

ASP.NET MVC به همراه HtmlHelper توکاری جهت نمایش یک ChekBoxList نیست؛ اما سیستم Model binder آن، این نوع کنترل‌ها را به خوبی پشتیبانی می‌کند. برای مثال، یک پروژه جدید خالی ASP.NET MVC را آغاز کنید. سپس یک کنترلر Home جدید را نیز به آن اضافه کنید. در ادامه، برای متد Index آن، یک View خالی را ایجاد نمایید. سپس محتوای این View را به نحو زیر تغییر دهید:

```
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
@using (Html.BeginForm())
{
    <input type='checkbox' name='Result' value='value1' />
    <input type='checkbox' name='Result' value='value2' />
    <input type='checkbox' name='Result' value='value3' />
    <input type="submit" value="submit" />
}
```

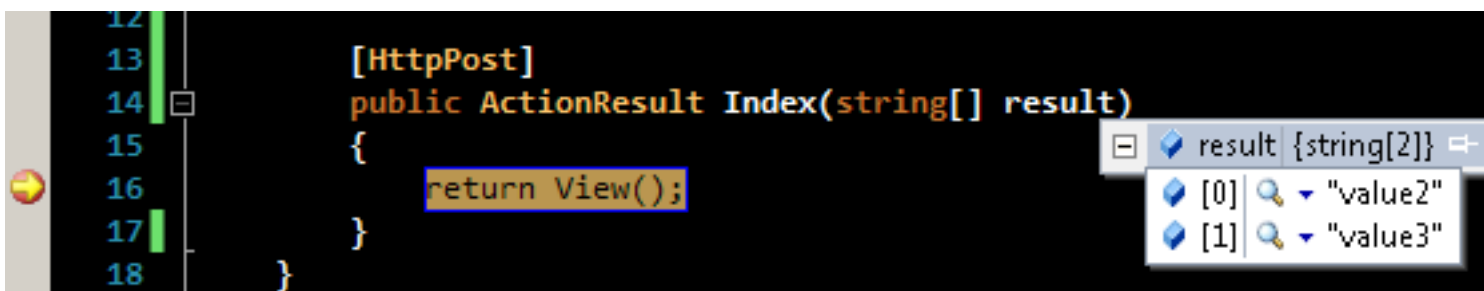
و کنترلر Home را نیز مطابق کدهای زیر ویرایش کنید:

```
using System.Web.Mvc;

namespace MvcApplication21.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(string[] result)
        {
            return View();
        }
    }
}
```

یک breakpoint را در تابع Index دوم که آرایه‌ای را دریافت می‌کند، قرار دهید. سپس برنامه را اجرا کرده، تعدادی از checkboxها را انتخاب و فرم نمایش داده شده را به سرور ارسال کنید:



بله. همانطور که ملاحظه می‌کنید، تمام عناصر ارسالی انتخاب شده که دارای نامی مشابه بوده‌اند، به یک آرایه قابل بایند هستند و سیستم model binder می‌داند که چگونه باید این اطلاعات را دریافت و پردازش کند. از این مقدمه می‌توان به عنوان پایه و اساس نوشتن یک HtmlHelper سفارشی CheckBoxList استفاده کرد. برای این منظور یک پوشه جدید را به نام app_code، به ریشه پروژه اضافه نمائید. سپس یک فایل خالی را به نام Helpers.cshtml نیز به آن اضافه کنید. محتوای این فایل را به نحو زیر تغییر دهید:

```
@helper CheckBoxList(string name, List<System.Web.Mvc.SelectListItem> items)
{
    <div class="checkboxList">
        @foreach (var item in items)
        {
            @item.Text
            <input type="checkbox" name="@name"
                value="@item.Value"
                @if (item.Selected) { <text>checked="checked"</text> }
            />
            <br />
        }
    </div>
}
```

