public class Rating
    {
        public double? TotalRating { get; set; }
        public int? TotalRaters { get; set; }
        public double? AverageRating { get; set; }
    }
}
```

اگر با EF Code first آشنا باشید، خاصیت Rating تعریف شده در اینجا می‌تواند از نوع ComplexType تعریف شود که شامل جمع امتیازهای داده شده، تعداد کل رای دهنده‌ها و همچنین میانگین امتیازهای حاصل است.

منبع داده فرضی برنامه

```

using System.Collections.Generic;
using System.Linq;
using jQueryMvcSample03.Models;

namespace jQueryMvcSample03.DataSource
{
    /// <summary>
    /// منبع داده فرضی
    /// </summary>
    public static class BlogPostDataSource
    {
        private static IList<BlogPost> _cachedItems;
        /// <summary>
        /// با توجه به استاتیک بودن سازنده کلاس، تهیه کش، پیش از سایر فراخوانی‌ها صورت خواهد گرفت
        /// باید دقت داشت که این فقط یک مثال است و چنین کشی به معنای
        /// تهیه یک لیست برای تمام کاربران سایت است
        /// </summary>
        static BlogPostDataSource()
        {
            _cachedItems = createBlogPostsInMemoryDataSource();
        }

        /// <summary>
        /// هدف صرفاً تهیه یک منبع داده آزمایشی ساده تشکیل شده در حافظه است
        /// </summary>
        private static IList<BlogPost> createBlogPostsInMemoryDataSource()
        {
            var results = new List<BlogPost>();
            for (int i = 1; i < 30; i++)
            {
                results.Add(new BlogPost { Id = i, Title = "عنوان " + i, Body = "متن ... متن ... متن "
+ i, Rating = new Rating { TotalRaters = i + 1, AverageRating = 3.5 } });
            }
            return results;
        }

        /// <summary>
        /// پارامترهای شماره صفحه و تعداد رکورد به ازای یک صفحه بندی نیاز هستند
        /// شماره صفحه از یک شروع می‌شود
        /// </summary>
        public static IList<BlogPost> GetLatestBlogPosts(int pageNumber, int recordsPerPage = 4)
        {
            var skipRecords = pageNumber * recordsPerPage;
            return _cachedItems
                .OrderByDescending(x => x.Id)
                .Skip(skipRecords)
                .Take(recordsPerPage)
                .ToList();
        }
    }
}

```

در این مثال نیز از یک منبع داده فرضی تشکیل شده در حافظه استفاده خواهیم کرد تا امکان اجرای پروژه پیوستی را بدون نیاز به بانک اطلاعاتی خاصی و بدون نیاز به مقدمات برپایی آن، به سادگی داشته باشید.

در این منبع داده ابتدا لیستی از مطالب تهیه شده و سپس کش می‌شوند. در ادامه توسط متد `GetLatestBlogPosts` بازه‌ای از این اطلاعات قابل بازیابی خواهند بود که برای استفاده در حالات صفحه بندی اطلاعات بهینه سازی شده است.

آشنایی با طراحی افزونه jQuery Star Rating

افزودن CSS نمایش امتیازها در ذیل هر مطلب

```

/* star rating system */
.post_rating
{
    direction: ltr;
}

```

```

}
.rating
{
text-indent: -99999px;
overflow: hidden;
background-repeat: no-repeat;
display: inline-block;
width: 8px;
height: 16px;
}
.rating.stars
{
background-image: url('Images/star_rating.png');
}
.rating.stars.active
{
cursor: pointer;
}
.star-left_off
{
background-position: -0px -0px;
}
.star-left_on
{
background-position: -16px -0px;
}
.star-right_off
{
background-position: -8px -0px;
}
.star-right_on
{
background-position: -24px -0px;
}

```

برای نمایش ستاره‌ها و کار با تصویر Images/star_rating.png (که در پروژه پیوست قرار دارد) ابتدا نیاز است CSS فوق را به پروژه خود اضافه نمائید.

افزودن افزونه jQuery Star rating

