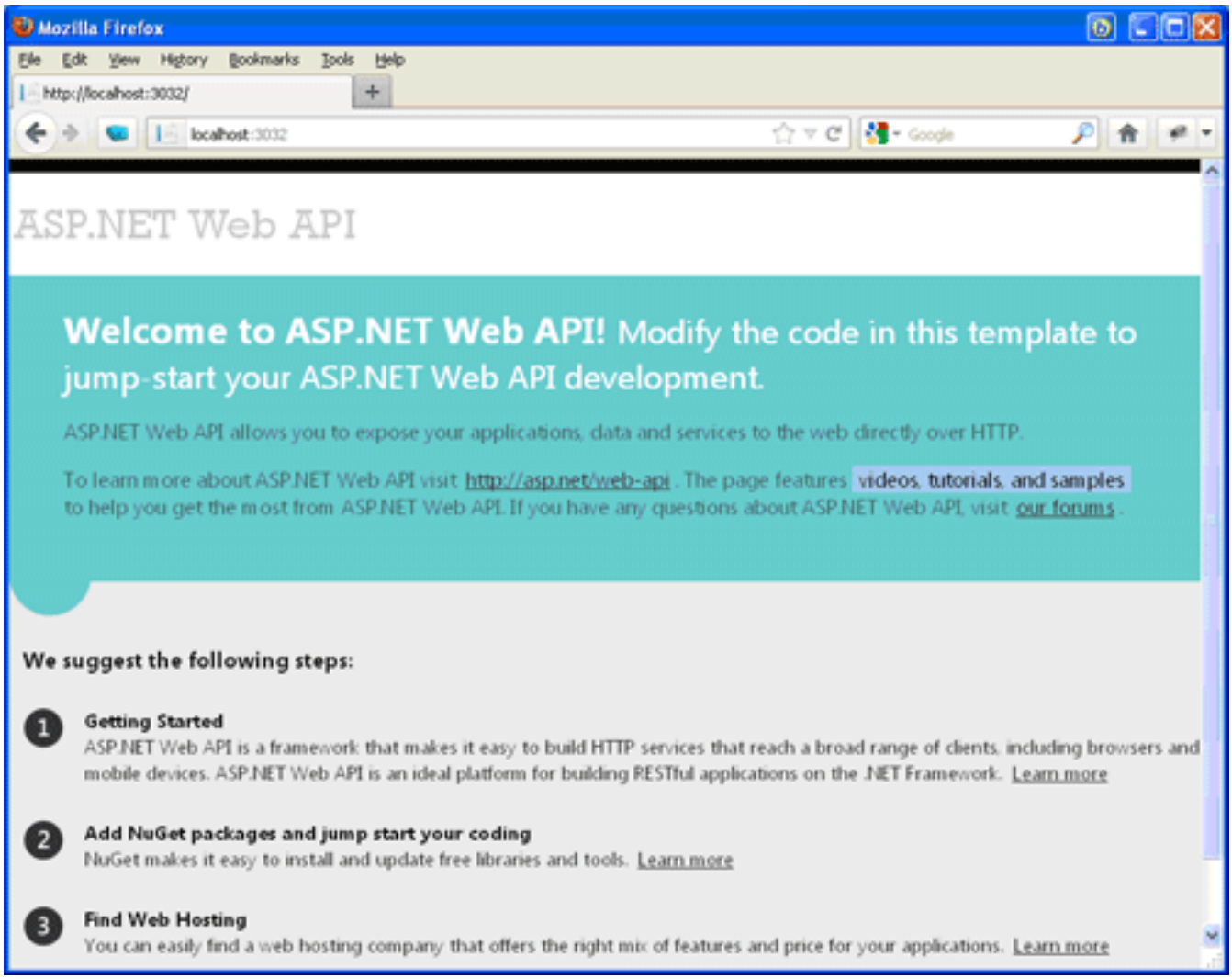
عنوان: **ASP.NET Web API - قسمت سوم**
نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۴/۱۶ ۱۲:۱
آدرس: www.dotnettips.info
برچسب‌ها: WCF, ASP.NET Web API, ASMX, Web Service

در قسمت اول به دلایل ایجاد Web API پرداخته شد و در قسمت دوم مثالی ساده از Web API را بررسی کردیم. در این قسمت، مثال قبل را تست کرده و نحوه‌ی تعامل jQuery با آن را بررسی می‌کنیم.

