

## بخش هایی از کتاب "مرجع کامل ASP.NET MVC (با پوشش کامل ASP.NET MVC 4)"

ترجمه و تالیف: بهروز راد

وضعیت: در دست چاپ

## Web API چیست؟

Web API، نوع قالب جدیدی برای پروژه‌های مبتنی بر وب در .NET است که بر مبنای اصول و الگوهای موجود در ASP.NET MVC ساخته شده است و همراه با ASP.NET MVC 4 وجود دارد. Web API توسعه گران را قادر می‌سازد تا با استفاده از یک الگوی ساده که در Controllerها پیاده سازی می‌شود، وب سرویس‌های مبتنی بر پروتوکل HTTP را با کدها و تنظیمات کم ایجاد کنند. این سبک جدید برای ایجاد وب سرویس‌ها، می‌تواند در انواع پروژه‌های .NET مانند ASP.NET MVC، ASP.NET Web Forms، Windows Application و ... استفاده شود.

یک سوال کاملاً منطقی در اینجا به وجود می‌آید. چرا نیاز به بستری جدید برای ایجاد وب سرویس داریم؟ آیا در حال حاضر مایکروسافت بستری محبوب و فراگیر برای توسعه‌ی وب سرویس‌هایی که بتوانند با پروتوکل SOAP تعامل داشته باشند در اختیار ندارد؟ مگر وب سرویس‌های ASMX از زمان معرفی ASP.NET وجود نداشته‌اند؟ آیا تکنولوژی WCF مایکروسافت، بیشترین انعطاف پذیری و قدرت را برای تولید وب سرویس‌ها در اختیار قرار نمی‌دهد؟ وب سرویس‌ها جایگاه خود را یافته‌اند و توسعه گران با تکنولوژی‌های موجود به خوبی آنها را پیاده سازی و درک می‌کنند. چرا Web API؟

## چرا Web Api؟

برای پاسخ به این سوال، باید برخی مشکلات را بررسی کنیم و ببینیم ابزارهای موجود چه راه حلی برای آنها در نظر گرفته‌اند. اگر با گزینه‌هایی که در ادامه می‌آیند موافق هستید، خواندن این مطلب را ادامه دهید، و اگر اعتقادی به آنها ندارید، پس نیازهای شما به خوبی با بسترهای موجود پاسخ داده می‌شوند. من معتقد هستم که راه بهتری برای ایجاد وب سرویس‌ها وجود دارد. من معتقد هستم که روش‌های ساده‌تری برای ایجاد وب سرویس‌ها وجود دارد و WCF بیش از حد پیچیده است. من معتقد هستم که تکنولوژی‌های پایه‌ی وب مانند GET، POST، PUT و DELETE برای انجام اعمال مختلف توسط وب سرویس‌ها کافی هستند. اگر همچنان در حال خواندن این مطلب هستید، توضیحات خود را با شرح تفاوت میان Web API و تکنولوژی‌های دیگر هم حوزه‌ی آن ادامه می‌دهیم و خواهید دید که استفاده از Web API چقدر آسان است.

## تفاوت WCF و Web API

وب سرویس‌های ASMX تا چندین سال، انتخاب اول برای ایجاد وب سرویس‌های مبتنی بر پروتوکل SOAP با استفاده از پروتوکل HTTP بودند. وب سرویس‌های ASMX، از وب سرویس‌های ساده که نیاز به قابلیت تعامل پایین داشتند و در نتیجه به پروتوکل SOAP نیز وابسته نبودند پشتیبانی نمی‌کردند. WCF جای وب سرویس‌های ASMX را گرفت و خود را به عنوان آخرین و بهترین روش برای ایجاد وب سرویس‌ها در بستر .NET معرفی کرد. نمونه‌ای از یک سرویس WCF بر مبنای پروتوکل HTTP در .NET به صورت ذیل است.

```
[ServiceContract]
public interface IService1
{
    [OperationContract]
    string GetData(int value);
    [OperationContract]
    CompositeType GetDataUsingDataContract(CompositeType composite);
}
...
public class Service1 : IService1
{
    public string GetData(int value)
    {
        return string.Format("You entered: {0}", value);
    }
}
```

```

    }
    public CompositeType GetDataUsingDataContract(CompositeType composite)
    {
        if (composite == null)
        {
            throw new ArgumentNullException("composite");
        }
        if (composite.BoolValue)
        {
            composite.StringValue += "Suffix";
        }
        return composite;
    }
}

