

در حین کار با [WCF RIA Services](#) اگر تمام موجودیت‌های تعریف شده انتخاب شوند یک فایل طویل DomainService حاصل خواهد شد. کار کردن با این فایل نه زیبا است و نه ساده. بعد از یک مدت شاید بگوئیم، خوب! من به ازای هر جدول یک DomainService جدا تولید می‌کنم با نامی مختص به آن و این اطلاعات را در فایلی جداگانه نیز ذخیره خواهم کرد. پس از انجام اینکار با خطای زیر مواجه خواهیم شد:

The entity type ... is exposed by multiple DomainService types. Entity types

cannot be shared

across DomainServices

به صورت خلاصه: مهم نیست سیستم شما از چند جدول تشکیل شده است؛ مهم این است که تنها یک فایل DomainService را باید تولید کنید و البته این یک محدودیت نیست؛ یک هدف محسوب می‌شود؛ از این دیدگاه که موجودیت‌های مرتبط باید در یک Domain قرار گیرند و تنها در یک دومین هستند که روابط بین آن‌ها معنا پیدا می‌کند.

- روش توصیه شده برای مدیریت این کلاس DomainService طویل، استفاده از واژه‌ی کلیدی partial است (برای مثال public partial class MyDomainService). به این صورت هر موجودیت را می‌توان در یک فایل جداگانه قرار داد و به این ترتیب مدیریت ساده‌تری را بر روی اطلاعات تولید شده داشت و همچنین تمامی این فایل‌ها در نهایت یک کلاس واحد را تشکیل می‌دهند و اصل وجود یک DomainService واحد در برنامه زیر سؤال نخواهد رفت.

- کدهایی را که خودتان نیز به این مجموعه اضافه خواهید کرد، در لابلای کدهای تولید شده قرار ندهید. در صورت تغییری در جداول نیاز است تا این فایل‌ها مجدداً تولید شوند و اینجا است که تمام تغییرات خود را از دست خواهید داد. برای این منظور باز هم یک کلاس partial دیگر را تعریف کنید تا کدهای سفارشی خود را بتوان به صورت مجزایی از کدهای تولید شده به صورت خودکار، در آن قرار داد (برای مثال به نام MyServiceNameDomainService.extensions.cs).

- جایی که قرار است یک سری از فایل‌ها مجدداً تولید شوند استفاده از برنامه‌های [سورس کنترل](#) را فراموش نکنید؛ تا هر زمانی بتوان کدهای جدید را با کدهای نگارش‌های قبل به سادگی مقایسه کرد؛ یا حتی به نگارش‌های قبلی بازگشت نمود.

- این نکته را به خاطر داشته باشید که اگر اطلاعاتی به صورت خودکار از پیش تعریف شده و موجود است (کلاس دومین یا متادیتای آن)، این اطلاعات حین استفاده از امکانات تولید کد خودکار، مجدداً تولید نخواهند شد. به همین جهت باید ابتدا آن‌ها را به صورت comment در آورد (یا از پروژۀ خارج نمود؛ استفاده از امکانات include و exclude پوشه‌ها یا فایل‌ها در ویژوال استودیو) تا همواره کدهای آخرین اطلاعات موجود تولید گردند.

پیشتر در مورد [ELMAH](#) مطلبی را [منتشر کرده بودم](#) و اگر برنامه نویسی ASP.NET هستید و با ELMAH آشنایی ندارید، جدا نیمی از عمر کاری شما بر فنا است!

هاست پیش فرض یک WCF RIA Service هم یک برنامه‌ی ASP.NET است. بنابراین کلیه‌ی خطاهای رخ داده در سمت سرور را باید بتوان به نحوی لاگ کرد تا بعداً با مطالعه‌ی آن‌ها اطلاعات ارزشمندی را از نقایص برنامه در عمل و پیش از گوشزد شدن آن‌ها توسط کاربران، دریافت، بررسی و رفع کرد.

کلیه خطاها را لاگ می‌کنم تا:

- بدانم معنای جمله‌ی "برنامه کار نمی‌کند" چی هست.
- بدون روبرو شدن با کاربران یا حتی سؤال و جوابی از آن‌ها بدانم دقیقاً مشکل از کجا ناشی شده.
- بدانم رفتارهای عمومی کاربران که منجر به بروز خطا می‌شوند کدام‌ها هستند.
- بدانم در کدامیک از قسمت‌های برنامه تعیین اعتبار ورودی کاربران یا انجام نشده یا ضعیف و ناکافی است.
- بدانم زمانیکه دوستی (!) قصد پایین آوردن برنامه را با تزریق SQL داشته، دقیقاً چه چیزی را وارد کرده، در کجا و چه زمانی؟
- بتوانم Remote worker خوبی باشم.