```

// 
(function ($) {
$.fn.StarRating = function (options) {
var defaults = {
ratingStarsSpan: '.rating.stars',
postInfoUrl: '/',
loginUrl: '/login',
errorHandler: null,
completeHandler: null,
onlyOneTimeHandler: null
};
var options = $.extend(defaults, options);

return this.each(function () {
var ratingStars = $(this);

$(ratingStars).unbind('mouseover');
$(ratingStars).mouseover(function () {
var span = $(this).parent("span");
var newRating = $(this).attr("value");
setRating(span, newRating);
});

$(ratingStars).unbind('mouseout');
$(ratingStars).mouseout(function () {
var span = $(this).parent("span");
var rating = span.attr("rating");
setRating(span, rating);
});

$(ratingStars).unbind('click');
$(ratingStars).click(function () {
var span = $(this).parent("span");
var newRating = $(this).attr("value");
</pre>
</div>
<div data-bbox="494 956 530 971" data-label="Page-Footer">۳/۱۱</div>
```

```

        var text = span.children("span");
        var pID = span.attr("post");
        var type = span.attr("sectiontype");
        postData({ postID: pID, rating: newRating, sectionType: type });
        span.attr("rating", newRating);
        setRating(span, newRating);
    });

    function setRating(span, rating) {
        span.find(options.ratingStarsSpan).each(function () {
            var value = parseFloat($(this).attr("value"));
            var imgSrc = $(this).attr("class");
            if (value <= rating)
                $(this).attr("class", imgSrc.replace("_off", "_on"));
            else
                $(this).attr("class", imgSrc.replace("_on", "_off"));
        });
    }

    function postData(dataJsonArray) {
        $.ajax({
            type: "POST",
            url: options.postInfoUrl,
            data: JSON.stringify(dataJsonArray),
            contentType: "application/json; charset=utf-8",
            dataType: "json",
            complete: function (xhr, status) {
                var data = xhr.responseText;
                if (xhr.status == 403) {
                    window.location = options.loginUrl;
                }
                else if (status === 'error' || !data) {
                    if (options.errorHandler)
                        options.errorHandler(this);
                }
                else if (data == "nok") {
                    if (options.onlyOneTimeHandler)
                        options.onlyOneTimeHandler(this);
                }
                else {
                    if (options.completeHandler)
                        options.completeHandler(this);
                }
            }
        });
    }
})(jQuery);
// ]]>

```

اطلاعات فوق، فایل jquery.StarRating.js را تشکیل می‌دهند که باید به پروژه اضافه گردند. کاری که این افزونه انجام می‌دهد ردیابی حرکت ماوس بر روی ستاره‌های نمایش داده شده و سپس ارسال سه پارامتر ذیل به اکشن متدی که توسط پارامتر postInfoUrl مشخص می‌گردد، پس از کلیک کاربر می‌باشد:

```
{ postID: pID, rating: newRating, sectionType: type }
```

همانطور که ملاحظه می‌کنید به ازای هر قطعه رای گیری که به صفحه اضافه می‌شود، Id مطلب، رای داده شده و نام قسمت جاری، به اکشن متدی خاص ارسال خواهند گردید. sectionType از این جهت اضافه گردیده است تا بتوانید با بیش از یک جدول کار کنید و از این افزونه در قسمت‌های مختلف سایت به سادگی بتوانید استفاده نمایید. در اینجا از errorHandler برای نمایش خطاها، از completeHandler برای نمایش تشکر به کاربر و از onlyOneTimeHandler برای نمایش اخطار مثلاً «یکبار بیشتر مجاز نیستید به ازای یک مطلب رای دهید»، می‌توان استفاده کرد.

بنابراین تا اینجا فایل layout برنامه تقریباً چنین مداخلی را خواهد داشت:

```

<head>
<title>@ViewBag.Title</title>
<link href="@Url.Content("Content/starRating.css")" rel="stylesheet" type="text/css" />
<link href="@Url.Content("Content/Site.css")" rel="stylesheet" type="text/css" />

```