```

در WCF، پایه و اساس وب سرویس را یک interface تشکیل می‌دهد. در حقیقت اجزای وب سرویس را باید در یک interface تعریف کرد. هر یک از متدهای وب سرویس در interface تعریف شده که صفت OperationContract برای آنها در نظر گرفته شده باشد، به عنوان یکی از اعمال و متدهای قابل فراخوانی توسط استفاده کننده از وب سرویس در دسترس هستند. سپس کلاسی باید ایجاد کرد که interface ایجاد شده را پیاده سازی می‌کند. در قسمت بعد، با مفاهیم پایه‌ی Web API و برخی کاربردهای آن در محیط ASP.NET MVC آشنا می‌شوید.

### نتیجه گیری

Web API، یک روش جدید و آسان برای ایجاد وب سرویس‌ها، بر مبنای مفاهیم آشنای ASP.NET MVC و پایه‌ی وب است. از این روش می‌توان در انواع پروژه‌های .NET استفاده کرد.

## نظرات خوانندگان

نویسنده: یوسف نژاد  
تاریخ: ۱۱:۵۹ ۱۳۹۱/۰۴/۱۱

این مقوله خیلی مفیده و کاربردی هست. خیلی وقت بود میخواستم در موردش بیشتر تحقیق کنم. با تشکر بابت زحماتتون و آغاز این سری مطلب جدید.

نویسنده: شهرز جعفری  
تاریخ: ۱۶:۷ ۱۳۹۱/۰۴/۱۱

زمان انتشار کی هس؟

نویسنده: بهروز راد  
تاریخ: ۱۹:۵ ۱۳۹۱/۰۴/۱۱

بستگی به ناشر داره. اما نباید بیشتر از دو هفته طول بکشه.

نویسنده: شهرز جعفری  
تاریخ: ۱۹:۸ ۱۳۹۱/۰۴/۱۱

نظر کلی من اینه که همیشه به wcf گفت پیچیده آخه هدفش فرق میکنه. و در ضمن مگه API حدوداً همون Rest با معماری خیلی ساده تر نیست؟

نویسنده: شهرز جعفری  
تاریخ: ۱۹:۸ ۱۳۹۱/۰۴/۱۱

بیصبرانه منتظرش هستم

نویسنده: بهروز راد  
تاریخ: ۲۲:۴۷ ۱۳۹۱/۰۴/۱۱

REST بیشتر برای مواقعی هست که شما عملیات CRUD انجام میدید. در حالی که با Web API میتونید علاوه بر CRUD، کارهای بسیار بیشتری انجام بدید. مفاهیم و قابلیت های موجود در ASP.NET MVC مانند فیلترها به خوبی در Web API پشتیبانی و به راحتی قابل استفاده هستند. ضمن اینکه با Web API میتونید معماری REST رو با تغییر کوچکی در route پیش فرض به دست بیارید و بدین شکل، مهاجرت از REST به Web API بسیار راحت هست. در اوایل معرفی Web API، از پروتوکول OData نیز پشتیبانی اولیه میشد که متأسفانه مایکروسافت در نسخه ی RC این پشتیبانی رو حذف کرد. شاید در نسخه های بعدی این قابلیت نیز اضافه بشه که به قدرتمندتر شدن Web API کمک می کنه.

ضمناً، پشتیبانی مایکروسافت از WCF REST API نیز به اتمام رسیده و پیشنهاد شده که از Web API استفاده کنید.

<http://aspnet.codeplex.com/wikipage?title=WCF%20REST>

نویسنده: رضا.ب  
تاریخ: ۳:۳ ۱۳۹۱/۰۴/۱۲

در بند سوم اشاره کردین : من معتقد هستم که تکنولوژی های پایه ی وب مانند GET، POST، PUT و DELETE برای انجام اعمال مختلف توسط وب سرویس ها کافی هستند. اگر ضروری نیستند بیشتر از CRUD باشند پس خاصیت ویژه ای که شما میگین "کارهای بسیار بیشتری" میتونه انجام بده چی هست که WCF پاسخگو نیست؟

در ضمن فکر میکنم REST فقط با منابع و ورب های HTTP کار داره. و برای همین سهولت و سادگی پروتکول SOAP نسخه منسوخ

شده‌ی وب‌سرورها به حساب میاد. اینطور نیست؟

سوال دیگه‌ام در مورد میزان نقش Web API هست. آیا رسالت واقعی یک وب‌سرویس رو هدف گرفته؟ یعنی پیاده سازی یک Endpoint که شامل یه سری interface هستند که امکاناتی رو در اختیار کلاینت قرار میده؟ ممنون از توجه‌تون.

