

چندین نمونه استفاده از jQuery Ajax در ASP.NET Webforms را در این سایت می‌توانید پیدا کنید؛ برای مثال:

[بارگذاری یک یوزرکنترل با استفاده از جی‌کوئری](#)

[بررسی وجود نام کاربر با استفاده از jQuery Ajax در ASP.Net](#)

[استفاده از افزونه‌ی jQuery Autocomplete در ASP.NET](#)

[jQuery Ajax و نحوه صحیح ارسال مقادیر به یک وب سرویس](#)

[استفاده از jQuery Ajax جهت تعیین اعتبار یک فرم](#)

سؤالی که در تمام این موارد حائز اهمیت است این مورد می‌باشد که "از کجا متوجه شوم وب سرویس مورد استفاده واقعا توسط اسکریپت سایت جاری فراخوانی شده و نه توسط یک برنامه‌ی خارجی؟"

در اینجا می‌توان از سورس‌های ASP.NET MVC کمک گرفت : ( [+](#) ). همان متد IsAjaxRequest را در ASP.NET Webforms هم می‌شود استفاده کرد:

```
public static bool IsAjaxRequest(this HttpRequestBase request)
{
    if (request == null)
    {
        throw new ArgumentNullException("request");
    }

    return (request["X-Requested-With"] == "XMLHttpRequest") ||
           ((request.Headers != null) && (request.Headers["X-Requested-With"] == "XMLHttpRequest"));
}
```

حاصل IsAjaxRequest باید در ابتدای تمام درخواست‌های رسیده بررسی شود. البته باید دقت داشت که این بررسی را به آسانی می‌توان دور زد (چون بر اساس هدرهای رسیده است)، اما باز هم بهتر از هیچ نوع نظارتی می‌باشد.

## نظرات خوانندگان

نویسنده: وحید نصیری  
تاریخ: ۱۸:۴۱:۱۲ ۱۳۹۰/۰۱/۱۵

جهت تکمیل: در اینجا HttpRequestBase همان HttpContext.Current.Request می‌باشد.

نویسنده: ایمان اسلامی  
تاریخ: ۲۰:۰۸ ۱۳۹۱/۰۴/۱۳

با تشکر از مطالب پر بارتون  
مهندس جهت تکمیل این بحث و کامل شدن مسئله امنیتی می‌توان این مطلب شما را نیز به عنوان تکمیل کننده این بحث در نظر گرفت؟

[حذف هدرهای مربوط به وب سرور از یک برنامه‌ی ASP.Net](#)

نویسنده: وحید نصیری  
تاریخ: ۲۰:۱۴ ۱۳۹۱/۰۴/۱۳

حذف هدر وب سرور بیشتر جهت مقابله با اسکنرهای خودکار می‌تونه کاربرد داشته باشه و عموم این اسکنرها هم کاری به Ajax ندارند یا درکی از آن ندارد (تا جایی که بررسی کردم).

نویسنده: ایمان اسلامی  
تاریخ: ۲۰:۲۰ ۱۳۹۱/۰۴/۱۳

پس برای تکمیل این بحث که نقطه ضعفش طبق اشاره شما همان هدرهای رسیده می‌باشد نظارت دیگری هم میشه انجام داد؟

نویسنده: وحید نصیری  
تاریخ: ۲۰:۳۳ ۱۳۹۱/۰۴/۱۳

هدر رسیده نقطه ضعف نیست. نقطه قوت است. چون بر اساس آن (تا حدودی) مشخص می‌شود که مرورگر درخواست را ارسال کرده یا یک ربات.  
بررسی referrer هم خوب است.  
بررسی host ارسال کننده درخواست هم مفید است. تمام این‌ها در هدرهای رسیده لحاظ می‌شوند.

نویسنده: ایمان اسلامی  
تاریخ: ۲۳:۳۷ ۱۳۹۱/۰۴/۱۳

ممنونم. مثل همیشه عالی.

نویسنده: خیام  
تاریخ: ۲:۳۲ ۱۳۹۲/۱۱/۲۵

برای اینکه کاملاً مطمئن بشیم که وب سرویس توسط اسکریپت سایت جاری فراخوانی میشه ، بجای این روشی که شما فرمودین راه دیگه ای هست ؟ با توجه به گفته شما باز هم راه برای دور زدن آن هست و اینکه خود ASP.NET MVC چطور مدیریت کامل روی این موضوع دارد ؟

به نظر شما استفاده از هندلرها در چنین مواقعی که مثلن قرار است از بانک اطلاعاتی واکنشی داشته باشیم بهتر است ؟ یا همان وب سرویس استفاده کنیم ؟

نویسنده: وحید نصیری

- علاوه بر مطلب و نکاتی که در کامنت‌ها مطرح شده، روش استفاده از [آنتی‌فورجری توکن](#) هم مفید است. ترکیبی است از یک فیلد مخفی به همراه یک کوکی رمزنگاری شده.
- بهتر است بجای این‌ها از [ASP.NET Web API](#) استفاده کنید.