

اجرای این نوع صفحات کار سختی نیست؛ با کمی جستجو در اینترنت مثلا در [اینجا](#) میتوانید چیزهای خوبی پیدا کنید. اما متأسفانه اکثر مثال‌ها چیزی شبیه قرار دادن پارشال "ورود اعضا" در کنار پارشال "ثبت نام" هستند. حتما متوجه شده‌اید که معمولا این دو صفحه پس از PostBack به صفحه‌ای جدید هدایت میشوند و یا در بهترین حالت به کمک Ajax، پس از انجام عملیات، پیامی به کاربر نمایش میدهیم.

در این مقاله سعی شده روشی برای ایجاد چند فرم در یک View توضیح داده شود با این شرط که:

اولا : از Ajax یا هلپر ایجکسی استفاده نکنیم.

ثانیا : پس از post-back، عملیات Redirect را انجام ندهیم و صفحه جاری را حفظ کنیم؛ چه قرار باشد همه چیز درست انجام شده باشد و چه مشکلی پیش آمده باشد و پیام خطایی در کنار فیلدها نمایش داده شود.

در این روش به این نکته توجه شده که هر مدل پس از Post-back حفظ شود و مستقل از دیگری رفتار کند. مثلا اگر یکی از فرم‌ها ناقص پر شد و دکمه‌ی ارسال آن فشرده شد، پس از Post-back، فقط و فقط اجزای همین فرم Validate شود و فرم دوم بدون تغییر باقی بماند.

ویوی زیر را در نظر بگیرید. در layout، دو پارشال، به کمک اکشن‌متد فراخوانی شده‌اند:

ViewModel1های مرتبط با این دو بخش به شکل زیر هستند :

ContactVM .cs

```
public class ContactVM
{
    [Display(Name = "نام")]
    [Required(ErrorMessage = "{0} را وارد کنید")]
    public string Name { get; set; }

    [EmailAddress(ErrorMessage = "آدرس ایمیل صحیح نیست")]
    [DataType(DataType.EmailAddress)]
    [Display(Name = "آدرس ایمیل")]
    [Required(ErrorMessage = "{0} را وارد کنید")]
    public string EmailAddress { get; set; }

    [Display(Name = "متن پیام")]
    [Required(ErrorMessage = "حرفی برای گفتن ندارید؟")]
    public string Description { get; set; }

    [Required(ErrorMessage = "{0} را وارد کنید")]
    [Display(Name = "حاصل جمع")]
    public string Captcha { get; set; }
}
```

SubscriberVM .cs

```
public class SubscriberVM
{
    /*[RegularExpression("^([a-zA-Z0-9_\\.-]+)@([a-zA-Z0-9-]+\\.)+[a-zA-Z]{2,6}$", ErrorMessage =
    "آدرس ایمیل صحیح نیست")]*/
    [EmailAddress(ErrorMessage = "آدرس ایمیل صحیح نیست")] /*.Net4.5*/
    [Display(Name = "ایمیل")]
    [Required(ErrorMessage = "{0} را وارد کنید")]
    public string Email { get; set; }

    [Display(Name = "وضعیت")]
    public bool IsActive { get; set; }
}
```

در Layout، دو اکشن متد صدا زده شده‌اند که وظیفه ارسال ویوهای هر کدام به Layout را به عهده دارند :

```
<div class="row footerclass">
    <div class="col-md--6">
        @Html.Action("Subscribers", "Home")
    </div>
    <div class="col-md-6">
        @Html.Action("Contact", "Home")
    </div>
</div>
```

اکشن متدهای این دو پارشال به شکل زیر هستند :

```
public ActionResult Contact()
{
    return PartialView("_Contact", model);
}

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]

public ActionResult Contact(ContactVM model)
{
    if (ModelState.IsValid)
    {
        //Do Something
    }
    return PartialView("_Contact", model);
}

public ActionResult Subscribers()
{
}
```

```

        return PartialView("_Subscribers");
    }

    [HttpPost]
    [AllowAnonymous]
    [ValidateAntiForgeryToken]
    public ActionResult Subscribers(SubscriberVM model)
    {
        if (ModelState.IsValid)
        {
            //Do Something
        }
        return PartialView("_Subscribers", model);
    }