نویسنده: بهروز راد  
تاریخ: ۱۳۹۱/۰۴/۱۲ ۸:۳۷

دقت داشته باشید که Web API عرضه نشده تا WCF رو منسوخ کنه. برنامه هایی که صرفاً از بستر پروتوکل HTTP به عنوان یک سرویس برای رد و بدل کردن داده‌ها استفاده می‌کنند، بهتره که از این به بعد از Web API استفاده کنند. ضمن سادگی و مفاهیم آشنای ASP.NET MVC، روش یکپارچه ای برای ایجاد وب سرویس‌های HTTP نیز به وجود اومده که مشکلات استفاده از WCF رو از بین می‌بره. WCF ذاتاً برای پیغام‌های SOAP محور طراحی شده و به کار گرفتن اون برای وب سرویس‌های HTTP یا به زور خوراندن HTTP به اون بی معنیه. در WCF راه‌های مختلفی برای ایجاد وب سرویس‌های HTTP وجود داره که باعث گمراهی و سردرگمی توسعه گر میشه و حتی فریمورک‌های مختلفی مانند OpenRasta و ServiceStack نیز بدین منظور وجود دارند. بنابراین پشتیبانی WCF از HTTP به یک پروژه‌ی دیگه تحت نام ASP.NET Web API منتقل شده و WCF Web API دیگه پشتیبانی نمیشه. کمی تغییر نام و کمی جابجایی مفاهیم در اینجا صورت گرفته. WCF همچنان قدرتمنده و نباید Web API به هیچ وجه به عنوان جایگزینی برای اون تصور بشه. ایجاد بسترهایی برای ارتباطات دو طرفه یا صفی از پیغام‌ها یا سوئیچ بین کانال‌ها در هنگام فعال نبودن یک کانال، اینها همه از قابلیت‌هایی هست که Web API هرگز جایگزینی برای اونها نخواهد بود و مختص WCF هستند.

نویسنده: ramini  
تاریخ: ۱۳۹۱/۰۴/۱۲ ۹:۲۶

سلام آقای راد  
بخشید که سوال بی ربط رو اینجا میپرسم  
آیا برنامه ای برای انتشار ویرایش جدید کتاب Entity Framework دارین؟

نویسنده: بهروز راد  
تاریخ: ۱۳۹۱/۰۴/۱۲ ۱۰:۱۴

لطفاً سوالات اینچنینی رو از طریق ایمیل behrouz.rad[at]signlmail بپرسید.  
بله، بعد از کتاب ASP.NET MVC، کتاب Entity Framework رو آپدیت می‌کنم.

نویسنده: torisoft  
تاریخ: ۱۳۹۱/۰۴/۱۶ ۲۳:۵

سلام جناب راد  
از Web API تو سیلورلایت هم میشه استفاده کرد ؟  
اگه استفاده میشه آیا مثبت میدونید استفاده از اونو تو سیلور ؟  
با تشکر

نویسنده: بهروز راد  
تاریخ: ۱۳۹۱/۰۴/۱۷ ۸:۱۱

بله مشکلی نداره. پروژه‌ی Silverlight رو در یک پروژه‌ی وب Host کنید.  
Silverlight هم یک نوع پروژه است، مثل Web و Desktop. اگر پروژه‌ی شما بر مبنای Silverlight هست و نیاز دارید تا امکانات اون رو به صورت سرویس ارائه بدید، می‌تونید از Web API برای عرضه‌ی این امکانات استفاده کنید.

نویسنده: حمید  
تاریخ: ۱۳۹۱/۰۴/۱۷ ۱۱:۲۲

سلام. وقت بخیر.

مطالب خیلی خوب و به روزی دارین و خدا قوت..

