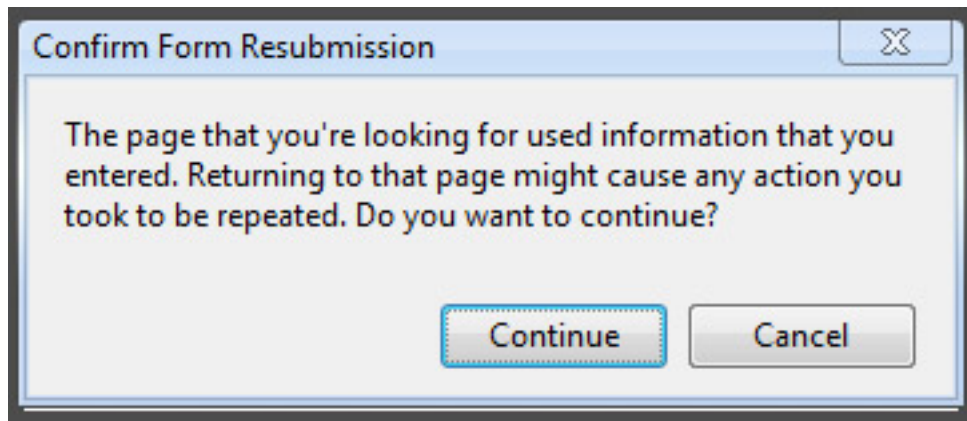


تا حالا با این پنجره حتما مواجه شدید:



دارید اطلاعات یک فرم داخل صفحه رو به سمت سرور میفرستید و پس از اتمام عملیات، صفحه دوباره نمایش داده میشه. در این حالت اگه دکمه F5 یا دکمه Refresh مرور گر رو بزنید، با این پنجره مواجه میشید که میگه دارید اطلاعات قبلی رو دوباره به سمت سرور میفرستید. بعضی وقت‌ها کاربران به هر دلیل دوباره صفحه رو Refresh میکنند و با این پنجره روبرو میشن بدون اینکه بدونن جریان از چه قراره، دوباره اطلاعات رو به سمت سرور میفرستن و این کار باعث ثبت اطلاعات تکراری میشه. برای جلوگیری از این کار الگویی به نام [Post/Redirect/Get](#) هست که راه حلی رو برای اینکار پیشنهاد میده.

راه حل به این صورت هست که پس از پست شدن فرم به سمت سرور و انجام عملیات، بجای اینکه صفحه، دوباره با استفاده از متد GET به کاربر نشون داده بشه، کاربر Redirect بشه به صفحه. برای توضیح این مسئله به سراغ AccountController که بصورت پیش فرض وقتی یک پروژه ASP.NET MVC رو از نوع Internet ایجاد میکنید، وجود داره.

اکشن Register از نوع GET صفحه ثبت نام کاربر رو نمایش میده.

```
[HttpGet]
[AllowAnonymous]
public ActionResult Register()
{
    return View();
}
```

پس از اینکه کاربر اطلاعات داخل فرم رو پر کرد و به سمت سرور فرستاد و صحت اطلاعات فرستاده معتبر و عمل ثبت موفقیت آمیز بود برای ادامه کار به دو روش میتوان عمل کرد:

۱- کاربر به صفحه دیگری منتقل بشه و در اون صفحه پیام موفقیت آمیز بودن عملیات نشون داده بشه. مثلاً معمولاً پس از انجام عمل ثبت نام، کاربر به صفحه شخصی یا صفحه اصلی سایت منتقل میشه و یا در موقع ویرایش اطلاعات پیش از انجام عمل ویرایش کاربر به صفحه دیگری که لیستی از آیتمها که کاربر یکی از آنها را ویرایش کرده باز گردانده میشه.

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public ActionResult Register(RegisterModel model)
{
    if (ModelState.IsValid)
```

```

{
    // Attempt to register the user
    try
    {
        WebSecurity.CreateUserAndAccount(model.UserName, model.Password);
        WebSecurity.Login(model.UserName, model.Password);
        ViewBag.Message = "Successfully Registered!";

        // PRG has been maintained
        return RedirectToAction("Index", "Home");
    }
    catch (MembershipCreateUserException e)
    {
        ModelState.AddModelError("", ErrorCodeToString(e.StatusCode));
    }
}
// If we got this far, something failed, redisplay form
return View(model);
}

```

۲- نمایش دوباره صفحه ولی با تغییر هدر صفحه به کد 303 . کد 303 به مرورگر اعلام میکند صفحه ریدایرکت شده است

```

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public ActionResult Register(RegisterModel model)
{
    if (ModelState.IsValid)
    {
        // Attempt to register the user
        try
        {
            WebSecurity.CreateUserAndAccount(model.UserName, model.Password);
            WebSecurity.Login(model.UserName, model.Password);
            ViewBag.Message = "Successfully Registered!";

            // PRG has been maintained
            return RedirectToAction("Register");
        }
        catch (MembershipCreateUserException e)
        {
            ModelState.AddModelError("", ErrorCodeToString(e.StatusCode));
        }
    }
    // If we got this far, something failed, redisplay form
    return View(model);
}

```

در این حالت دوباره صفحه ثبت نام نمایش داده میشود ولی با زدن دکمه رفرش، اطلاعات دوباره به سمت سرور فرستاده نمیشود

نظرات خوانندگان

نویسنده: محمد رضا ابراهیم راد
تاریخ: ۱۳۹۲/۰۳/۰۷

با تشکر از مطلبی که نوشتید ولی من تو این کدها که 303 نمی‌بینم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۰۷

البته روش توضیح داده شده همان روش متداول PRG است. اگر حتما نیاز به 303 دارید به روش زیر باید عمل کرد:

```
using System.Web.Mvc;

namespace TestMvcPRG.Helper
{
    public class Redirect303 : ActionResult
    {
        private string _url;

        public Redirect303(string url)
        {
            _url = url;
        }

        public override void ExecuteResult(ControllerContext context)
        {
            context.HttpContext.Response.StatusCode = 303; // redirect using GET
            context.HttpContext.Response.RedirectLocation = _url;
        }
    }

    public abstract class BaseController : Controller
    {
        public Redirect303 Redirect303(string actionName)
        {
            return new Redirect303(Url.Action(actionName));
        }

        public Redirect303 Redirect303(string actionName, object routeValues)
        {
            return new Redirect303(Url.Action(actionName, routeValues));
        }

        public Redirect303 Redirect303(string actionName, string controllerName)
        {
            return new Redirect303(Url.Action(actionName, controllerName));
        }

        public Redirect303 Redirect303(string actionName, string controllerName, object routeValues)
        {
            return new Redirect303(Url.Action(actionName, controllerName, routeValues));
        }
    }
}
```

و بعد برای استفاده:

```
using System.Web.Mvc;
using TestMvcPRG.Helper;

namespace TestMvcPRG.Controllers
{
    public class HomeController : BaseController
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

```
[HttpPost]
public ActionResult Index(string data)
{
    if (ModelState.IsValid)
    {
        return Redirect303("Index"); // post-redirect-get
    }
    return View();
}
}
```