

[در نگارش‌های قبلی](#) ASP.NET Web forms اگر نیاز به ارسال محتوای HTML ایی وجود داشت، می‌بایستی کل سیستم اعتبارسنجی حداقل یک صفحه را غیرفعال کرد. برای مثال:

```
<%@ Page Language="C#" ValidateRequest="false" %>
```

این نقیصه‌ی همه یا هیچ، در ASP.NET MVC وجود ندارد و می‌توان به ازای یک خاصیت خاص، اعتبارسنجی پیش فرض را با اعمال ویژگی AllowHtml موقتا غیرفعال کرد؛ اما مابقی فیلدها و خاصیت‌های فرم همچنان تحت نظر سیستم اعتبارسنجی‌های ورودی ASP.NET قرار خواهند داشت و به این ترتیب امکان ورود اطلاعات خطرناک، خصوصا از لحاظ مباحث XSS، حداقل در آن فیلدها وجود نخواهد داشت.

در ASP.NET 4.5 مفهوم جدیدی به نام Deferred validation معرفی شده‌است تا رفتار Web forms را نیز در برابر ورودی‌های ارسال شده همانند ASP.NET MVC کند. به این معنا که اگر جهت دسترسی به مقدار کوئری استرینگ lastName همانند قبل عمل کنید:

```
Request["lastName"]
```

و کاربر ورودی خطرناکی را وارد کرده باشد، همچنان استثنای A potentially dangerous request.form value was detected صادر می‌شود.

اما اگر در ASP.NET 4.5، کوئری استرینگ را به نحو ذیل دریافت کنید:

```
Request.Unvalidated.QueryString["lastName"]
```

به صورت خودکار سیستم اعتبارسنجی غیرفعال شده و امکان دسترسی به محتوای اصلی کوئری استرینگ lastName را خواهید داشت. به این ترتیب دیگر نیازی نخواهید داشت تا اعتبارسنجی کل صفحه را برای دسترسی به یک مقدار خاص غیرفعال نمائید. برای دسترسی به مقادیر اعتبارسنجی نشده فیلدهای یک فرم ارسالی نیز می‌توان به صورت ذیل عمل کرد:

```
Request.Unvalidated.Form[txtData1.UniqueID]
```

کلاس [Request.Unvalidated](#) شامل خاصیت‌های Cookies، Files، Headers و امثال آن نیز می‌باشد. در اینجا اگر دقت کرده باشید از UniqueID بجای Id استفاده شده‌است؛ از این جهت که مجموعه Form، حاوی Idهای واقعی دریافت شده از کاربر مانند ct100\$MainContent\$MessageText می‌باشد.

روش دوم، استفاده از خاصیت جدید [ValidateRequestMode](#) یک کنترل سمت سرور است و در اینجا نیز می‌توان صرفا اعتبارسنجی یک کنترل را بجای کل صفحه، غیرفعال کرد:

```
<asp:TextBox ID="txtASPNet" ValidateRequestMode="Disabled" runat="server" />
```

تنظیم خاصیت ذکر شده برای کنترل‌های سمت سرور ضروری است. از این جهت که در حین تشکیل ViewState از محتوای این کنترل‌ها نیز استفاده می‌شود. اینجا است که اگر ValidateRequestMode غیرفعال نشده باشد، باز همان خطای ورودی خطرناک را دریافت خواهید کرد.

البته برای فعال سازی اعتبارسنجی با تاخیر نیاز است اصلاح ذیل را نیز به web.config اعمال کنید ( [مقدار پیش فرض آن](#) 4 است):

```
<httpRuntime targetFramework="4.5" requestValidationMode="4.5" />
```