با عرض معذرت می‌خواستم بگم من MVC4 رو نصب کردم اما بازم بعد انتخاب MVC4 از لیست Template های ویژوال استودیو گزینه Web API رو مشاهده نمی‌کنم. آیا افزونه یا برنامه خاصی باید نصب کنم. از قبل از زحمتتون تشکر می‌کنم.

نویسنده:

وحید نصیری

تاریخ:

۱۱:۵۰ ۱۳۹۱/۰۴/۱۷

مراجعه کنید به [قسمت دوم](#) ، تصویر سوم

نویسنده:

حمید

تاریخ:

۲۲:۵۷ ۱۳۹۱/۰۴/۲۰

سلام. مشکل من همینکه که همین تصویر سوم رو که می‌گین تو این بخش من گزینه Web API رو ندارم.

نویسنده:

وحید نصیری

تاریخ:

۲۳:۱۱ ۱۳۹۱/۰۴/۲۰

از طریق NuGet هم می‌تونید برای نصب آن اقدام کنید. این رو هم تست کنید:

<http://nuget.org/packages/aspnetwebapi>

نویسنده:

Saeed M. Farid

تاریخ:

۱۱:۳ ۱۳۹۱/۰۴/۲۱

سلام و ممنون از مطلب مفید:

امکانش هست در مورد "سویچ بین کانال‌ها در هنگام فعال نبودن یک کانال" کمی بیشتر راهنمایی کنید یا مرجع (لینک) معرفی کنید؟ من از صحبت شما اینطور برداشت کردم که میشه در [channel shape](#) (های)ی که مثلاً برای duplex communications (یعنی [IDuplexChannel](#)) پیاده سازی کردم، اگه چنین کانالی در دسترس نبود، سوئیچ کنه روی طراحی مبتنی بر one-way messaging من؟ اصلاً چنین امکانی در سطح [IChannelListener](#) هست یا [ChannelFactory](#)؟ کلاً اگه ممکنه یه توضیح کلی در مورد چنین امکانی که در موردش صحبت کردین بدین یا اگه جایی در موردش قبلاً بحث شده (که حتماً شده!) من رو هدایت کنید به اون، چون گلوگاه سیستم‌هام همین مورد هست. پیشاپیش ازتون ممنونم...

نویسنده:

بهروز راد

تاریخ:

۱۹:۲۹ ۱۳۹۱/۰۴/۲۱

اصولاً در Web API چیزی با عنوان Channel با اون مفهوم که در WCF هست نداریم. در Web API فقط یک Transport Channel برای HTTP وجود داره، چون هدف ایجاد Web API، فقط برقراری ارتباط در سطح HTTP هست، نه مثلاً MSMQ. Protocol Channel هم همان مفاهیمی هستند که در ASP.NET MVC وجود دارند و مثلاً قسمتی از اون، تصدیق هویت و تعیین مجوز کاربر برای دسترسی به منابع با استفاده از فیلتر Authorize هست. لطفاً دنبال تطبیق و تناظر بین مفاهیم پیچیده‌ی WCF و یافتن معادل در Web API نباشید. Web API به وجود آمده تا ایجاد وب سرویس‌ها در بستر HTTP رو ساده کنه، همین!

نویسنده:

سیروس

تاریخ:

۱۸:۳۳ ۱۳۹۱/۱۲/۰۲

سلام

یک سوال مهم داشتم، آیا استفاده از web api در Windows Form مانند WCF ممکن است، یعنی پروژه ما هم هاست و هم کلاینت رو MVC یا ASP.Net نیست، اگه میشه یه منبع معرفی کنید.

نویسنده: محسن  
تاریخ: ۱۳۹۱/۱۲/۰۲ ۲۲:۳۶

اولین نتیجه جستجوی گوگل در مورد winforms web api :

[Using Microsoft Web API from a Windows and WinRT Client Application](#)

نویسنده: محمد آزاد  
تاریخ: ۱۳۹۱/۱۲/۰۲ ۲۲:۳۸

تو مقدمه به این مطلب اشاره شده دوست عزیز  
این سبک جدید برای ایجاد وب سرویس ها، می تواند در انواع پروژه های .NET. مانند ASP.NET MVC, ASP.NET Web Forms, Windows Application و ... استفاده شود.