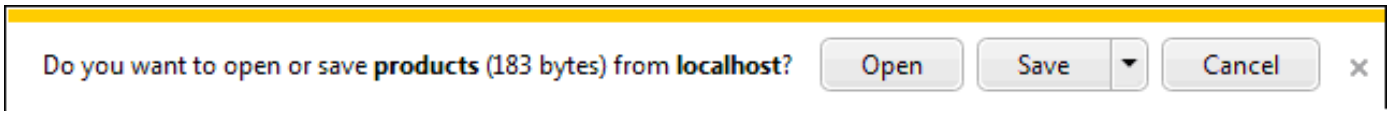
فراخوانی Web API از طریق مرورگر

با فشردن کلید F5، پروژه را اجرا کنید. شکل ذیل ظاهر می‌شود.



صفحه ای که ظاهر می‌شود، یک View است که توسط HomeController و متد Index آن برگشت داده شده است. برای فراخوانی متدهای موجود در کلاس Controller مثال قسمت قبل که مربوط به Web API است، باید به یکی از آدرس‌های اشاره شده در قسمت قبل برویم. به عنوان مثال، برای به دست آوردن لیست تمامی محصولات، به آدرس http://localhost:*xxxx*/api/products بروید. xxxx، شماره‌ی پورتی است که Web Server داخلی Visual Studio در هنگام اجرای پروژه به آن اختصاص می‌دهد. آن را نسبت به پروژه‌ی خود تغییر دهید.
نتیجه‌ی دریافتی بستگی به نوع مرورگری دارد که استفاده می‌کنید. Internet Explorer از شما در مورد باز کردن یا ذخیره‌ی

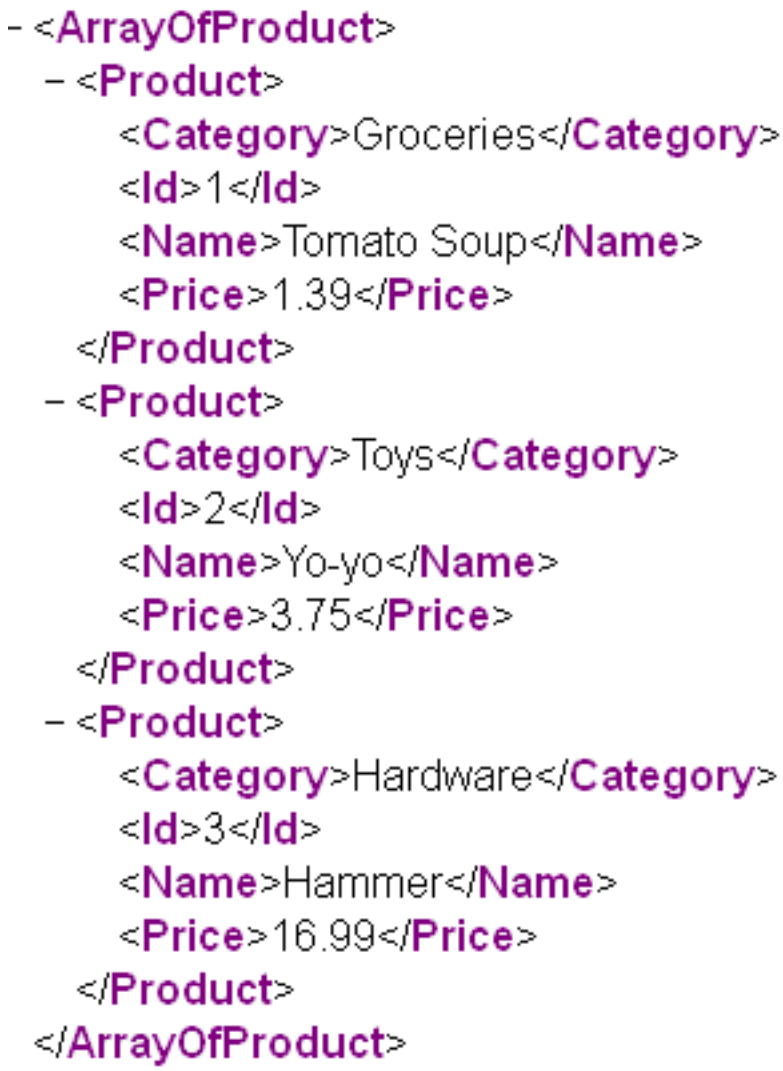
فایلی با نام products پرسش می‌کند (شکل ذیل).