و برای استفاده از آن، کنترلر Home را مطابق کدهای زیر ویرایش کنید:

```
using System.Collections.Generic;
using System.Web.Mvc;

namespace MvcApplication21.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            ViewBag.Tags = new List<SelectListItem>
            {
                new SelectListItem { Text = "Item1", Value = "Val1", Selected = false },
                new SelectListItem { Text = "Item2", Value = "Val2", Selected = false },
                new SelectListItem { Text = "Item3", Value = "Val3", Selected = true }
            };
            return View();
        }

        [HttpPost]
        public ActionResult GetTags(string[] tags)
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(string[] result)
        {
            return View();
        }
    }
}
```

```

    }
}

```

و در این حالت View برنامه به شکل زیر درخواهد آمد:

```

@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
@using (Html.BeginForm())
{
    <input type='checkbox' name='Result' value='value1' />
    <input type='checkbox' name='Result' value='value2' />
    <input type='checkbox' name='Result' value='value3' />
    <input type="submit" value="submit" />
}

@using (Html.BeginForm(actionName: "GetTags", controllerName: "Home"))
{
    @Helpers.CheckBoxList("Tags", (List<SelectListItem>)ViewBag.Tags)
    <input type="submit" value="submit" />
}

```

با توجه به اینکه کدهای Razor قرار گرفته در پوشه خاص app_code در ریشه سایت، به صورت خودکار در حین اجرای برنامه کامپایل می‌شوند، متد Helpers.CheckBoxList در تمام Viewهای برنامه در دسترس خواهد بود. در این متد، یک نام و لیستی از SelectListItemها دریافت می‌گردد. سپس به صورت خودکار یک CheckBoxList را تولید خواهد کرد. برای دریافت مقادیر ارسالی آن به سرور هم باید مطابق متد GetTags تعریف شده در کنترلر Home عمل کرد. در اینجا Value عناصر انتخابی به صورت آرایه‌ای از رشته‌ها در دسترس خواهد بود.

روشی جامع‌تر

در آدرس زیر می‌توانید یک HtmlHelper بسیار جامع را جهت تولید CheckBoxList در ASP.NET MVC بیابید. در همان صفحه روش استفاده از آن، به همراه چندین مثال ارائه شده است:

<https://github.com/devnoob/MVC3-Html.CheckBoxList-custom-extension>

نظرات خوانندگان

نویسنده: Mohsen Shams
تاریخ: ۱۳۹۱/۰۲/۱۲ ۰۷:۵۸:۱۴

با سلام خدمت آقای نصیری عزیز
روز معلم و استاد را به شما تبریک می گویم
شما حق بزرگی بر گردن ما دارید
با توجه به اینکه بارها و بارها در این بلاگ اشاره کرده اید که شما را استاد نخوانیم، لذا بنده به ناچار از این لغب استفاده نمی کنم
اما لازم به ذکر است، شما به گردن تمام بازدید کنندگان حق بزرگی دارید
زیرا تمام بازدید کنندگان بلاگ شما، به نحوی از این مطالب استفاده میکنند، یکی در پروژه هایش، یکی در کتابش، یکی در تحقیق و
غیره
پس تبریک این روز به شما حداقل کاری است که می توانیم انجام دهیم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۱۲ ۰۸:۴۴:۱۸

سلام، ممنون. شما لطف دارید.
سلامت باشید و موفق

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۲/۱۲ ۱۷:۰۱:۴۹

من تعریف نمیکنم چون میترسم خوستون نیاد.
فقط همینقدر که امام علی فرمودند : "هر کسی کلمه ای را به من بیاموزد مرا بنده ی خویش کرده است."
ولی من تا الان اینجا اندازه 30 تا کتاب مفید، مطلب یاد گرفتم.
روزت مبارک مهندس

نویسنده: سعید شیرزادیان
تاریخ: ۱۳۹۱/۰۲/۱۲ ۲۱:۰۴:۲۴

استاد نصیری عزیز خسته نباشید
زحمات بسیاری در رابطه با آموزش کشیدید متشکر. امکانش هست که مثالی از نحوه فراخوانی اطلاعات از دیتابیس بوسیله
Entity Framework در mvc نیز بیاورید
متشکرم

نویسنده: reza
تاریخ: ۱۳۹۱/۰۲/۱۳ ۲۲:۴۶:۴۸

این دوست عزیز راست می گه بعضی ها نون شبی که می برن خونه از برکت مطالب شماست واقعا مطالبتون عالی و کاربردی
است