ELMAH هم برای لاگ کردن خطاهای مدیریت نشده‌ی یک برنامه‌ی ASP.NET ایجاد شده است. بنابراین باید بتوان این دو (WCF RIA Services و ELMAH) را به نحوی با هم سازگار کرد. برای اینکار نیاز است تا یک مدیریت کننده‌ی خطای سفارشی را با پیاده سازی اینترفیس `ExceptionHandler` تهیه کنیم (تا خطاهای مدیریت نشده‌ی حاصل را به سمت ELMAH هدایت کند) و سپس آن‌را به کمک یک ویژگی یا `Attribute` به `DomainService` خود جهت لاگ کردن خطاها اعمال نمائیم. روش تعریف این `Attribute` را در کدهای بعد ملاحظه خواهید نمود (در اینجا نیاز است تا دو ارجاع را به اسمبلی‌های `Elmah.dll` که دریافت کرده‌اید و اسمبلی استاندارد `System.ServiceModel` نیز به پروژه اضافه نمائید):

```
//add a reference to "Elmah.dll"
using System;
using System.ServiceModel.Channels;
using System.ServiceModel.Dispatcher;
using System.Web;

namespace ElmahWcf
{
    public class HttpExceptionHandler : IExceptionHandler
    {
        #region IExceptionHandler Members
        public bool HandleError(Exception error)
        {
            return false;
        }

        public void ProvideFault(Exception error, MessageVersion version, ref Message fault)
        {
            if (error == null)
                return;

            if (HttpContext.Current == null) //In case we run outside of IIS
                return;

            Elmah.ErrorSignal.FromCurrentContext().Raise(error);
        }
        #endregion
    }
}
```

```
//add a ref to "System.ServiceModel" assembly
using System;
```

```

using System.Collections.ObjectModel;
using System.ServiceModel;
using System.ServiceModel.Channels;
using System.ServiceModel.Description;
using System.ServiceModel.Dispatcher;

namespace ElmahWcf
{
    public class ServiceErrorBehaviorAttribute : Attribute, IServiceBehavior
    {
        Type errorHandlerType;
        public ServiceErrorBehaviorAttribute(Type errorHandlerType)
        {
            this.errorHandlerType = errorHandlerType;
        }

        #region IServiceBehavior Members

        public void AddBindingParameters(
            ServiceDescription serviceDescription,
            ServiceHostBase serviceHostBase, Collection<ServiceEndpoint> endpoints,
            BindingParameterCollection bindingParameters)
        { }

        public void ApplyDispatchBehavior(
            ServiceDescription serviceDescription,
            ServiceHostBase serviceHostBase)
        {
            IErrorHandler errorHandler;
            errorHandler = (IErrorHandler)Activator.CreateInstance(errorHandlerType);
            foreach (ChannelDispatcherBase cdb in serviceHostBase.ChannelDispatchers)
            {
                ChannelDispatcher cd = cdb as ChannelDispatcher;
                cd.ErrorHandlers.Add(errorHandler);
            }
        }

        public void Validate(
            ServiceDescription serviceDescription,
            ServiceHostBase serviceHostBase)
        { }
        #endregion
    }
}

```

اکنون پس از تعریف ویژگی `ServiceErrorBehavior`، نوبت به اعمال آن می‌رسد. به فایل `DomainService` خود مراجعه کرده و یک سطر زیر را به آن اضافه نمائید:

```

[ServiceErrorBehavior(typeof(HttpErrorHandler))] //Integrating with ELMAH
[EnableClientAccess()]
public partial class MyDomainService : LinqToEntitiesDomainService<myEntities>