محتوای فایل، بدنه‌ی پاسخ دریافتی است. اگر این فایل را باز کنید، خواهید دید که که محتوای آن، لیستی از محصولات با فرمت JSON مانند ذیل است.

```
[{"Id":1,"Name":"Tomato soup","Category":"Groceries","Price":1.39},{"Id":2,"Name":
"Yo-yo","Category":"Toys","Price":3.75},{"Id":3,"Name":"Hammer","Category":
"Hardware","Price":16.99}]
```

اما مرورگر Firefox، محصولات را در قالب XML نشان می‌دهد (شکل ذیل).



دلیل تفاوت در نتیجه‌ی دریافتی این است که مرورگر Internet Explorer و Firefox، هر یک مقدار متفاوتی را در هدر Accept درخواست، ارسال می‌کنند. بنابراین، Web API نیز مقدار متفاوتی را در پاسخ برگشت می‌دهد.

حال به آدرس‌های ذیل بروید:

http://localhost: *xxxx* /api/products/1
http://localhost: *xxxx* /api/products?category=hardware

اولین آدرس، باید محصولی با مشخصه‌ی 1 را برگشت دهد و دومین آدرس، لیستی از تمامی محصولاتی که در دسته‌ی hardware قرار دارند را برگشت می‌دهد (در مثال ما فقط یک آیتم این شرط را دارد).

نکته: در صورتی که در هنگام فراخوانی هر یک از متدهای Web API با خطای ذیل مواجه شدید، دستور [("AcceptVerbs("GET","POST"] را به ابتدای متدها اضافه کنید.

```
The requested resource does not support http method 'GET'
```

فراخوانی Web API با استفاده از کتابخانه‌ی jQuery

در قسمت قبل، متدهای Web API را مستقیماً از طریق وارد کردن آدرس آنها در نوار آدرس مرورگر فراخوانی کردیم. اما در اکثر اوقات، این متدها با روش‌های برنامه نویسی توسط یک Client فراخوانی می‌شوند. اجازه بدهید Clientیی ایجاد کنیم که با استفاده از jQuery، متدهای ما را فراخوانی می‌کند.
در Solution Explorer، از پوشه‌ی Views و سپس Home، فایل Index.cshtml را باز کنید.

تمامی محتویات این View را حذف و کدهای ذیل را در آن قرار دهید.

```
<!DOCTYPE html>
<html>
<head>
    <title>ASP.NET Web API</title>
    <script src="../../Scripts/jquery-1.7.2.min.js"
        type="text/javascript"></script>
</head>
<body>
    <div>
        <h1>All Products</h1>
        <ul id='products' />
    </div>
    <div>
        <label for="prodId">ID:</label>
        <input type="text" id="prodId" size="5"/>
        <input type="button" value="Search" onclick="find();" />
        <p id="product" />
    </div>
</body>
</html>
```

بازیابی لیستی از محصولات

برای بازیابی لیستی از محصولات، فقط کافی است تا یک درخواست از نوع GET به آدرس "api/products/" بفرستید. این کار با jQuery به صورت ذیل انجام می‌شود.

```
<script type="text/javascript">
    $(document).ready(function () {
        // Send an AJAX request
        $.getJSON("api/products/",
        function (data) {
            // On success, 'data' contains a list of products.
            $.each(data, function (key, val) {

                // Format the text to display.
                var str = val.Name + ': $' + val.Price;

                // Add a list item for the product.
                $('<li/>', { html: str })
                .appendTo($('#products'));
            });
        });
    });
</script>
```

متد getJSON، یک درخواست AJAX از نوع GET را ارسال می‌کند و پاسخ دریافتی آن نیز با فرمت JSON خواهد بود. دومین پارامتر متد getJSON، یک callback است که پس از دریافت موفقیت آمیز پاسخ اجرا می‌شود.