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۴/۱۲ ۱۶:۴۳

با سلام و خسته نباشید و تشکر از وبلاگ واقعا عالی و پر محتواتون
یه سوال از خدمتتون داشتم
من به این روشی که گفتین در پروژه ام از CheckBoxList استفاده کردم

به این صورت که من از Membership خود دات نت برای مدیریت کاربران و نقش‌های کاربری استفاده کردم. نقش‌ها را از دیتابیس میخونم و در CheckBoxList نشون میدم. برای ایجاد یک User ممکنه چند نقش انتخاب بشه و اینو داخل پایگاه داده ثبت میکنم. حالا سوال من اینجاست که وقتی میخام موقع ویرایش یک User نقش‌های اونو از پایگاه داده بخونم چطور این نقش‌ها رو به CheckBoxList بایند کنم؟

چون من View مربوط به تابع Edit دارم که از نوع Strongly Typed هستش و نمیتونم این کارو انجام بدم. لطفا منو راهنمایی کنین

نویسنده: وحید نصیری
تاریخ: ۱۷:۱۱۳۹۱/۰۴/۱۲

از توضیحات مطلب فوق ایده بگیرید. مثلا:

```
@if (item.Selected) { <text>checked="checked"</text> }
```

این روشی هست برای انتخاب یک گزینه بر اساس شرط. اگر انتخاب شده بود (یا هر شرط دیگری)، چک مارک آن قرار داده می‌شود.

نویسنده: محسن
تاریخ: ۱۱:۲۷۱۳۹۱/۰۴/۱۳

با سلام مجدد. با تشکر از جواب شما. من مشکلمو توی تابع Edit() حل کردم. به این صورت که ابتدا همه نقش‌ها رو از پایگاه داده میخونم و توی یه `List<SelectListItem>` نگهداری میکنم. و برای هر کاربر هم نقش هاشو میخونم. سپس مقایسه میکنم که هر نقش که کاربر داره رو توی اون `List<SelectListItem>` خاصیت Selected رو true میکنم. این روش جواب داد اما ببخشید متوجه نشدم روشی که شما می‌گین چطوری هستش. فکر میکنم روش شما ساده‌تر باشه. کدهای من اینه:

```
//this method is in "UserController" class that select all available roles form database
[NonAction]
public List<SelectListItem> GetRoleList()
{
    var usrMgmt = new UserManagement();
    var RoleList = new List<SelectListItem>();

    foreach (KeyValuePair<string, string> pair in usrMgmt.GetAllRoles())
    {
        RoleList.Add(new SelectListItem { Text = pair.Value, Value = pair.Key, Selected = false
    });
    }

    return RoleList;
}

//-----
//this method is in "UserController" class that use for editing a user
[HttpGet]
public ActionResult Edit(string Username)
{
    var roles = GetRoleList();

    UserManagement usrMgmt = new UserManagement();
    var query = usrMgmt.Select(Username);

    foreach (string role in query.Roles)
    {
        int index = roles.FindIndex(x=>x.Value == role);
        roles[index].Selected = true;
    }
    ViewBag.Roles = roles;

    return View(query);
}

//-----
//this method is in "UserManagement" class in model layer, return a
//dictionary<string,string> that first string is english role name and second one is persian role
//name
```

```
public Dictionary<string,string> GetAllRoles()
{
    Dictionary<string, string> roles = new Dictionary<string, string>();
    var db = new SamaEntities();
    string query = "";
    string[] roleNames = Roles.GetAllRoles();

    foreach (string role in roleNames)
    {
        query = db.aspnet_Roles.Single(x => x.RoleName == role).Description;
        roles.Add(role, query);
    }
    return roles;
}
```

نویسنده: وحید نصیری
تاریخ: ۱۱:۵۵ ۱۳۹۱/۰۴/۱۳

- روش شما هم خوبه. یعنی نهایتا همان SelectListItem بحث شده در مطلب فوق رو ایجاد و استفاده کردید. مطلبی که عنوان کردم در مورد قسمت اصلی پیاده سازی checkbox list بود.