```

در ادامه نحوه‌ی افزودن تعاریف متناظر با ELMAH به `Web.Config` برنامه ذکر شده است. این تعاریف برای IIS6 و 7 به بعد هم تکمیل گردیده است. خطاها هم به صورت فایل‌های XML در پوشه‌ای به نام `Errors` که به ریشه‌ی سایت اضافه خواهید نمود (یا هر پوشه‌ی دلخواه دیگری)، لاگ می‌شوند. به نظر من این روش، از ذخیره سازی اطلاعات لاگ‌ها در دیتابیس بهتر است. چون اساساً زمانیکه خطایی رخ می‌دهد شاید مشکل اصلی همان ارتباط با دیتابیس باشد. قسمت ارسال خطاها به صورت ایمیل نیز `comment` شده است که در صورت نیاز می‌توان آن را فعال نمود:

```

<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <sectionGroup name="elmah">
      <section name="security" requirePermission="false" type="Elmah.SecuritySectionHandler, Elmah"/>
      <section name="errorLog" requirePermission="false" type="Elmah.ErrorLogSectionHandler, Elmah" />
      <section name="errorMail" requirePermission="false" type="Elmah.ErrorMailSectionHandler, Elmah" />
      <section name="errorFilter" requirePermission="false" type="Elmah.ErrorFilterSectionHandler, Elmah"/>
      <section name="errorTweet" requirePermission="false" type="Elmah.ErrorTweetSectionHandler,

```

```

Elmah"/>
</sectionGroup>
</configSections>

<elmah>
  <security allowRemoteAccess="1" />
  <errorLog type="Elmah.XmlFileErrorLog, Elmah" logPath="~/Errors" />
  <!-- <errorMail
    from="errors@site.net"
    to="nasiri@site.net"
    subject="prj-error"
    async="true"
    smtpPort="25"
    smtpServer="mail.site.net"
    noYsod="true" /> -->
</elmah>

<system.webServer>
  <modules runAllManagedModulesForAllRequests="true">
    <add name="ErrorLog" type="Elmah.ErrorLogModule, Elmah"/>
    <add name="DomainServiceModule"
      precondition="managedHandler"
      type="System.ServiceModel.DomainServices.Hosting.DomainServiceHttpModule,
System.ServiceModel.DomainServices.Hosting, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" />
  </modules>
  <validation validateIntegratedModeConfiguration="false" />
  <handlers>
    <add name="Elmah" verb="POST,GET,HEAD" path="myelmah.axd" type="Elmah.ErrorLogPageFactory, Elmah"
  />
  </handlers>
</system.webServer>
<system.web>
  <globalization
    requestEncoding="utf-8"
    responseEncoding="utf-8"
  />
  <authentication mode="Forms">
    <!--one month ticket-->
    <forms name=".403AuthV"
      cookieless="UseCookies"
      slidingExpiration="true"
      protection="All"
      path="/"
      timeout="43200" />
  </authentication>
  <httpHandlers>
    <add verb="POST,GET,HEAD" path="myelmah.axd" type="Elmah.ErrorLogPageFactory, Elmah" />
  </httpHandlers>
  <httpModules>
    <add name="ErrorLog" type="Elmah.ErrorLogModule, Elmah"/>
    <add name="DomainServiceModule"
      type="System.ServiceModel.DomainServices.Hosting.DomainServiceHttpModule,
System.ServiceModel.DomainServices.Hosting, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" />
  </httpModules>
  <compilation debug="true" targetFramework="4.0">
    <assemblies>
      <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" />
    </assemblies>
  </compilation>
</system.web>
<connectionStrings>
</connectionStrings>
<system.serviceModel>
  <serviceHostingEnvironment
    aspNetCompatibilityEnabled="true"
    multipleSiteBindingsEnabled="true" />
</system.serviceModel>
</configuration>

```

اکنون برای مثال به یکی از متدهای DomainService خود سطر زیر را اضافه کرده و برنامه را آزمایش کنید:

```
throw new Exception("This is an ELMAH test");
```

سپس به آدرس <http://localhost/myelmah.axd> مراجعه نموده و اطلاعات لاگ شده حاصل را بررسی کنید:

Error Log for / on VAHIDPCXP

[RSS FEED](#)
[RSS DIGEST](#)
[DOWNLOAD LOG](#)
[HELP](#)
[ABOUT](#)

Errors 1 to 2 of total 2 (page 1 of 1). Start with [10](#), [15](#), [20](#), [25](#), [30](#), [50](#) or [100](#) errors per page.

Host	Code	Type	Error	User	Date	Time
VAHIDPCXP	404	Http	Path '/elmah.axd' was not found. Details...	کاربر *	2010/08/11	ب.ظ 02:56
VAHIDPCXP	0	0, Culture=neutral, PublicKeyToken=31bf3856ad364e35]]	This is an ELMAH test Details...		2010/08/11	ب.ظ 02:54

Powered by [ELMAH](#), version 1.1.11517.2009. Copyright (c) 2004-9, Atif Aziz. All rights reserved. Licensed under [Apache License, Version 2.0](#). Server date is Wednesday, 11 August 2010. Server time is 15:00:54. All dates and times displayed are in the Iran Daylight Time zone. This log is provided by the XML File-Based Error Log.