```

و اما ویوهایی که قرار است نمایش داده شوند:

_Contact.Cshtml

```

@model IrsaShop.Models.ViewModel.ContactVM

<span></span><span>ما تماس با</span>
<hr />

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    @Html.ValidationSummary(true)
    <div>
        @Html.TextBoxFor(m => m.Name, new { @class = "form-control", @id = "name", @name = "name",
placeholder = "نام" })
        @Html.ValidationMessageFor(m => m.Name)
    </div>
    <div>
        @Html.TextBoxFor(m => m.EmailAddress, new { @class = "form-control", @id = "email", @name =
"email", placeholder = "ایمیل", @style = "direction: ltr" })
        @Html.ValidationMessageFor(m => m.EmailAddress)
    </div>
    <div>
        @Html.TextAreaFor(model => model.Description, new { @class = "form-control", @id = "message",
@name = "message", placeholder = "پیام", @style = "max-width: 100%;height: 90px;" })
    </div>
    <div>
        <input type="button" value="" id="refresh" />
        
    </div>
    <div>
        @Html.TextBoxFor(model => model.Captcha, new { @class = "form-control", placeholder = "حاصل
جمع؟" })
        @Html.ValidationMessageFor(model => model.Captcha)
    </div>
    <div>
        <input type="submit" value="ارسال" name="submitValue" />
    </div>
}

```

Subscriber.Cshtml

```

@model IrsaShop.Models.SubscriberVM

<span></span><span>خبرنامه</span>
<hr />

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    @Html.ValidationSummary(true)
    <div>
        <div>
            @Html.TextBoxFor(m => m.Email, new { @class = "form-control right-buffer top-buffer pull-

```

```
right", @id = "email", @name = "email", placeholder = "ایمیل", @style = "direction: ltr; width: 50%",
@required = "required" })
    @*
        <button type="submit" name="submitValue">ثبت ایمیل</button>*@
        <input type="submit" value="ثبت ایمیل" name="submitValue" />
    </div>

</div>
@Html.ValidationMessageFor(m => m.Email, "", new { @class = "right-buffer pull-right" })
}
```

نکته اول : هیچ نوع ورودی برای `Html.BeginForm` در نظر گرفته نشده است. اگر اکشن متدی را برای صدا زدن در این بخش در نظر بگیرید، هنگام Postback به مشکل برخورد خواهید کرد؛ چون آدرس آن اکشن متد به شکل صریح در آدرس مرورگر فراخوانی میشود و پارشال ما پس از Post-back به تنهایی و بدون Layout نمایش داده خواهد شد. اسم بردن از اکشن متد وقتی کارساز است که آن اکشن متد قرار باشد یک Redirect انجام دهد ولی هدف ما این است که صفحه را از دست ندهیم و پیام‌های خطای ModelState را در همان صفحه قبل و پس از Post-back ببینیم و همچنین پس از انجام عملیات (مثلا ارسال پیام) همین صفحه نمایش داده شود.

نکته دوم : نکته اول یک مشکل دارد! اگر به شکل صریح اکشن متد مربوط به Post-back مشخص نشود، بطور اتوماتیک تمامی اکشن متدهایی که ویژگی [HttpPost] دارند اجرا خواهند شد. این یعنی هر دو اکشن متد Contact و Subscriber اجرا می‌شوند و بنابر آنچه در اکشن متدها نوشته‌ایم هر دو ModelState بررسی می‌شود که این هدف ما نیست. مثلا فرم سمت چپ را تکمیل کرده ایم و دکمه "ثبت ایمیل" را فشار داده‌ایم و صفحه Postback می‌شود و با اینکه ایمیل در بانک ثبت شده اما فرم سمت راستی با خطا ظاهر میشود که چرا فیلدها خالی هستند؟!