- ضمن اینکه در کدهای شما به نظر می‌رسه از الگوی واحد کار استفاده نشده. شما در هر متدی یک new دارید. یعنی یک اتصال به بانک اطلاعاتی. این مورد در برنامه‌های وب می‌تونه سرعت کار و بار بانک اطلاعاتی رو (بسته به تعداد کاربر) به شدت بالا ببره.

کار الگوی واحد کار، مدیریت یک Context به ازای تمام متدهای درگیر است. یک کانکشن و یک تراکنش به ازای n متد و n کلاسی که برای انجام یک کار مورد نیاز هستند.

در این مورد قبلا مفصل [توضیح دادم](#).

نویسنده: محسن
تاریخ: ۱۲:۱۴ ۱۳۹۱/۰۴/۱۴

سلام آقای نصیری ممنون از مطلب خوبتون. ما تازه کاریم. حالا حالاها مونده تا تجربیات شما رو بدست بیاریم. ممنون که اشکالمو بهم گفتین

نویسنده: علی
تاریخ: ۱۳:۵۸ ۱۳۹۲/۰۸/۰۲

من یک گرید دارم که به جدول یک بایند هست. یک ستون در این گرید شامل یک دکمه هست که یک پنجره درون اون باز میشه که اون پنجره شامل یک گرید هست که اطلاعات جدول ب درون اون هست یک ستون از این گرید 2 چک باکس هست.. یک دکمه هم جهت ارسال اطلاعات انتخابی به کنترلر خود هم در این پنجره داریم ولی با توجه به اینکه من مقداری ورودی خود را آرایه ای مثل مثال فوق result[] string گرفتم و اسم چک باکس من هم result بوده ولی پس از زدن دکمه مقدار نال ارسال میگردد به کنترلر از فروم‌های مشابه قبلا استفاده کردم و همچنین از مثال‌های تلریکدر این مورد واسه گرید استفاده کردم ولی به این شکل نیستند (گرید < پنجره < گرید 2) نتونستم مقادیر انتخابی رو واسه گریدم ارسال کنم...

نویسنده: وحید نصیری
تاریخ: ۱۴:۰۶ ۱۳۹۲/۰۸/۰۲

- [کد کاملت باید باشه](#) تا بتونم دیباگش کنم. این توضیحات رو جایی ارسال کنید، تصور نکنید که یک نفر مثلا یک ساعتی سعی در بازسازی مثال شما خواهد کرد. بعد سعی می‌کند تا خطای شما را به نحوی باز تولید کند و بعد هم مثلا یک مقاله جهت توضیح جزئیات آن در نصف روز منتشر می‌کند. این امر هیچ وقت رخ نخواهد داد. برای پاسخ دادن به یک سؤال، سرعت بازسازی آن مهم است.

- اگر نال دریافت می‌کنی ممکن است اطلاعات شما در یک فرم صحیح که به کنترلر اشاره می‌کند قرار ندارد یا حتی ممکن است نام‌های دیگری در سمت کاربر رندر شده باشند. هر دو مورد را با استفاده از نکات مطلب « [نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی بر jQuery](#) » می‌توانید بررسی کنید. به این ترتیب مشخص میشه اصلا چه اطلاعاتی در برگه شبکه

این افزونه زمان ارسال به سرور لاگ شده. این را بررسی کنید و سپس با ساختار داده کنترلر خودتون مقایسه‌اش کنید. به صورت خلاصه، برگه network افزونه فایرباگ را در حین ارسال اطلاعات به سرور بررسی کنید. ساختار اطلاعات آن چیست؟ آیا تطابقی با پارامترهای کنترلر شما دارد یا خیر؟ مثلاً من در مثال فوق نوشتم result. این مورد به این معنا نیست که خروجی نهایی HTML کار شما هم حتماً result نام دارد (خصوصاً اگر خودکار تولید می‌شود). بسته به نام واقعی عناصر انتخابی، نام این پارامتر متغیر خواهد بود. اما اصول نهایی یکی است.

- ضمناً روش دوم یافتن نام واقعی پارامترهای ارسالی، استفاده از روش قدیمی [Request.Form](#) در یک اکشن متد هست. این خاصیت دریافتی را دیباگ کنید تا به نام‌های واقعی برسید.