نویسنده: سیروس  
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۰:۱۰

محسن <= دوست عزیز من اون مطلب رو قبلا هم مطالعه کردم، قسمت هاست رو MVC ست. اینقدر بی سواد نیستم که نتونم سرچ کنم.  
آزاد <= میدونم که تو Win APP قابل استفاده هست، اما می خوام بدونم پروژه هاست مثل WCF می تونه رو مستقل از Asp.Net باشه یا نه چون ظاهرا پیاده سازی WebAPI فقط روی ASP.Net امکان پذیر است.

نویسنده: محسن  
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۰:۱۶

نتیجه جستجوی گوگل در مورد [wep api self host](#) :

[Self-Host a Web API](#)

نویسنده: یاسر مرادی  
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۲:۳۶

من فکر کنم مطلب این دوستان رو این جوری مطرح کنم بهتره  
وقتی شما از WCF Data Services استفاده می کنید، WCF Data Services Client دارید، که به شما امکان نوشتن کوئری های Linq در سمت کلاینت، Merge و Change Tracking و ... رو می ده  
اما من همچین آیتی رو برای Web API پیدا نکردم، بهترین چیزی که دیدم Http Client بوده که در حد مثال زدن خوبه، ولی به درد پروژه نویسی نمی خوره، این که شما یک کلاینت قوی داشته باشید، خیلی مهمه، Http Client تفاوت مفهومی زیادی با \$.ajax نداره  
حتی در JayData هم همین طور هستش، و شما پشتیبانی خیلی بهتری از WCF Data Services می بینید تا از Web API، همین طور در Breeze.js در اندروید و iOS هم شما پشتیبانی WCF Data Services Client رو دارید، ولی Web API خیر موفق باشید

نویسنده: محسن  
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۲:۵۸

سؤال مطرح شده در مورد هاست کردن یک سرویس در برنامه ویندوزی بود که اصطلاحاً Self hosting نام دارد.

Web API امکان استفاده از OData را هم دارد:

### [Getting started with ASP.NET Web API OData in 3 simple steps](#)

نویسنده: یاسر مرادی  
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۳:۳۵

قبول، ولی در هر حال آیا راهی جز Http Client برای دسترسی به Web API وجود دارد ؟

مثلاً مدل Linq به OData ؟  
به همراه Change Tracking و ...  
در ضمن موارد مهمی از OData مانند \$batch در Web API پشتیبانی نشده اند، و باید برایشان Message Formatter نوشت، این نیز کار را سخت می‌کند  
بر خلاف نظر دوستان به نظر من به هیچ وجه هیچ فریم ورکی راحت‌تر از WCF Data Services وجود ندارد، که جمعا با 3 خط کد راه اندازی می‌شود.

نویسنده: محسن  
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۷:۲۴

NuGet مربوط به [Web API OData](#) مرتباً به روز میشه. آخرین به روز رسانی آن مربوط به 5 روز قبل بوده.

ضمن اینکه خروجی OData استاندارد است. بنابراین با کلاینت‌های موجود کار می‌کنه. فرقی نمی‌کنه تولید کننده چی هست تا زمانیکه استاندارد رعایت بشه.

نویسنده: یاسر مرادی  
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۸:۰۵

دوست عزیز، فکر کنم سوال من خیلی واضح باشه  
مسئله اول این هستش که مواردی از OData هست که در WCF Data Services وجود داره، ولی در Web API خیر، OData یک سری استاندارد هستش، بالاخره باید یک جایی پیاده سازی بشه، مثل HTML 5، که قسمت‌های مختلفش در درصدهای متفاوت در مرورگرهای متفاوت پیاده سازی شده، در این میان Chrome بهتر از IE هستش، چرا ؟ چون استانداردهای بیشتری رو پیاده سازی کرده  
دوم این که آیا شما به صورت عملی از js Breeze و Jay Data و WCF Data Services Client استفاده کرده اید ؟ درسته که اینها به OData وصل می‌شوند، ولی میزان امکانات اینها برای WCF Data Services قابل قیاس با Web API نیست.  
سوال اصلی من با این تفاسیر این است :  
اگر قبول کنیم که راهی برای دسترسی به Web API وجود ندارد، الا استفاده از jQuery Ajax و Http Client، شما به چه صورت یک پروژه بزرگ رو با Web API می‌نویسید ؟  
Change Tracking رو چه جوری پیاده سازی می‌کنید ؟  
به چه صورت در کلاینت‌هایی مانند اندروید، و یا Win RT و ... از Linq برای دسترسی به سرویس هاتون استفاده می‌کنید ؟  
اگر فرض کنیم که می‌خواهیم یک سرویس عمومی بنویسیم که همه جا به سادگی قابل استفاده باشه، آیا از Web API استفاده می‌کنید ؟  
خلاصه : مزیت واقعی Web API چیست و چه زمانی پروژه ای رو با Web API شروع می‌کنید ؟  
موفق و پایدار باشید