برای حل این مشکل کافیت نام مربوط به دکمه‌ها را به شکل یک ورودی [برای اکشن متدها بفرستیم](#) و بر اساس وضعیت آن تنها state مدل مورد نظر خودمان را بررسی کنیم. پس اصلاح زیر را برای اکشن متدهای دارای ویژگی [HttpPost] انجام میدهیم.

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]

public ActionResult Contact(ContactVM model, , string submitValue)
{
    if (submitValue == "ارسال")
    {
        if (ModelState.IsValid)
        {
            //Do Something
        }
    }
    else
    {
        ModelState.Clear();
    }
    return PartialView("_Subscribers", model);
}

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public ActionResult Subscribers(SubscriberVM model, string submitValue)
{
    if (submitValue == "ثبت ایمیل")
    {
        if (ModelState.IsValid)
        {
            //Do Something
        }
    }
    else
    {
        ModelState.Clear();
    }
    return PartialView("_Subscribers");
}
```

نکته سوم : در این روش سعی کنید از ViewModel استفاده کنید و سعی کنید ویو مدل‌ها پراپرتی‌های با نام یکسان نداشته باشند. مثلا پراپرتی Email در ویو مدل‌ها نام‌های متفاوتی داشته باشند (مثل ContactMail , Email , EmailAddress و ...). با اینکار در زمان Postback احتمال اینکه فیلدهای مشترک اتوماتیک پر شده به ما نمایش داده باشند صفر خواهد شد.

نکته چهارم : حواستان باشد پس از انجام عملیات مرتبط با هر فرم در اکشن متد مربوط به آن (مثلا ارسال ایمیل، ثبت در بانک یا ... در صورتی که عملیات با موفقیت انجام شد حتما ModelState را clear کنید. با اینکار پس از Post-back فیلدهای پارشال‌ها خالی میشوند.

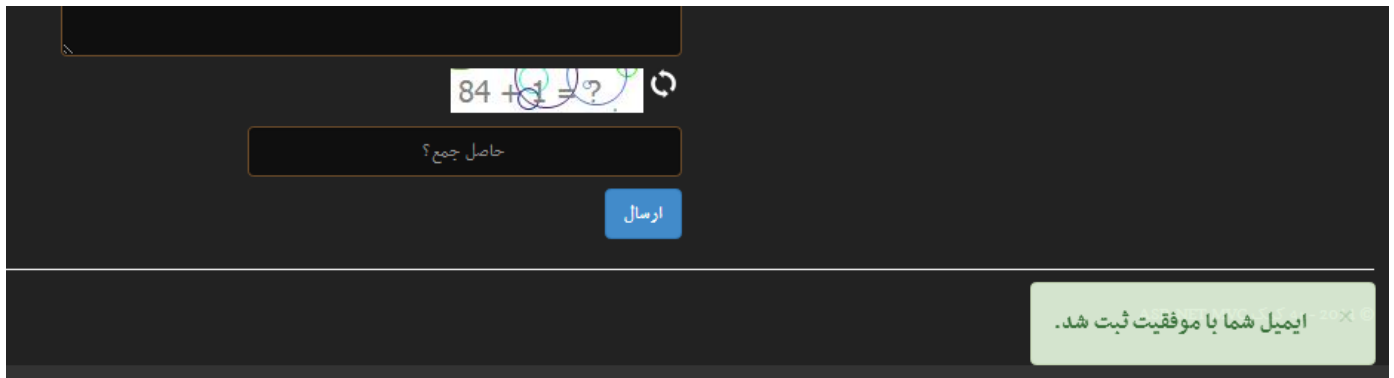
نکته پنجم : میتوانید به سادگی مدیریت خطا را به کمک جی کوئری انجام دهید؛ مثلا فرض کنید میخواهیم اگر ایمیل کاربر برای دریافت خبرنامه با موفقیت ثبت شد، پیامی مبنی بر موفقیت برای وی بفرستیم؛ اکشن متد HttpPost مربوط به Subscriber را به شکل زیر تکمیل میکنیم :

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public ActionResult Subscribers(SubscriberVM model, string submitValue)
{
    if (submitValue == "ثبت ایمیل")
    {
        if (ModelState.IsValid)
        {
            Subscriber mail = new Subscriber() { Email = model.EmailSubscriber, IsActive = true };
            context.Subscribers.Add(mail);
            context.SaveChanges();
            ViewBag.info = "ایمیل شما با موفقیت ثبت شد.";
            ViewBag.color = "alert-success";
            ModelState.Clear();
        }
        else
        {
            ModelState.Clear();
        }
        return PartialView("_Subscribers ");
    }
}
```

