عنوان: **ساخت ActionResult سفارشی**
نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۱۱/۲۱ ۱۷:۵۵
آدرس: www.dotnettips.info
برچسب‌ها: Unit testing, ASP.Net MVC

پیشتر با انواع ActionResult آشنا شدید. حال فرض کنید می‌خواهید نوعی رو برگردونید که براش ActionResult موجود نباشه مثلا RSS و یا فایل از نوع Excel و...
خوب، فرض کنید می‌خواهید اکشن متدی رو بنویسید که قراره نام یک فایل متنی رو بگیره و انو تو مروگر به کاربر نمایش بده.
برای اینکار از کلاس ActionResult، کلاس دیگه‌ی رو بنام TextResult به ارث می‌بریم و از این ActionResult سفارشی شده، در اکشن متد مربوطه استفاده می‌کنیم:

```
public class TextResult : ActionResult
{
    public string FileName { get; set; }
    public override void ExecuteResult(ControllerContext context)
    {
        var filePath = Path.Combine(context.HttpContext.Server.MapPath(@"~/Files/"), FileName);
        var data = File.ReadAllText(filePath);
        context.HttpContext.Response.Write(data);
    }
}
```

نحوه استفاده

```
public ActionResult DownloadTextFile(string fileName)
    {
        return new TextResult { FileName = fileName };
    }
```

در واقع متد اصلی اینجا ExecuteResult هست که نتیجه‌ی کار یک اکشن رو می‌تونیم پردازش کنیم.
خوب، سوالی که اینجا پیش میاد اینه که چرا این همه کار اضافی، چرا از Return File استفاده نمی‌کنی؟

```
public ActionResult DownloadTextFile(string fileName)
    {
        var filePath = Path.Combine(HttpContext.Server.MapPath(@"~/Files/"), fileName);
        return File(filePath, "text");
    }
```

یا کلا دلیل استفاده از ActionResult سفارشی چیه؟

جلوگیری از پیچیدگی و تکرار کد
همیشه کار مثل مورد بالا راحت و کم کد! نیست.
به مثال زیر توجه کنید که قراره خروجی CSV بهمون بده.

```
public class CsvActionResult : ActionResult
{
    public IEnumerable ModelListing { get; set; }

    public CsvActionResult(IEnumerable modelListing)
    {
        ModelListing = modelListing;
    }

    public override void ExecuteResult(ControllerContext context)
    {
        byte[] data = new CsvFileCreator().AsBytes(ModelListing);
        var fileResult = new FileContentResult(data, "text/csv")
        {
            FileDownloadName = "CsvFile.csv"
        };
        fileResult.ExecuteResult(context);
    }
```

```
}
```

و نحوه استفاده:

```
public ActionResult ExportUsers()
    {
        IEnumerable<User> model = UserRepository.GetUsers();
        return new CsvActionResult(model);
    }
```

حال فرض کنید بخواهیم همه این کدها رو داخل اکشن متد داشته باشیم، یکم پیچیده میشه و یا فرض کنید کنترلر دیگه‌ای نیاز به خروجی CSV داشته باشه، تکرار کد زیاد میشه.

راحت کردن گرفتن تست واحد از اکشن‌ها متدها

کاربرد ActionResult سفارشی تو تست واحد اینه که وابستگی‌های یک اکشن رو که Mock کردنش سخته می‌بریم داخل ActionResult و هنگام نوشتن تست واحد درگیر کار با اون وابستگی نمی‌شیم.
به مثال زیر توجه کنید که قراره برای اکشن Logout تست واحد بنویسیم
ابتدا بردن وابستگی‌ها به خارج از اکشن به کمک ActionResult سفارشی

```
public class LogoutActionResult : ActionResult
{
    public RedirectToRouteResult ActionAfterLogout { get; set; }
    public LogoutActionResult(RedirectToRouteResult actionAfterLogout)
    {
        ActionAfterLogout = actionAfterLogout;
    }
    public override void ExecuteResult(ControllerContext context)
    {
        FormsAuthentication.SignOut();
        ActionAfterLogout.ExecuteResult(context);
    }
}
```

نحوه استفاده از ActionResult سفارشی

```
public ActionResult Logout()
    {
        var redirect = RedirectToAction("Index", "Home");
        return new LogoutActionResult(redirect);
    }
```

و سپس نحوه تست واحد نوشتن

```
[TestMethod]
    public void The_Logout_Action_Returns_LogoutActionResult()
    {
        //arrange
        var account = new AccountController();

        //act
        var result = account.Logout() as LogoutActionResult;

        //assert
        Assert.AreEqual(result.ActionAfterLogout.RouteValues["Controller"], "Home");
    }
```

خوب به راحتی ما میایم فراخوانی متد SignOut رو از داخل اکشن می‌کشیم بیرون و این کار از اجرای متد SignOut از داخل اکشن متد جلوگیری می‌کنه و همچنین با این کار هنگام تست واحد نوشتن نیاز نیست با Mock کردن کلاس FormsAuthentication سروکار داشته باشیم و فقط کافیه چک کنیم خروجی از نوع LogoutActionResult هست یا خیر و یا می‌تونیم ActionAfterLogout

رو چک کنیم.

منابع و مراجع: + و +