

در مطالب قبلی با پروتکل OData و WCF Data Service و قراردادهای کوثری نویسی آن آشنا شدید. حال می‌خواهیم با استفاده از JQuery به داده‌های وب سرویس WCF Data Service دسترسی یابیم. اما پیش نیازهای لازم است

### پیش نیاز اول : دسترسی به خروجی Json وب سرویس WCF Data

خروجی پیش فرس وب سرویس WCF Data Services ساختار Xml دارد پس می‌بایست وب سرویس را متوجه سازیم که ما با خروجی Json نیاز داریم. از نسخه 5 به بعد اگر MaxProtocolVersion را بر روی V3 قرار دهیم دیگر با Accept Header برابر application/json کار نخواهد کرد و می‌بایست از application/json;odata=verbose استفاده نمود یا نسخه پروتکل را بر روی V2 یا پایین‌تر تنظیم کنید. علاوه بر آن کتابخانه‌های و قطعه کدهای تهیه شده است که با پارامتر \$format این کار را برای ما انجام می‌دهد در زیر آدرس دو نمونه آورده شده است.

[DataServicesJSONP](#)

[WCF Data Services Toolkit](#)

قطعه کد اول یک Attribute است که با اضافه کردن آن به بالای کلاس WebService و استفاده از پارامتر \$format=json در آدرس وب سرویس این کار را برای ما انجام می‌دهد.

```
[JSONPSupportBehavior]
public class Northwind : DataService<NorthwindEntities>
```

و نمونه آدرس

```
http://localhost:8358/Northwind.svc/Products?$format=json
```

دومی کتابخانه ای است که مانند روش اول عمل می‌کند اما به جای ارث برای از کلاس DataService می‌بایست از کلاس ODataService استفاده نماییم.

نکته: در صورتی که بخواهیم از نسخه V3 استفاده نماییم Accept Header را باید به application/json;odata=verbose تغییر دهیم

```
public class Northwind : ODataService<NorthwindEntities>
```

### استفاده از WCF Data Services به کمک JQuery

تابع getJSON مخصوص درخواست‌های است خروجی بصورت json برگردانده می‌شود اما با نسخه V3 سازگار نمی‌باشد و از روش پارامتر \$format می‌توان استفاده نمود

```
$.getJSON("Northwind.svc/Products?$format=json", function (data) {
    $.each(data.d, function (i, item) {
        $("

```

همچنین از تابع Ajax که امکانات بیشتری را در اختیارمان قرار می دهد به راحتی می توان استفاده نمود به مثال زیر دقت کنید:

```
$.ajax('Northwind.svc/Products', {
    dataType: "json",
    beforeSend: function (xhr) {
        xhr.setRequestHeader("Accept", "application/json;odata=verbose");
        xhr.setRequestHeader("MaxDataServiceVersion", "3.0");
    },
    success: function (data) {
        $.each(data.d, function (i, item) {
            $("<p/>").html(item.Product_Name + " " +
item.Unit_Price).appendTo("#products");
        });
    }
});
```

با استفاده از beforeSend مقدار Accept Header و MaxDataServiceVersion را تعیین نموده ایم.  
بنابراین به کمک قراردادهای کوئری نویسی که در مطالب قبلی گفته شد می توان با استفاده از Url تابع Ajax به داده مورد نظر خود رسید.

### نظرات خوانندگان

نویسنده: مرادی

تاریخ: ۱۷:۴۲ ۱۳۹۱/۱۱/۰۵

با سلام، بهتر نیست از jay data یا از breeze.js استفاده کنید ؟

نویسنده: مجتبی کاویانی

تاریخ: ۱۸:۳۴ ۱۳۹۱/۱۱/۰۵

در مطلب بعدی به این دو اشاره خواهم کرد قطعا امکانات بیشتری در اختیارمان قرار می‌دهد

نویسنده: abdali

تاریخ: ۱۵:۱۹ ۱۳۹۲/۰۴/۱۱

لطف میکنید کد رو قرار بدین ، چند بار امتحان کردم با خطا مواجه شدم . مرسی