در انتهای پارشال Subscriber_ هم چند خط کد زیر را مینویسیم :

```
@if (!String.IsNullOrEmpty(ViewBag.info))
{
    <div id="info" style="position: fixed; bottom: 0; right: 0; margin-right: 1%;">
        <div class="alert @ViewBag.color alert-dismissible">
            <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button>
            <strong> @ViewBag.info</strong>
        </div>
    </div>
    <script>
        $(function () {
            $("#info").fadeOut(15000);
        });
    </script>
}
```

نتیجه این خواهد بود که پس از PostBack در صورت موفقیت تصویر زیر را خواهیم دید و 15 ثانیه المان سبزرنگ بوت استرپ زیر نمایش داده خواهد شد.



The screenshot shows a web form on a dark background. At the top, there is a text input field. Below it is a captcha image displaying the equation $84 + 1 = ?$ with a refresh icon to its right. Under the captcha is a text input field containing the Persian text "حاصل جمع؟" (Sum result?). Below this is a blue button labeled "ارسال" (Send). At the bottom right, there is a green notification box with a close icon and the Persian text "ایمیل شما با موفقیت ثبت شد." (Your email has been successfully registered).

این روش نوعی مدیریت میان اکشن متدهای دارای ویژگی `HttpPost` است و همانطور که گفتیم به علت اینکه پس از `Post-Back` نیاز به ساختار به هم نخورده‌ی صفحه‌ی قبلی داریم، نمیتوانستیم به شکل صریح، اکشن متد برای `Html.BeginForm` تعریف کنیم تا این دروسرها را نداشته باشیم.

حین نوشتن این مقاله به علت وجود `if` های تو در تو، امیدوار بودم که روش‌های بهتری برای اینکار موجود باشند و هنوز هم امیدوارم نظرات شما چنین چیزی را نشان دهد.

نظرات خوانندگان

نویسنده: میثم ثوامری
تاریخ: ۲۳:۵۱ ۱۳۹۳/۰۳/۲۸

حین نوشتن این مقاله به علت وجود `if` های تو در تو، امیدوار بودم که روش‌های بهتری برای اینکار موجود باشند و هنوز هم امیدوارم نظرات شما چنین چیزی را نشان دهد.

[Multiple submit buttons with ASP.NET MVC](#)

نویسنده: مرتضی دلیل
تاریخ: ۰:۳۳ ۱۳۹۳/۰۳/۲۹

مرسی از لینک، اینو دیده بودم بچه‌های اینجا هم [مقاله اش](#) رو نوشته بودن، تو اون مقاله یک فرم میسازیم با چند کارکرد ولی داستان اینجا کمی متفاوت. به نظر من به جاهایی ساختن پارشال یا اکشن‌هایی که پارشال صدا میزنن خیلی ارزشمند هست و به ما برای رسیدن به یک فریم ورک شخصی کمک میکنه. بیشتر هدف من از این مقاله ساختار دادن به برنامه به کمک پارشال‌ها بود. یک ویوی تصور کنید که دارای پنج پارشال با پنج فرم مختلف باشه، ممکنه الان تصورش مسخره باشه ولی شاید به روزی به کار اومد. حداقلش به بازی خوب هست برای فهمیدن ظرفیت‌های MVC

نویسنده: رمزینه
تاریخ: ۱۱:۱۳ ۱۳۹۳/۰۳/۲۹

مرسی
من دقیقاً الان دنبال همین موضوع بودم و به ویو دارم با 9 پارشال ویو که از همین روش باید استفاده کنن.