```
<script src="@Url.Content("~/Scripts/jquery-1.9.1.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.unobtrusive-ajax.min.js")"
type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.StarRating.js")" type="text/javascript"></script>
@RenderSection("JavaScript", required: false)
</head>
```

طراحی یک HTML helper برای نمایش ستاره‌های امتیاز دهی

ابتدا پوشه استاندارد app_code را به پروژه اضافه کرده و سپس فایلی را به نام StarRatingHelper.cshtml با محتوای ذیل به آن اضافه نمائید:

```
@using System.Globalization
@helper AddStarRating(int postId,
                    double? average = 0, int? postRatingsCount = 0, string type = "BlogPost",
                    string tooltip = "لطفا جهت رای دادن کلیک نمائید")
{
    string actIt = "active ";
    if (!average.HasValue) { average = 0; }
    if (!postRatingsCount.HasValue) { postRatingsCount = 0; }

    <span class='postRating' rating='@average' post='@postId' title='@tooltip' sectiontype='@type'>
        @for (double i = .5; i <= 5.0; i = i + .5)
        {
            string left;
            if (i <= average)
            {
                left = (i * 2) % 2 == 1 ? "left_on" : "right_on";
            }
            else
            {
                left = (i * 2) % 2 == 1 ? "left_off" : "right_off";
            }
            <span class='rating stars @(actIt)star-@left' value='@i'></span>
        }
        &nbsp;
        @if (postRatingsCount > 0)
        {
            var ratingInfo = string.Format(CultureInfo.InvariantCulture, "{1} از 5 توسط {0:0.00} امتیاز
نفر", average, postRatingsCount);
            <span>@ratingInfo</span>
        }
        else
        {
            <span></span>
        }
    </span>
}
```

از این Html helper برای تشکیل ساختار نمایش قطعه امتیاز دهی به یک مطلب استفاده خواهیم کرد که توسط افزونه جی‌کوئری فوق ردیابی می‌شود.

کنترلر ذخیره سازی اطلاعات دریافتی برنامه

```
using System.Web.Mvc;
using System.Web.UI;
using jQueryMvcSample03.DataSource;
using jQueryMvcSample03.Security;

namespace jQueryMvcSample03.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
        }
    }
}
```

```

{
    var postsList = BlogPostDataSource.GetLatestBlogPosts(pageNumber: 0);
    return View(postsList); //نمایش صفحه اصلی
}

[HttpPost]
[AjaxOnly]
[OutputCache(Location = OutputCacheLocation.None, NoStore = true)]
public ActionResult SaveRatings(int? postId, double? rating, string sectionType)
{
    if (postId == null || rating == null || string.IsNullOrEmpty(sectionType))
        return Content(null); //اعلام بروز خطا

    if (!this.HttpContext.CanUserVoteBasedOnCookies(postId.Value, sectionType))
        return Content("nok"); //اعلام فقط یکبار مجاز هستید رای دهید

    switch (sectionType) //قسمت‌های مختلف سایت که در جداول مختلفی قرار دارند نیز می‌توانند گزینه امتیاز دهی داشته باشند
    {
        case "BlogPost":
            //الان شماره مطلب و رای ارسالی را داریم که می‌توان نسبت به ذخیره آن اقدام کرد
            //مثلا
            //_blogPostsService.SaveRating(postId.Value, rating.Value);
            break;

            //سایر قسمت‌های دیگر سایت ...

        default:
            return Content(null); //اعلام بروز خطا
    }

    return Content("ok"); //اعلام موفقیت آمیز بودن ثبت اطلاعات
}

[HttpGet]
public ActionResult Post(int? id)
{
    if (id == null)
        return Redirect("/");

