

اگر بخواهید در کد وب پارت خود از ویژگی Session استفاده کنید ، ممکن است SharePoint این اجازه را به شما ندهد... راه حل چیست؟

اگر کد خود را بعد از پیاده سازی Session اجرا کنید چنین پیغامی را مشاهده خواهید کرد:

Server Error in '/' Application

Session state can only be used when enableSessionState is set to true, either in a configuration file or in the Page directive. Please also make sure that System.Web.SessionStateModule or a custom session state module is included in the <configuration>\<system.web>\<httpModules> section in the application configuration. این پیغام نشان می دهد که تنظیمات Session در فایل web.config فعال نیست. یکی از راه حل های این مشکل چک کردن ویژگی `enableSessionState = "true"` در هدر صفحه و در تگ Page است

اگر این ویژگی True باشد و همچنان همان پیغام فوق را مشاهده کردید به فایل web.config وبسایت مورد نظر مراجعه کرده و به روش زیر عمل کنید :

```
// 1 - Find This Node
<modules runAllManagedModulesForAllRequests="true">
// 2 - Add These Nodes
<remove name="Session" />
<add name="Session" type="System.Web.SessionState.SessionStateModule" preCondition="" />
```

اکنون امکان استفاده از Session برای شما فراهم شده است .

در غیر این صورت باید تغییراتی در Permission Policy های شیرپوینت که در فایل های wss_minimaltrust.config و wss_mediumtrust.config ... انجام دهید و یک **Permission Set** به آن اضافه کنید

این فایل ها در زیر مجموعه 14 و در پوشه **CONFIG** قرار دارند .

موفق باشید

نظرات خوانندگان

نویسنده: هیمن روحانی
تاریخ: ۱۳۹۲/۱۱/۰۱ ۹:۳۱

سلام . تمام بخش‌های اصلی شیرپوینت بدون استفاده از session طراحی شدن. آیا بعد از فعال کردن session مشکلی برای بقیه وب پارتها پیش نمیاد؟
ممکنه با فعال کردن session انجام بعضی کارها برای توسعه دهنده ساده بشه ولی باید در نظر داشته باشید که session سر بار زیادی برای سرور داره و مایکروسافت بر اساس یه حساب کتابی غیرفعالش کرده.

نویسنده: محمد باقر سیف الهی
تاریخ: ۱۳۹۲/۱۱/۰۱ ۱۴:۳۵

سلام . [لطفا به اینجا رجوع کنید](#)

همانطور که بیان شد، در صورت نیاز به استفاده از Session و مشاهده خطا، یکی از راهکار ها، این مورد می‌باشد و هیچ اصراری به استفاده از اون نیست !
همچنین در این پست فقط دلیل این خطا و راه حل اون بیان شد و کسی کاری به مایکروسافت نداره !

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