نویسنده: علیرضا همتی
تاریخ: ۱۲:۱۱ ۱۳۹۳/۰۳/۳۰

سلام.
مرسی از مقاله خوبتون.
من به مشکلی دارم شاید اگه ممکنه منو هم راهنمایی کنین.
فرض کنیم به صفحه داریم که توی اون صفحه چند تا فرم وجود داره و تعداد این فرم‌ها هم نا مشخص هستش و این فرم‌ها به صورت پارشل ایجاد میشن. و همچنین قصد بر این است که همه این فرم‌ها توسط یک دکمه و به ترتیب سابمیت بشن. من خودم این کار رو به کمک جاوا اسکریپت انجام دادم. حالا می‌خواستم ببینم پیشنهاد شما چیه. ممنون از همتون.

نویسنده: محسن خان
تاریخ: ۱۲:۱۸ ۱۳۹۳/۰۳/۳۰

در حاشیه! تلفظش پارشال یا پارشل هست: <https://www.google.com/search?q=partial>

نویسنده: رشوند
تاریخ: ۲۰:۵۱ ۱۳۹۳/۰۵/۳۰

با سلام سپاس؛

- 1- اگر لطف کنید کمی در باره `Html.Action` توضیح بدید ممنون میشم.
- 2- دو تا `partial` را با استفاده از `Html.Action` و `return` اکشن مربوط، به صفحه اصلی اضافه کردید.

```
public ActionResult Contact()
{
    return PartialView("_Contact", model);
}
```

model هر کدام از partialها رو مشخص کردید

```
@model IrsaShop.Models.ViewModel.ContactVM
```

```
@model IrsaShop.Models.SubscriberVM
```

سوال: در صفحه اصلی مدل ما به چه صورتی خواهد بود؟

```
@model IrsaShop.Models.ViewModel.?
```

آیا باید از viewmodel استفاده کنیم که هر دو (SubscriberVM , ContactVM) و موارد مورد نیاز دیگر صفحه را در بر بگیرد؟
با فرض اینکه مثلا خود صفحه هم از مدل دیگری برای استفاده از helper استفاده می‌کند،
آیا مدل برنامه تاثیری بر partialها دارد؟ (در این روش)
@Html.Action@ مدلی را به partial ارسال نمی‌کند برخلافه @Html.Partial که model مورد نیاز را ارسال می‌کند.

(لطفا مودویو مورد نیاز رو شرح بدید)

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۵/۳۰ ۲۱:۱۵

در مورد @Html.Action : [نمایش اطلاعات از کنترلرهای مختلف در یک صفحه](#)

نویسنده: رشوند
تاریخ: ۱۳۹۳/۰۶/۰۱ ۲۲:۱۸

سپاس، در لینک گفته شده است:
حین فراخوانی متد @Html.Action، یک متد در یک کنترلر فراخوانی خواهد شد (که شامل ارائه درخواست و طی سیکل کامل پردازشی آن کنترلر نیز خواهد بود). سپس آن متد با بازگشت دادن یک **PartialView**، اطلاعات پردازش شده یک partial view را به فراخوان بازگشت می‌دهد
دقیقا همین مورد در این آموزش اجرا شده است

```
return PartialView("_Contact", model);
```