بازیابی یک محصول با استفاده از مشخصه‌ی آن

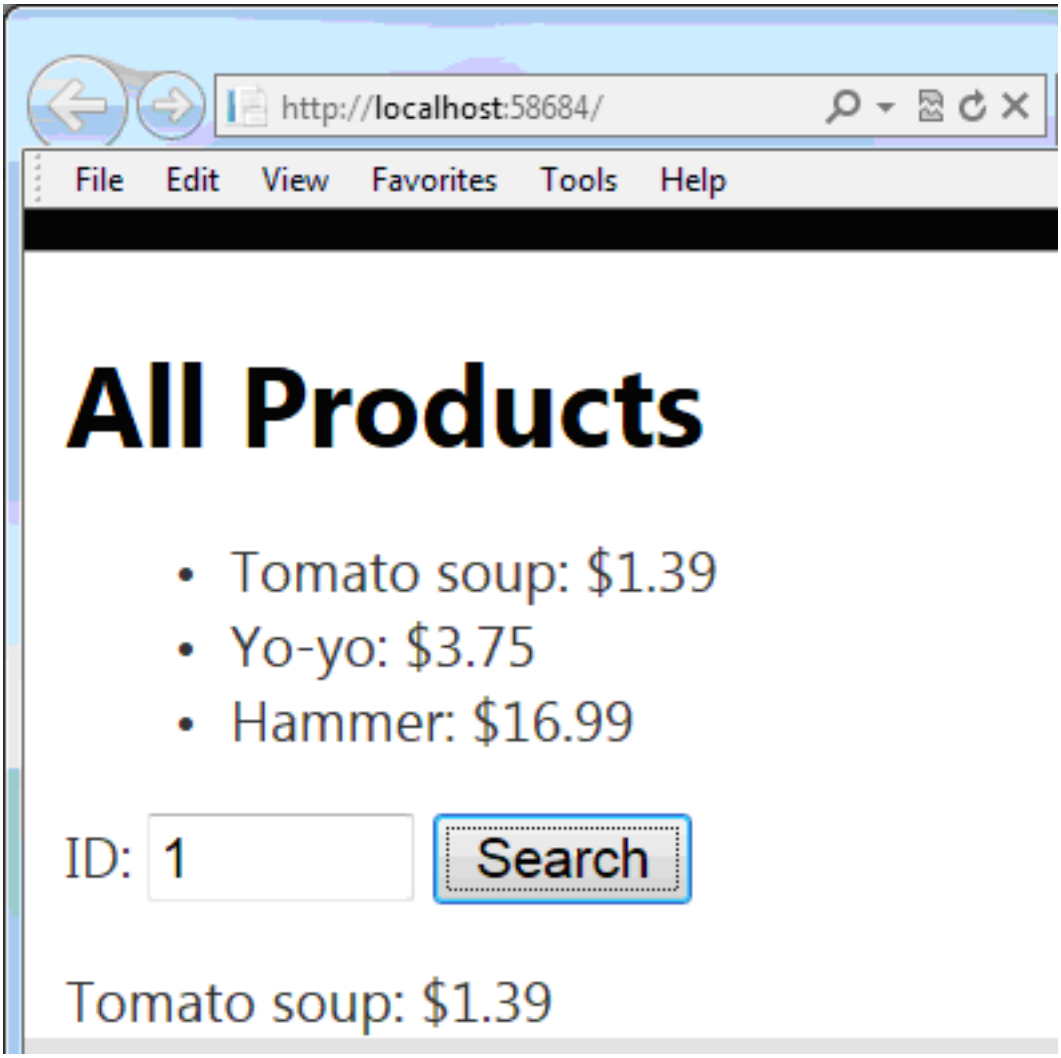
برای بازیابی یک محصول با استفاده از مشخصه‌ی آن، یک درخواست از نوع GET به آدرس "api/products/ *id*" / ارسال کنید. id، مشخصه‌ی محصول است. کد ذیل را در ادامه‌ی کد قبل و پیش از تگ <script/> قرار دهید.

```
function find() {
    var id = $('#prodId').val();
    $.getJSON("api/products/" + id,
        function (data) {
            var str = data.Name + ': $' + data.Price;
            $('#product').html(str);
        })
    .fail(
        function (jqXHR, textStatus, err) {
            $('#product').html('Error: ' + err);
        });
}
```

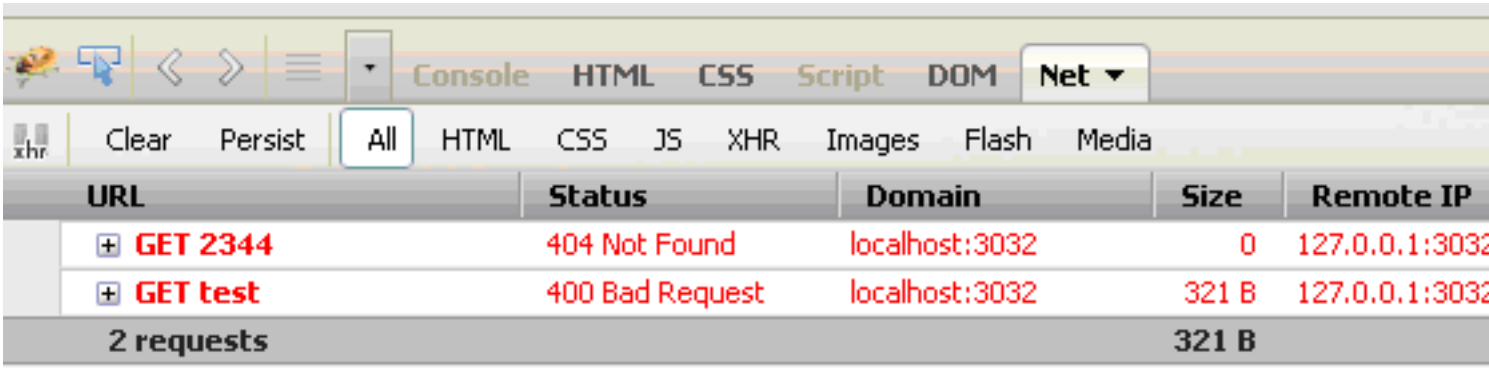
باز هم از متد getJSON استفاده کردیم، اما این بار مقدار id برای آدرس از یک Text Box خوانده و آدرس ایجاد می‌شود. پاسخ دریافتی، یک محصول در قالب JSON است.