این روش با WCF Services های متداول هم کار می کند. فقط در این سرویس ها باید `aspNetCompatibilityEnabled` مطابق تگ های ذکر شده ی `system.serviceModel` فوق در `web.config` لحاظ شوند (این مورد به صورت پیش فرض در WCF RIA Services وجود دارد). همچنین ویژگی زیر نیز باید به سرویس شما اضافه گردد:

```
[AspNetCompatibilityRequirements(RequirementsMode = AspNetCompatibilityRequirementsMode.Allowed)]
```

منابع مورد استفاده:

[Integrating ELMAH for a WCF Service](#)

[Making WCF and ELMAH play nice together](#)

[Getting ELMAH to work with WCF services](#)

پ.ن.

اگر به خطاهای ASP.NET دقت کرده باشید که به [yellow screen of death](#) هم مشهور هستند (در مقابل صفحات آبی ویندوز!)، ابتدای آن خیلی بزرگ نوشته شده `Server Error` و سپس ادامه ی خطا. همین مورد دقیقاً یادم هست که هر بار سبب بازخواست مدیران شبکه بجای برنامه نویسی ها می شد! (احتمالاً این هم یک نوع بدجنسی تیم ASP.NET برای گرفتن حال ادمین های شبکه است! و گر نه مثلاً می توانستند همان ابتدا بنویسند `program/application error` بجای `server error`)

نظرات خوانندگان

نویسنده: peyman naji
تاریخ: ۱۳۸۹/۰۵/۲۱ ۱۵:۲۲:۰۱

عالی بود . تشکر

نویسنده: vahid
تاریخ: ۱۳۸۹/۰۶/۱۶ ۱۵:۴۹:۴۲

سلام

من از روش تشخیص هویت ویندوز استفاده نمی کنم چطور می تونم جلوی دسترسی افراد به صفحه myelmah.axd رو بگیرم

ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۶/۱۶ ۱۸:۵۵:۴۱

الف - این اسم هر چیزی می تونه باشه که در قسمت path های ذکر شده فوق قابل تغییر است (بنابراین از اسم های پیش فرض استفاده نکنید).

ب - بهترین مرجع برای ELMAH ، فایل web.config پوشه samples آن است. در این مورد تنظیمات لازم وجود دارد (در مورد بستن دسترسی ریموت یا استفاده از authorization).

خلاصه این دو مورد رو یکی از اعضای اصلی تیم ASP.NET در عمل نشان داده:

<http://haacked.com/archive/2007/07/24/securely-implement-elmah-for-plug-and-play-error-logging.aspx>

+

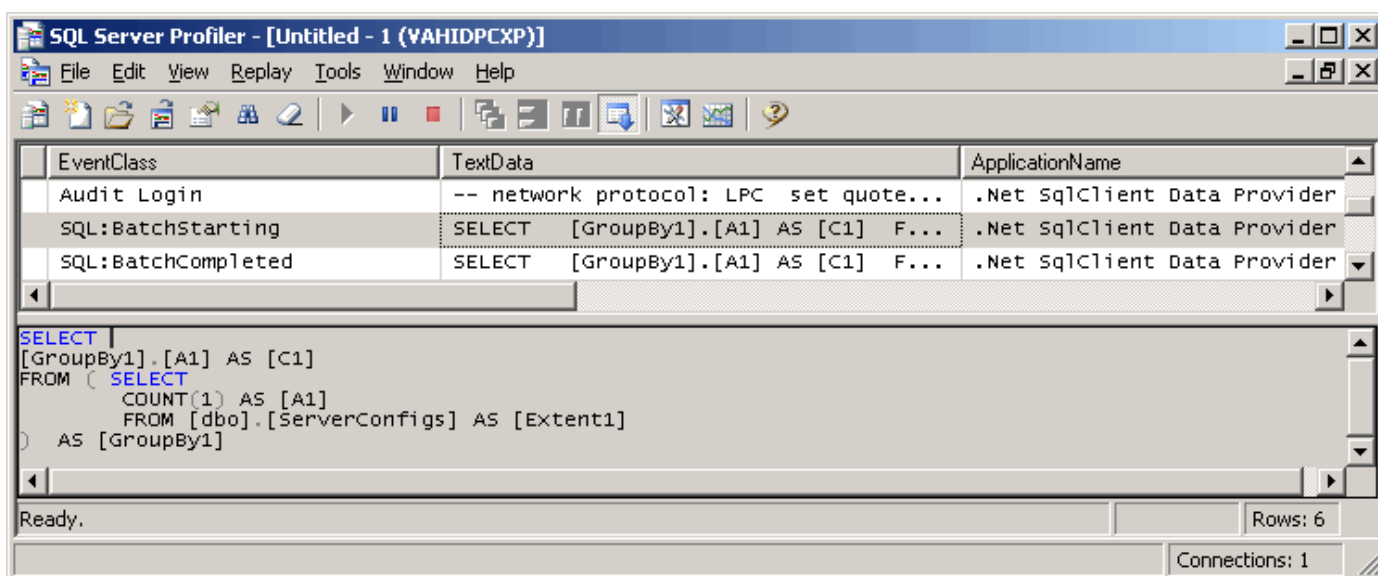
<http://code.google.com/p/elmah/wiki/SecuringErrorLogPages>

