عنوان: **CSS پویا در ASP.NET MVC**
نویسنده: ناصر طاهری
تاریخ: ۱۳۹۲/۰۹/۱۲ ۲۲:۵
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net MVC, Dynamic Css

سناریو هایی وجود دارد که نیاز است مشتری ، خود شیوه نامه هایی (CSS) را برای قسمت‌های مختلف سایت انتخاب کند. برای مثال تنظیماتی را برای منوی سایت در نظر گرفته ایم که مشتری بتواند رنگ و قلم و ... را متناسب با سلیقه‌ی خود تغییر دهد و یا یک قسمت کلی برای اعمال شیوه نامه‌ها به سایت ایجاد کرده ایم که در همه‌ی قسمت‌های سایت اعمال شود. بدین شکل در صورتی که مشتری، اطلاعات اندکی هم در مورد CSS داشته باشد میتواند ظاهر سایت خود را به آسانی تغییر دهد و تا حدودی بار را از روی دوش پشتیبان سایت بر میدارد.
و برای همه‌ی این‌ها نیاز است تا فیلدی در دیتابیس برای ذخیره‌ی شیوه نامه‌های مشتری ایجاد شود و یا در یک فایل متنی ذخیره شود که بسته به سیاست برنامه نویس دارد.
در این مطلب تصمیم داریم این سناریو را به صورت ساده در یک پروژه ASP.NET MVC پیاده سازی کنیم.
ایتدا یک پروژه از نوع ASP.NET MVC 4 ایجاد میکنیم. سپس قطعه کد زیر را به فایل _Layout.cshtml موجود در مسیر Views/Shared به صورت زیر اضافه میکنیم :

```
<head>
    @*سایر شیوه نامه‌ها و اسکریپت ها*@
    @RenderSection("styles", required: false)
</head>
```

RenderSection این امکان را میدهد که ما بتوانیم شیوه نامه‌ی دیگری را در صفحات دیگر به MasterPage خود تزریق کنیم.
سپس کنترلری به نام Home ایجاد میکنیم :

```
namespace DynamicCssExample.Controllers
{
    public class HomeController : Controller
    {

        public ActionResult Index()
        {
            return View();
        }

        public string GetStyle()
        {
            Response.ContentType = "text/css";
            //در این قسمت میتوانیم به دیتابیس متصل شویم و شیوه نامه‌ی مورد نظر را واکشی کنیم و بازگشت
دهیم
            return "h2{color:yellow}";
        }

    }
}
```

در اینجا یک متد به نام GetStyle داریم که وظیفه‌ی بازگشت یک رشته را بر عهده دارد و نوع این مقدار بازگشتی از نوع text/css است و میتواند شیوه نامه هایی که در دیتابیس و یا یک فایل متنی ذخیره کرده ایم را واکشی و به استریم ارسال کند.
در انتها برای استفاده از این متد کافی است در View مربوطه بدین شکل عمل کنیم :

```
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

@section styles
{
    <link rel="stylesheet" href="@Url.Action("GetStyle", "Home")" type="text/css"/>
}

<h2>Index</h2>
```

در سکشن Style لینک شیوه نامه ای تعریف کرده ایم که ویژگی href آن به اکشن GetStyle در کنترلر Home اشاره میکند و این اکشن نیز محتوای شیوه نامه را برگشت میدهد.

مثال این مطلب : DynamicCssExample.zip به همراه بازسازی کامل پوشه packages بسته‌های NuGet به صورت خودکار

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۲ ۲۲:۵۴

این روش را می‌شود کمی بهبود داد؛ برای اینکه بشود داخل فایل CSS با کدهای Razor هم کار کرد (یعنی چیزی شبیه به LESS اما پیاده سازی شده با Razor و تفسیر شده توسط موتور آن؛ مانند یک View یا Partial View معمولی و کامل. حتی می‌شود داخل آن if و else یا حلقه نوشت):

```
public class CSS
    {
        public string Color { set; get; }
    }
```

```
public ActionResult GetDynamicStyle()
        {
            var color = "White";
            if (DateTime.Now.Hour > 18 || DateTime.Now.Hour < 8)
            {
                color = "Black";
            }

            this.Response.ContentType = "text/css";
            return PartialView(viewName: "~/Views/Home/_CSS.cshtml", model: new CSS { Color = color });
        }
```

با این محتوای Views/Home/_CSS.cshtml :

```
@model DynamicMvcCSS.Controllers.CSS

.foo {
    color: @Model.Color;
}
```

نویسنده: ناصر طاهری
تاریخ: ۱۳۹۲/۰۹/۱۲ ۲۳:۴۲

ممنون خیلی بهتر شد. برای موارد ثابت این روش خیلی شکیلتر و قانونمندتره مثل تنظیمات منوها و بعضی قسمت‌ها که فقط یک سری موارد خاص رو اجازه‌ی تغییر داره. اما اگر بخواهیم به مشتری اجازه بدهیم که هر قسمتی که دوست دارد را تغییر دهد، یعنی شیوه نامه ای که ایجاد میکنه بر کل سایت تاثیر بزاره ،دیگه فکر نکنم بتونیم اینطور قانونمند عمل کنیم درسته؟ راهی هم برای این مسئله وجود داره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۲ ۲۳:۴۶

- روش شما برای بازگشت یک رشته متغیر از پیش تعریف شده خوب است.
+ نیازی نیست کل CSS را در اختیار کاربر برای ویرایش قرار داد. قسمت‌هایی را که قرار است تغییر کنند به صورت فیلد دربیارید تا کاربر بتواند مقدار دهی کند. بعد با روش بالا (که model آن به صورت پویا قابل مقدار دهی است) می‌شود در فایل razor نهایی مثل یک view عناصر را مقدار دهی و استفاده کرد.
- ضمنا استفاده از Output cache هم توصیه می‌شود.