ولی اگر بخواهیم بعد انجام عملیات، اکشن دیگری اجرا بشه چه کار باید کرد؟
من از RedirectToAction استفاده کردم ولی متاسفانه error داد...

```
if (this.ModelState.IsValid)
{
    return RedirectToAction(actionName: "show", controllerName: "Posts", routeValues: new {
id });
}
```

پیغام:

Error executing child request for handler
."System.Web.Mvc.HttpHandlerUtil+ServerExecuteHttpHandlerAsyncWrapper

.Child actions are not allowed to perform redirect actions

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۰۲ ۰:۴۰

child action قراره فقط محتوای قسمت کوچکی از صفحه رو تامین کنه. در اینجا redirect کل صفحه معنی نداره چون هنوز کار رندر صفحه تموم نشده و وسط کار هست.

نویسنده: رشوند
تاریخ: ۱۳۹۳/۰۶/۰۲ ۷:۵۶

به نظرتون راه حل چیست؟

در مثال این آموزش، بعد از پایان یافتن عملیات بصورت موفق، فقط "پیام با موفقیت ثبت شد" نمایش داده می‌شه، ولی بنده می‌خواهم بعد از پایان صحیح عملیات اکشن دیگه ای فعال بشه، مثلاً بعد از ثبت نظر مربوط به یک خبر در سایت، خبر مورد نظر دوباره نمایش داده بشه تا نظر جدید لود بشه. ولی این اتفاق با خطای مذکور همراه، حتی به صورت دستی هم صفحه را reload می‌کنم عملیات قبلی دوباره انجام می‌گیره (پی‌اپی نظر قبلی ثبت می‌شود)

شما فرمودید هنوز وسط کاره، حق با شماست، ایا راه حلی هست؟ یا باید از این شیوه برای اینجور کارها استفاده نکنم؟ در صورت عدم استفاده از این روش پیشنهادتون چیست؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۰۲ ۱۰:۰۸

در مثال بالا نیازی به استفاده از `Html.Action` و `child action` نبوده. از `Html.Action` زمانی استفاده میشه که قراره چند `partial view` از پیش مقدار دهی شده در صفحه، از منابع داده مختلفی مقدار دهی بشن. مثلاً قراره در کنار صفحه، از یک منبع داده، آمار سایت رو نمایش بدید (اطلاعات از پیش مقدار دهی شده) و از یک منبع دیگه، لیست مطالب و این دو مورد هم قراره در `layout` سایت قرار بگیرن. اینجا شما ویرایش اطلاعات رو که ندارید (برای مقدار دهی عناصر فرم از قبل). فقط قراره از کاربر اطلاعات بگیرید. بنابراین استفاده از رندر کردن پارشال مثل قبل کار می‌کنه. مهم هم نیست که هر پارشالی مدل مختلفی داره. برای ارسال اطلاعات فرم به سرور، فقط نام کنترلرها مهم است و نه نوع مدل شما. پروتکل HTTP چیزی از نوع مدل نمی‌دونه. ASP.NET MVC هست که نام‌ها رو می‌گیره و متصل می‌کنه به خواص یک شیء.

در کل مثال بالا برای حالت قرار دادن این فرم‌ها در `layout` سایت برای تمام صفحات طراحی شده. اگر یک صفحه معمولی و `view` معمولی هست، در ASP.NET MVC مجاز هستید n تا فرم در صفحه قرار بدید. برای اینکار نیازی به `Html.Action` نیست؛ با رندر پارشال هم کار می‌کنه. برای تامین داده اون‌ها هم از `ViewModel` استفاده کنید. این `ViewModel` یک خاصیتش، کل اطلاعات فرم یک هست و خاصیت دومش کل اطلاعات فرم 2 و الی آخر.

نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۶/۰۲ ۱۶:۴۳

نتیجه استفاده از هلیپر پارشال یا اکشن شبیه به هم هست و استفاده از اکشن برای من یک عادت شده، چون اضافات کدی رو میشه از اون طریق به پارشال بازگشتی تزریق کرد