این روزها با وجود ORMs ، کوئری SQL نوشتن شبیه به دورانی شده که با وجود زبان‌های سطح بالا، عده‌ای علاقمند هستند با استفاده از زبان اسمبلی برنامه نویسی کنند! WCF RIA Services به صورت پیش فرض از entity framework استفاده می‌کند (هر چند می‌توان از سایر ORMs هم استفاده کرد)، بنابراین عنوان صحیح‌تر بحث این خواهد بود: چگونه خروجی SQL تولید شده توسط Entity framework را بررسی کنیم؟

الف) استفاده از SQL Server profiler

اولین برنامه‌ای که از سال‌ها قبل، حتی پیش از ظهور ORMs وجود داشته، برنامه‌ی SQL server profiler است، که عموماً در مسیر ذیل قابل دستیابی است:

Start Menu->Programs->Microsoft SQL Server 2008->Performance Tools->SQL Server profiler

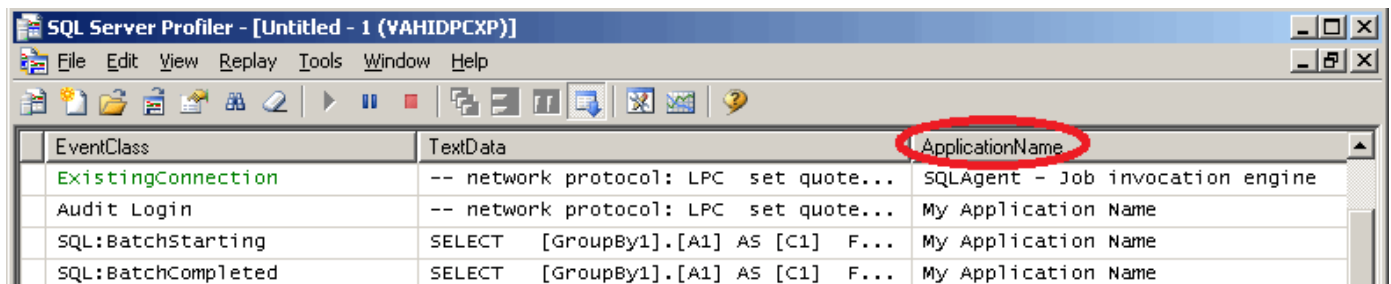


نکته مهم:

حین کار با SQL Server profiler ، ممکن است انبوهی از کوئری‌های دیگر مثلاً مرتبط با SQL Server agent یا reporting services و غیره نیز لاگ شوند. اما الان ما تنها به کوئری‌های برنامه‌ی خود نیاز داریم. برای این منظور به کانکشن استرینگ خود، گزینه‌ی Application Name=My Application Name را نیز اضافه کنید:

