

در پروژه فروشگاهی تحت Asp.Net MVC 4 بهترین روش برای ایجاد یک سبد خرید استفاده از یک Api Controller می‌باشد.

من در پروژه‌ای که در MVC 3 داشتم این مورد را بدین شکل انجام داده بودم که با ایجاد یک کلاس و درج چند سطر کد در Global.asax این مورد حل میشد و در Api Controller ای که میخواستیم اطلاعات را در آن به کمک Session دریافت یا ویرایش کنیم، امکان دسترسی به Session را داشتم:

یک کلاس ایجاد کرده و کدهای زیر را در داخل آن درج میکنیم:

```
//using System.Web.Http.WebHost;
//using System.Web.Routing;
//using System.Web.SessionState;

public class MyHttpControllerHandler : HttpControllerHandler, IRequiresSessionState
{
    public MyHttpControllerHandler(RouteData routeData): base(routeData)
    {
    }
}

public class MyHttpControllerRouteHandler : HttpControllerRouteHandler
{
    protected override IHttpHandler GetHttpHandler(RequestContext requestContext)
    {
        return new MyHttpControllerHandler(requestContext.RouteData);
    }
}
```

و برای اینکه این مورد، در برنامه‌ای که ساختیم کار کند باید در Global.asax این کدها را درج کنیم (دوباره تاکید می‌کنم این نکته فقط در MVC 3 کار می‌کند)

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    var apiroute = routes.MapHttpRoute(
        name: "DefaultApi",
        routeTemplate: "api/{controller}/{id}",
        defaults: new { id = RouteParameter.Optional }
    );
    apiroute.RouteHandler = new MyHttpControllerRouteHandler();

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );
}
```

با تنظیمات فوق در برنامه‌های Asp.NET MVC 3 امکان استفاده از Session در API Controller ها در دسترس قرار می‌گیرد. اما این کار در ASP.NET MVC 4 قابل استفاده نیست و خیلی بهتر و راحتتر انجام می‌شود (میتوانید به WebApiConfig.cs در فولدر App_Start پروژه مراجعه نمایید)

حال چه تنظیماتی نیاز است تا دوباره این امکان برقرار شود؟

تنها کدهای ذیل را در Global.asax درج نمایید و امکان استفاده از Session را در API Controller های MVC4، ایجاد نمایید.

```
public class MvcApplication : System.Web.HttpApplication
{
    public override void Init()
    {
        this.PostAuthenticateRequest += MvcApplication_PostAuthenticateRequest;
```

```
        base.Init();
    }

    void MvcApplication_PostAuthenticateRequest(object sender, EventArgs e)
    {
        System.Web.HttpContext.Current.SetSessionStateBehavior(
            SessionStateBehavior.Required);
    }

    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();