    //todo: show the content here
    return Content("Post " + id.Value);
}
}

```

در اینجا کنترلری را که کار پردازش کلیک کاربر را بر روی امتیازی خاص انجام می‌دهد، ملاحظه می‌کنید. امضای اکشن متد SaveRatings دقیقا بر اساس سه پارامتر ارسالی توسط jquery.StarRating.js که پیشتر توضیح داده شد، تعیین گردیده است. در این متد ابتدا بررسی می‌شود که آیا اطلاعاتی دریافت شده است یا خیر. اگر خیر، null را بازگشت خواهد داد. سپس توسط متد CanUserVoteBasedOnCookies بررسی می‌شود که آیا کاربر می‌تواند (خصوصا مجددا) رای دهد یا خیر. این افزونه برای رای دهی کاربران وارد نشده به سیستم نیز مناسب است. به همین جهت از کوکی‌ها برای ثبت اطلاعات رای دادن کاربران استفاده گردیده است. پیاده سازی متد CanUserVoteBasedOnCookies را در ادامه ملاحظه خواهید نمود. در ادامه در متد SaveRatings، یک switch تشکیل شده است تا بر اساس نام قسمت مرتبط به رای گیری، اطلاعات را بتوان به سرویس خاصی در برنامه هدایت کرد. مثلا اطلاعات قسمت مطالب به سرویس مطالب و قسمت نظرات به سرویس نظرات هدایت شوند.

متدهایی برای کار با کوکی‌ها در ASP.NET MVC

```

using System;
using System.Web;

namespace jQueryMvcSample03.Security
{
    public static class CookieHelper
    {
        public static bool CanUserVoteBasedOnCookies(this HttpContextBase httpContext, int postId, string sectionType)
        {

```

```

        string key = sectionType + "-" + postId;
        var value = httpContext.GetCookieValue(key);
        if (string.IsNullOrEmpty(value))
        {
            httpContext.AddCookie(key, key);
            return true;
        }
        return false;
    }

    public static void AddCookie(this HttpContextBase httpContextBase, string cookieName, string
value)
    {
        httpContextBase.AddCookie(cookieName, value, DateTime.Now.AddDays(30));
    }

    public static void AddCookie(this HttpContextBase httpContextBase, string cookieName, string
value, DateTime expires)
    {
        var cookie = new HttpCookie(cookieName)
        {
            Expires = expires,
            Value = httpContextBase.Server.UrlEncode(value) // For Cookies and Unicode characters
        };
        httpContextBase.Response.Cookies.Add(cookie);
    }

    public static string GetCookieValue(this HttpContextBase httpContext, string cookieName)
    {
        var cookie = httpContext.Request.Cookies[cookieName];
        if (cookie == null)
            return string.Empty; //cookie doesn't exist

        // For Cookies and Unicode characters
        return httpContext.Server.UrlDecode(cookie.Value);
    }
}

```

در اینجا یک سری متد الحاقی را ملاحظه می‌کنید که برای ثبت اطلاعات رای داده شده یک کاربر بر اساس Id مطلب و نام قسمت متناظر با آن در یک کوکی طراحی شده‌اند. بدیهی است اگر تمام قسمت‌های برنامه شما محافظت شده هستند و کاربران حتما نیاز است ابتدا به سیستم لاگین نمایند، می‌توانید این قسمت را حذف کرده و اطلاعات postId و SectionType را به ازای هر کاربر، جداگانه در بانک اطلاعاتی ثبت و بازیابی نمائید (دقیق‌ترین حالت ممکن؛ البته برای سیستمی بسته که حتما تمام قسمت‌های آن نیاز به اعتبار سنجی دارند).

پیشنهادی در مورد نحوه ذخیره سازی اطلاعات دریافتی

```

using jQueryMvcSample03.Models;

namespace jQueryMvcSample03.DataSource
{
    public interface IBlogPostsService
    {
        void SaveRating(int postId, double rating);
    }

    public class SampleService : IBlogPostsService
    {
        /// <summary>
        /// یک نمونه از متد ذخیره سازی اطلاعات پیشنهادی
        /// فقط برای ابژه گرفتن
        /// بدیهی است محل قرارگیری اصلی آن در لایه سرویس برنامه شما خواهد بود
        /// </summary>
        public void SaveRating(int postId, double rating)
        {
            BlogPost post = null;
            //post = _blogCtx.Find(postId); // فیلدهای آن تنظیم
            if (post == null) return;

            if (!post.Rating.TotalRaters.HasValue) post.Rating.TotalRaters = 0;
            if (!post.Rating.TotalRating.HasValue) post.Rating.TotalRating = 0;
        }
    }
}

```

می‌شوند