اجرای پروژه

پروژه را با فشردن کلید F5 اجرا کنید. پس از نمایش فرم، تمامی محصولات بر روی صفحه نمایش داده می‌شوند. عدد 1 را وارد و بر روی دکمه‌ی Search کلیک کنید، محصولی که مشخصه‌ی آن 1 است نمایش داده می‌شود (شکل ذیل).



اگر مشخصه ای را وارد کنید که وجود ندارد، خطای 404 با مضمون "Error: Not Found" بر روی صفحه نمایش داده می‌شود و در صورتی که به جای عدد، عبارتی غیر عددی وارد کنید، خطای 400 با مضمون: "Error: Bad Request" نمایش داده می‌شود. **در Web API، تمامی پاسخ‌ها باید در قالب کدهای وضعیت HTTP باشند** (شکل ذیل). این یکی از اصول اساسی کار با وب سرویس‌ها است. وفادار ماندن به مفاهیم پایه‌ی وب، دید بهتری در مورد اتفاقاتی که می‌افتد به شما می‌دهد.



در قسمت بعد با مفهوم مسیریابی در ASP.NET Web API آشنا می‌شوید.

نظرات خوانندگان

نویسنده: Nima
تاریخ: ۱۳۹۱/۰۴/۱۶ ۲۰:۶

سلام آقای راد

ممنون از مطلب مفیدتون..سوالی ار حضورتون داشتم ما در وب سرویسهای asmx میتونستیم از sessionها استفاده کنیم تا مثلا اگر میخواستیم از طریق jquery بخواهیم اون وب سرویس رو صدا کنیم این کار فقط برای کاربرانی که در سیستم وارد شده اند امکان پذیر باشد .از لحاظ ملاحظات امنیتی و استفاده از session آیا در قسمتهای بعدی بحث میکنید؟

با تشکر

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۴/۱۷ ۷:۴۶

در Web API در حالت پیش فرض نمی‌تونید از Session استفاده کنید. اصولاً REST اصطلاحاً Stateless هست، اما اگر اصرار به استفاده از Session دارید، باید یک Route Handler سفارشی ایجاد و اینترفیس IRequiresSessionState رو پیاده سازی کنید. سپس پیاده سازی جدید رو به عنوان Route Handler برای route مختص Web API تعریف کنید.
در مورد تصدیق هویت، معمولاً به این شکل عمل میشه که یک فیلتر Authorize سفارشی ایجاد و نام کاربری و کلمه‌ی عبور از طریق یک Header سفارشی به Server ارسال میشه. Web API به خوبی با مفهوم فیلترها در ASP.NET MVC هماهنگ هست. سعی می‌کنم در مطلب جدایی به این موارد بپردازم.

نویسنده: Nima
تاریخ: ۱۳۹۱/۰۴/۱۷ ۹:۵۷

با تشکر از شما آقای راد اگر این زحمت رو بکشین ممنون میشم .دونستن مسائل امنیتی باعث استفاده بهتر از مواردی که شما فرمودین میشه. موفق باشید

نویسنده: مهران کلانتری
تاریخ: ۱۳۹۱/۰۴/۱۷ ۱۸:۵۴

سلام آقای راد خیلی خوب و سلیس توضیح میدید

در رابطه با مسائل امنیتی در این روش خیلی خوب می‌شه اگر توضیحی ارائه بدید.

متشکرم

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۱۸ ۲۱:۲۵

در Rest قابلیتی بنام Syndication Feed Fromatter وجود دارد در Web API چطور؟

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۴/۱۹ ۱۰:۳۲

در Web API هم این قابلیت وجود داره.