

# Providing session state in ASP.NET WebAPI

1. By default, Session state is **disabled** in WEB API. So we need to configure the api project to Enable it.

**Add two class to enable Session state**

```
public class SessionStateRouteHandler : IRouteHandler
{
    IHttpHandler IRouteHandler.GetHttpHandler(RequestContext requestContext)
    {
        return new SessionableControllerHandler(requestContext.RouteData);
    }
}
```

**Remove the auto enabled code for routing in WebApiConfig**

```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        EnableCrossSiteRequests(config);
        config.MapHttpAttributeRoutes();

        //config.Routes.MapHttpRoute(
        //    name: "DefaultApi",
        //    routeTemplate: "api/{controller}/{action}/{id}",
        //    defaults: new { id = RouteParameter.Optional }
        // );
    }

    private static void EnableCrossSiteRequests(HttpConfiguration config)
    {
        var cors = new EnableCorsAttribute(
            origins: "*",
            headers: "*",
            methods: "*");
        config.EnableCors(cors);
    }
}
```

**Add a RouteConfig class to enable Custom Routing**

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

        routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Home", action = "Index", id = 1 }
        );
    }
}
```

```
using System.Web.Routing;
using System.Web.Http;

public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        RouteTable.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        ).RouteHandler = new SessionStateRouteHandler();
    }
}

protected void Application_Start()
{
    HttpContext.Current.SetSessionStateBehavior(SessionStateBehavior.Required);
    GlobalConfiguration.Configure(WebApiConfig.Register);
    RouteConfig.RegisterRoutes(RouteTable.Routes);
}
```

Add `RouteHandler`  
with session  
enabled class

Call the  
Methode from  
glabal.ascx