```

        if (!post.Rating.AverageRating.HasValue) post.Rating.AverageRating = 0;
        post.Rating.TotalRaters++;
        post.Rating.TotalRating += rating;
        post.Rating.AverageRating = post.Rating.TotalRating / post.Rating.TotalRaters;
        // todo: call save changes at the end.
    }
}
}

```

همانطور که عنوان شد، سه داده Id مطلب، رای داده شده و نام قسمت متناظر به اکشن متد ارسال می‌شود. از نام قسمت، برای انتخاب سرویس ذخیره سازی اطلاعات استفاده خواهیم کرد. این سرویس می‌تواند شامل متدی به نام SaveRating، همانند کدهای فوق باشد که Id مطلب و عدد رای حاصل به آن ارسال می‌گردند. ابتدا بر اساس این Id، مطلب متناظر یافت شده و سپس اطلاعات Rating آن به روز خواهد شد. در پایان هم ذخیره سازی اطلاعات باید صورت گیرد.

Viewهای برنامه

قسمت پایانی کار ما در اینجا تهیه دو View است:

الف) یک Partial view که لیست مطالب را به همراه گزینه رای دهی به آن‌ها رندر می‌کند.

ب) View کاملی که از این Partial View استفاده کرده و همچنین افزونه jquery.StarRating.js را فراخوانی می‌کند.

```

@using System.Text.RegularExpressions
@model IList<jQueryMvcSample03.Models.BlogPost>
<ul>
    @foreach (var item in Model)
    {
        <li>
            <fieldset>
                <legend>مطلب @item.Id</legend>
                <h5>
                    @Html.ActionLink(linkText: item.Title,
                                    actionName: "Post",
                                    controllerName: "Home",
                                    routeValues: new { id = item.Id },
                                    htmlAttributes: null)
                </h5>
                @item.Body
                <div class="post_rating">
                    @Html.Raw(Regex.Replace(@StarRatingHelper.AddStarRating(item.Id,
item.Rating.AverageRating, item.Rating.TotalRaters, "BlogPost").ToHtmlString(), @">\s+<", "><"))
                </div>
            </fieldset>
        </li>
    }
</ul>

```

کدهای ItemsList.cshtml را در اینجا ملاحظه می‌کند که در آن نحوه فراخوانی متد کمکی StarRatingHelper.AddStarRating ذکر شده است.

اگر به کدهای آن دقت کنید از Regex.Replace برای حذف فاصله‌های خالی و خطوط جدید بین تگ‌ها استفاده گردیده است. اگر اینکار انجام نشود، نیمه‌های ستاره‌های نمایش داده شده، با فاصله از یکدیگر رندر می‌شوند که صورت خوشایندی ندارد.

و نهایتاً View ایی که از این اطلاعات استفاده می‌کنید ساختار زیر را خواهد داشت:

```

@model IList<jQueryMvcSample03.Models.BlogPost>
@{
    ViewBag.Title = "Index";
    var postInfoUrl = Url.Action(actionName: "SaveRatings", controllerName: "Home");
}
<h2>
سیستم امتیاز دهی</h2>
@{ Html.RenderPartial("_ItemsList", Model); }
@section JavaScript
{
    <script type="text/javascript">

```