        WebApiConfig.Register(GlobalConfiguration.Configuration);
        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
        RouteConfig.RegisterRoutes(RouteTable.Routes);
        BundleConfig.RegisterBundles(BundleTable.Bundles);
        AuthConfig.RegisterAuth();
    }
}
```

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۲۱:۳۷ ۱۳۹۲/۰۷/۱۷

لطفا این قسمت « بهترین روش برای ایجاد یک سبد خرید استفاده از یک Api Controller می باشد.» را بیشتر توضیح بدید. با تشکر.

نویسنده: بهمن خلفی
تاریخ: ۲۱:۴۶ ۱۳۹۲/۰۷/۱۷

در اکثر فروشگاههایی که با Asp.net MVC توسعه پیدا کردند اضافه کردن یک کالا به سبد خرید پروسه ای زمان بر بوده و کاربر پسند نیست در حالی که در فروشگاههای متن باز مشابه این عمل بصورت زیبا و کاملا پرسرعت انجام میشود . API یکی از مباحث خوبی است که در MVC براحتی قابل استفاده بوده و این قدرت را به برنامه نویسی میده که بتواند از مباحثی مانند Ajax یا JSON مثلا در سبد کاربر استفاده کند.
بهترین روش به نظر من است ! بدلیل اینکه بسیار راحت و امن است . راه اندازی آن در حد 10 الی 15 دقیقه بیشتر طول نمیکشد و شما میتوانید تمامی مباحثی مانند احراز هویت و ... را طبق اصول MVC روی همه Actionهای مورد نیاز اعمال کنید.
اگر اغراق شده فقط یک نظر شخصی است. (-)

نویسنده: محسن خان
تاریخ: ۲۱:۴۹ ۱۳۹۲/۰۷/۱۷

ممنون. JSON و Ajax با کنترلرهای معمولی MVC هم قابل دسترسی است. چه مزیت بیشتری رو با Web API به دست آوردید؟

نویسنده: بهمن خلفی
تاریخ: ۲۱:۵۷ ۱۳۹۲/۰۷/۱۷

بله این امکان وجود دارد اما شما این امکان را میتوانید در یک برنامه بصورت کلی استفاده نمائید یعنی در هر جای برنامه باشید میتوانید به این بخش دسترسی داشته باشید .
از طرفی شما با استفاده از API دیگر برخی از اطلاعات یا آیتمهای صفحه را بدلیل Post شدن صفحه (Dispose) از دست نخواهید داد.
استفاده از Web API ها یک عمل عمومی بحساب می آید اما در این مطلب مقصود دسترسی به Session در API Controller ها میباشد.

نویسنده: مهدیار
تاریخ: ۱۹:۲۲ ۱۳۹۲/۰۷/۱۹

استفاده از session ؟
به نظرم یک جدول موقت برای کارت می گزینید بهتر بود . این هم یک [مطلب](#) مفید برای استفاده نکردن از session.

نویسنده: محسن خان
تاریخ: ۲۱:۲۶ ۱۳۹۲/۰۷/۱۹

کوکی هم برای یک طراحی Http friendly خوب هست. خصوصا اینکه یک شخص آنچنان اقلام زیادی را که هربار خرید نمی کند. همچنین [HTML5 local storage](#) نیز مفید است.

نویسنده: بهمن خلفی
تاریخ: ۸:۵۷ ۱۳۹۲/۰۷/۲۰

بله دوست عزیز ، استفاده از Session مزایا و معایب خودش را به همراه دارد.

اما در یک سیستم فروشگاهی حذف و اضافه شدن کالا در سبد خرید بصورت مکرر انجام میشود و برای اینکه شما چه کاربران مهمان و چه کاربران عضو را مدیریت نموده و بتوانید ترافیک روی بانک اطلاعاتی خود را کاهش دهید میتوانید از Session استفاده نمایید . (بطور مثال اگر 1000 نفر در یک سایت فروشگاهی که حداقل 100 کالا ارائه شده است را به سبد خرید خود اضافه یا تعداد و ... آنها را ویرایش نمایند در بهترین حال 100000 درخواست به بانک اطلاعاتی ارسال و دریافت خواهد شد)

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۷/۲۰ ۸:۵۹

بله دوست عزیز ، این هم یک امکان بسیار خوبی است که در سمت کاربر میتوان ازش استفاده نمود.
در مطالب آینده سعی خواهد شد کاربرد این امکان در یک برنامه کاربردی ارائه شود. باتشکر

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۴/۰۸ ۲۰:۴۵

برای طراحی سبد کالا یا هر نوع آیتم مشابه آن بهتر هست که از روش هایی غیر از سشن بهره برد. چون سشن خودش سنگینی زیادی داره و با افزایش کاربران همزمان این افزایش مصرف حافظه بیشتر هم خواهد شد.

مشکلات دیگری هم که داره این هست که در سشن یک شی زمان دار هست و اگر کاربر 15مثلاً دقیقه کار نکند بعد از آن سبد خالی می شود که بهتر هست انتخابها همواره حفظ گردد . پس بهترین راهها استفاده از [کوکی \(+ + \)](#) است که در این مقاله نحوه [ذخیره هر نوع داده](#) (به همراه فشرده سازی) ذکر شده است و مورد دوم استفاده از [indexedDB و local storage](#) است که به زودی قسمت دوم هم منتشر می شود.

مورد بعدی اینکه در صورت به روز رسانی وب سایت به خصوص وب کانفیگ سشنها ریست خواهند شد و سبدی که کاربر پر کرده است در آن لحظه از بین خواهد رفت که با استفاده روش های کلاینتی چنین مشکلی ایجاد نمی شود.

مورد بعدی اینکه با استفاده از جاوااسکریپت و توابع تحت کلاینت سرعت بهتری در اجرا و رسم سبد کالا می توانید داشته باشید بدون اینکه بار زیادی را به سرور تحمیل کنید.

نکته تکمیلی اینکه فقط باید در رسم مجدد سبد کالا تحت کلاینت باید موجودی ها و قیمت ها و تخفیف ها و ... هم مجددا بررسی شوند که اطلاعات قبلی نمایش داده نشود که البته این بر میگرده به طراحی ساختاری که ترتیب داده اید.
استفاده از جدول جداگانه برای نگهداری اطلاعات هم روش مناسبی حساب می شود که انتخاب بین اینها به نظر برنامه نویس مربوط می شود.

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۵/۰۳ ۱:۱۳

یک نکته دیگر که در مورد ذخیره آیتمها مثل یک سبد کالا به ذهنم خورد این هست که روش ذخیره در یک جدول در سمت سرور در بعضی موارد کمک حال بسیار خوبی خواهد بود. در این حالت از مزایای زیر بهره مند خواهیم شد:
اینکه بفهمیم در حال حاضر چه محصولی بیشتر در سبد کالای کاربران قرار گرفته است که هنوز در خرید آن مردد هستند؟ آیا با یک تخفیف مناسب به آن محصول می توان آن ها را به سمت تایید سبد خرید تشویق کرد؟
و دیگر گزارشات مشابه