

## یک مثال ساده برای شرح مساله

در اینجا مدل User، کنترلری به نام Home و View متناظر با آن را ملاحظه می‌کنید:

```
namespace ModelStateTest.Models
{
    public class User
    {
        public string Email { set; get; }
    }
}
```

```
using System.Web.Mvc;
using ModelStateTest.Models;

namespace ModelStateTest.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(User model)
        {
            model.Email = "?";
            return View(model);
        }
    }
}
```

```
@model ModelStateTest.Models.User

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)

    <fieldset>
        <legend>User</legend>

        <div class="editor-label">
            @Html.LabelFor(model => model.Email)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.Email)
            @Html.ValidationMessageFor(model => model.Email)
        </div>

        <p>
            <input type="submit" value="Create" />
        </p>
    </fieldset>
}
```

نکته‌ای که در اینجا مدنظر است، سطر زیر می‌باشد:

```
model.Email = "?";
```

پیش از اینکه برنامه را اجرا کنید، به نظر شما پس از postback به سرور، چه اطلاعاتی در Html.EditorFor تعریف شده در View برنامه نمایش داده خواهد شد؟  
احتمالا عنوان می‌کنید که خوب ... همان مقدار علامت سؤال انتساب داده شده. اما ... اینچنین نیست! دقیقا همان مقداری که در Postback به سرور ارسال شده، نمایش داده می‌شود.  
این مورد نکته‌ای است که عدم آشنایی با آن ممکن است چندین ساعت را به دیباگ یک برنامه اختصاص دهد، بدون اینکه نتیجه مفیدی حاصل شود.

[مطابق نظر](#) طراحان اصلی ASP.NET MVC، اینکار و این رفتار، دقیقا به همین نحو صحیح است و باگ نیست.  
«فرض کنید در فیلدی عددی، کاربر عبارت «تست» را وارد کرده است. نیاز است در خطای اعتبارسنجی پس از Postback به او عنوان کنیم، لطفا بجای «تست»، عدد وارد کنید. چون خاصیت متناظر قید شده در مدل، عددی است، مقدار «تست» وارد شده را از دست خواهیم داد. به همین جهت همان مقدار اولیه وارد شده را در HTML Helpers پس از Postback حفظ می‌کنیم.»

راه حل‌های ممکن، برای به روز رسانی وضعیت مدل پس از Postback

#### الف) استفاده از متد ModelState.Clear

این متد کلیه داده‌های موجود در ModelState را منجمله خطاهای حاصل از اعتبارسنجی، حذف می‌کند. در این حالت مطابق مثال فوق پس از Postback، مقدار علامت سؤال نسبت داده شده به خاصیت ایمیل، نمایش داده خواهد شد.

#### ب) استفاده از متد ModelState.Remove

```
this.ModelState.Remove("Email");
```

این حالت نیز مانند حالت الف است، با این تفاوت که اطلاعات اعتبارسنجی و سایر موارد مرتبط را حذف نمی‌کند.

#### ج) عدم استفاده از HTML Helpers

این مورد را فقط با متدهای کمکی For دار، مانند Html.EditorFor مشاهده خواهید کرد. اگر نحوه تعریف را به شکل زیر تغییر دهیم، نیازی به استفاده از متد ModelState.Remove نخواهد بود. البته، مزیت‌های استفاده از HTML Helpers دارای متدهای For دار را که Strongly typed هستند، از دست می‌دهیم.

```
<input type="text" name="Email" id="Email" value="@Model.Email" />
```