```
$(document).ready(function () {  
    $(".rating.stars.active").StarRating({  
        ratingStarsSpan: '.rating.stars',  
        postInfoUrl: '@postInfoUrl',  
        loginUrl: '/login',  
        errorHandler: function () {  
            alert('خطایی رخ داده است');  
        },  
        completeHandler: function () {  
            alert('با تشکر! رای شما با موفقیت ثبت شد');  
        },  
        onlyOneTimeHandler: function () {  
            alert('فقط یکبار می‌توانید به آرای هر مطلب رای دهید');  
        }  
    });  
});  
</script>  
}
```

در این View لیستی از مطالب دریافت و به partial view طراحی شده برای نمایش ارسال می‌شود. سپس افزونه StarRating نیز تنظیم و به صفحه اضافه خواهد گردید. نکته مهم آن تعیین صحیح اکشن متدی است که قرار است اطلاعات را دریافت کند و نحوه مقدار دهی آن را توسط متغیر postInfoUrl مشاهده می‌کنید.

دریافت کدها و پروژه کامل این قسمت

[jQueryMvcSample03.zip](#)

نظرات خوانندگان

نویسنده:

فرزین بیاتی

تاریخ:

۱۸:۱۲ ۱۳۹۲/۱۲/۱۸

سلام، برای من پیغام "خطایی رخ داده است" نمایش داده می‌شود و هیچگاه اکشن SaveRating صدا زده نمی‌شود، البته من از StarRatingHelper در یک view به جای partial view استفاده کردم (این view فقط یک مطلب از سایت من رو نشان میده و من می‌خواهم کاربر بتونه به مطلب امتیاز بده) نمیدونم مشکل از کجاست؟ مثال شما رو دانلود کردم و به درستی کار میکنه، ممنون میشم راهنمایی بفرمایید.

نویسنده:

وحید نصیری

تاریخ:

۱۸:۳۸ ۱۳۹۲/۱۲/۱۸

احتمال دارد مسیرها را صحیح تنظیم نکردید. مطلب زیر در مورد نحوه‌ی دیباگ برنامه‌های Ajax ایی مفید است:

« [نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی بر jQuery](#) »

نویسنده:

محمد صاحب

تاریخ:

۹:۳۳ ۱۳۹۲/۱۲/۱۹

آیا این دو اسکریپت با هم تداخل دارن؟ جدا جدا اجرا میشن ولی با هم سیستم امتیاز دهی از کار میفته یا نحوه استفاده‌ی من غلطه؟

```
$(document).ready(function () {
    $("#moreInfoButton").InfiniteScroll({
        moreInfoDiv: '#MoreInfoDiv',
        progressDiv: '#ProgressDiv',
        loadInfoUrl: '/Media/PagedIndex',
        loginUrl: '/login',
        errorHandler: function () { alert('خطایی رخ داده است'); },
        completeHandler: function () { },
        noMoreInfoHandler: function () { alert('اطلاعات بیشتری یافت نشد'); }
    });

    $(".rating.stars.active").StarRating({
        ratingStarsSpan: '.rating.stars',
        postInfoUrl: '/Media/SaveRatings',
        loginUrl: '/login',
        errorHandler: function () { alert('خطایی رخ داده است'); },
        completeHandler: function () { alert('با تشکر! رای شما با موفقیت ثبت شد'); },
        onlyOneTimeHandler: function () { alert('فقط یکبار می‌توانید به ازای هر مطلب رای دهید'); }
    });
});
```

با تشکر

نویسنده:

وحید نصیری

تاریخ:

۹:۴۹ ۱۳۹۲/۱۲/۱۹

تداخلی ندارند. فقط محتوایی که از طریق Ajax به صفحه اضافه می‌شود، باید [توسط متد on](#) (پیشتر live بود) تحت نظر قرار گرفته شود و گرنه ندید گرفته می‌شود؛ چون در زمان document.ready وجود خارجی نداشته‌است.

نویسنده:

ایمان اسلامی

تاریخ:

۱۵:۵۰ ۱۳۹۳/۰۲/۱۸

آیا میتوان از این مثال در asp.net web form هم استفاده کرد؟

نویسنده: وحید طجری
تاریخ: ۱۸:۵۹ ۱۳۹۳/۰۶/۰۵

بعد از آپدیت قسمت نظرات با ajax اسکریپت از کار میفته. توی این مثال از متد On چطوری باید استفاده کنیم؟ توی فایل jquery.StarRating باید تغییرات بدیم یا توی اسکریپتی که داخل صفحه مینویسیم؟

نویسنده: وحید نصیری
تاریخ: ۱:۷ ۱۳۹۳/۰۶/۰۶

الزاما نیازی به استفاده از متد On نیست. در رویدادگردان complete عملیات ajax ایی، متد StarRating را دوباره فراخوانی کنید.