نویسنده: محسن  
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۸:۳۱

«مزیت واقعی Web API چیست و چه زمانی پروژه ای رو با Web API شروع می کنید؟»

[WCF or ASP.NET Web APIs](#)

به علاوه هدف اصلی Web API و یکپارچگی آن با خصوصاً MVC (و بعد وب فرمها) در درجه اول توسعه ActionResult های پیش فرض MVC است (به همین جهت اول اسم آن ASP.NET است و نه مثلاً اندروید):

[ASP.NET Web API vs. ASP.NET MVC APIs](#)

نویسنده: یاسر مرادی  
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۹:۳

مقاله اول Web API رو با WCF خام مقایسه کرده، نه با WCF Data Services

مقاله دوم هم Action های Web API رو با MVC قیاس کرده  
اگر شما یک مقاله بنویسید که مثلاً Web API رو با ASP.NET Web Service قیاس بکنه، و نشون بده مزیت های Web API بیشتره، این می شه مزیت Web API بر ASP.NET Web Service، نه بر WCF Data Services  
ممکنه این موارد هم مهم باشند، ولی اون چیزی که برای من سوال شده این هستش که چه زمانی در یک پروژه WCF Data Services رو می گذاریم کنار و از Web API استفاده می کنیم؟  
در واقع با توجه به امکانات واقعاً زیاد WCF Data Services چرا باید اساساً از Web API استفاده بشه، اگر شما می فرمایید که 5 روز پیش برای Web API نسخه آمده، این عدد برای Data Services چهار روز پیش بوده  
اگر بحث امکانات هست، لیست زیادی از امکانات رو من شمردم و می شه شمرد، از امکاناتی که تو Data Services هست، ولی تو Web API نیست.  
اگر من اندروید رو مثال زدم، برای سمت کلاینت بود، شما در اندروید با چی به Web API وصل می شید؟  
با jQuery Ajax؟  
یا می خواهید به App Server های .NET. ای برنامه های دیگر، بگویید با Http Client از سرویس های شما استفاده کنند؟  
با سپاس

نویسنده: محسن  
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۹:۱۶

هدف میکروسافت از یکپارچه کردن WEB API با ASP.NET و خصوصاً MVC ارائه یک سری Super ActionResult است بجای ActionResult های معمولی MVC3. برای نمونه:

[Using Kendo UI grid with Web API and OData](#)

نویسنده: میثم 99  
تاریخ: ۱۳۹۳/۰۱/۲۱ ۲۳:۰

سلام  
می خواهم بدانم برای امنیت web api در پروژه های web form چه کارهایی باید انجام دهیم بیشتر مطالب در مورد mvc هست مثلاً Anti-Forgery Tokens برای mvc به راحتی می توان استفاده کرد ولی برای web form چکار بهتر است انجام دهیم؟  
در اینجا ما مستقیماً با دستورات post put و delete کار داریم که اطلاعات بانک اطلاعاتی رو تغییر می دهند. حالا چطور می توان امنیت رو کاملاً تامین کرد؟



مثلا کاربران شناسایی شده اطلاعات را وارد کنند و اینکه شخصی نتواند با یک دستور ای جکس توسط مرورگر اطلاعات اشتباه در سایت ثبت کند؟ و یا هر مشکل امنیتی دیگری که پیش بیاید؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۱/۲۱ ۲۳:۲۱

روش‌های زیادی برای تامین امنیت در وب API و کار با «کاربران شناسایی شده» وجود دارند. [لیست رسمی](#) از این لیست رسمی، دو مورد معروف آن در سایت جاری بررسی شده:

[ASP.NET Identity](#)

[Forms authentication](#)

مباحث پایه‌ای این‌ها مشترک است بین MVC و وب فرم‌ها و سایر فناوری‌های مشابه.