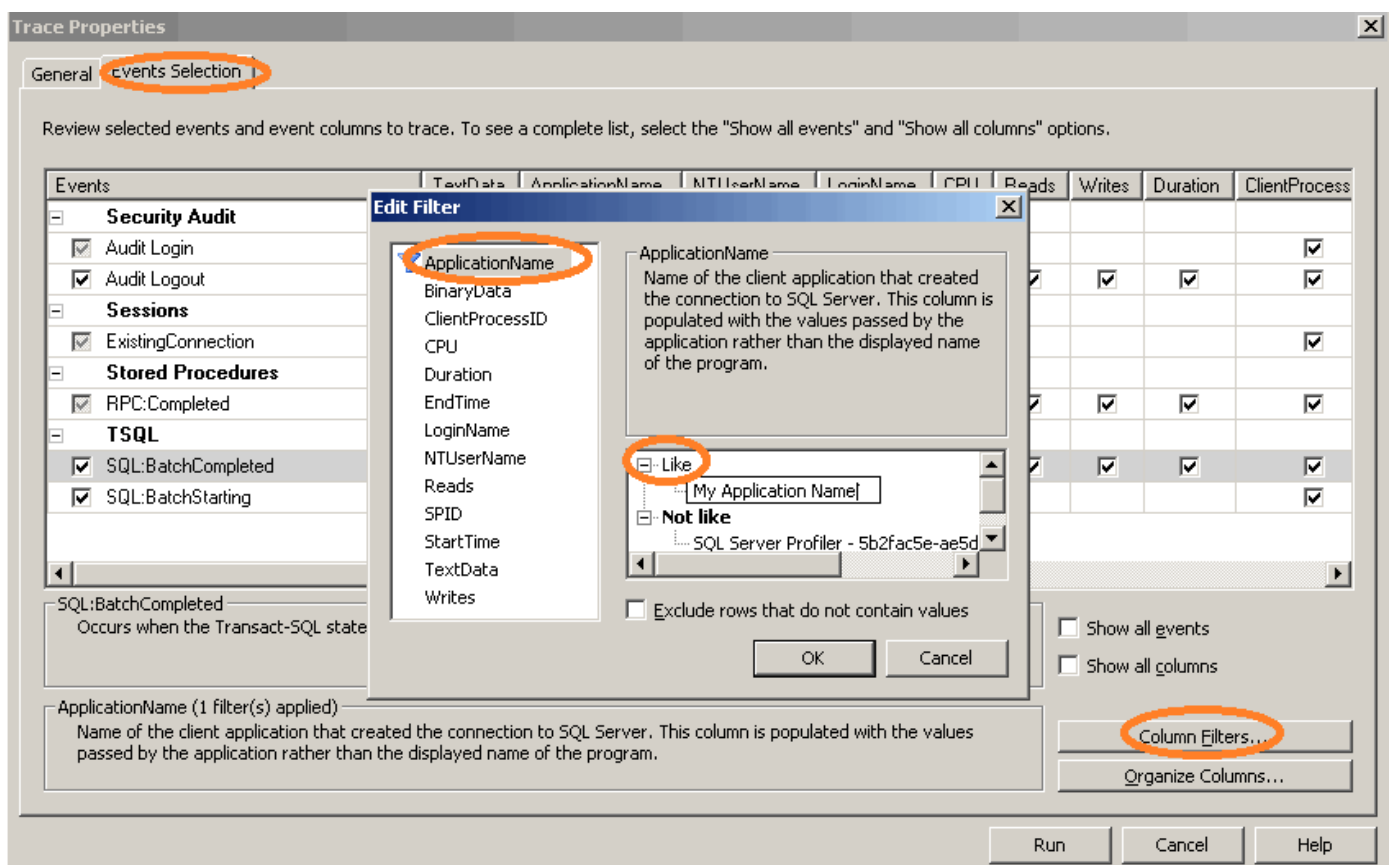
```
<connectionStrings>
  <add name="dmEntities"
connectionString="metadata=res://*/Models.dmDataModel.csdl|res://*/Models.dmDataModel.ssdl|res://*/Models.dmDataModel.msl;provider=System.Data.SqlClient;provider connection string="Data Source=(local);Initial Catalog=dm;Integrated Security=True;Application Name=My Application Name;MultipleActiveResultSets=True" providerName="System.Data.EntityClient" />
</connectionStrings>
```

اکنون اگر برنامه را با پروفایلر مورد بررسی قرار دهید خروجی به صورت زیر خواهد بود:



EventClass	TextData	ApplicationName
ExistingConnection	-- network protocol: LPC set quote...	SQLAgent - Job invocation engine
Audit Login	-- network protocol: LPC set quote...	My Application Name
SQL:BatchStarting	SELECT [GroupBy1].[A1] AS [C1] F...	My Application Name
SQL:BatchCompleted	SELECT [GroupBy1].[A1] AS [C1] F...	My Application Name

برای فیلتر کردن Application Name مورد نظر، در ابتدای کار که یک سشن جدید را آغاز می‌کنید به برگه‌ی events selection مراجعه کرده و بر روی دکمه‌ی column filter کلیک کنید. گزینه‌ی application name را در صفحه‌ی باز شده انتخاب نموده و در قسمت Like آن مطابق تصویر زیر، نام برنامه‌ی خود را وارد نمایید:



ب) استفاده از IntelliTrace در VS.NET 2010

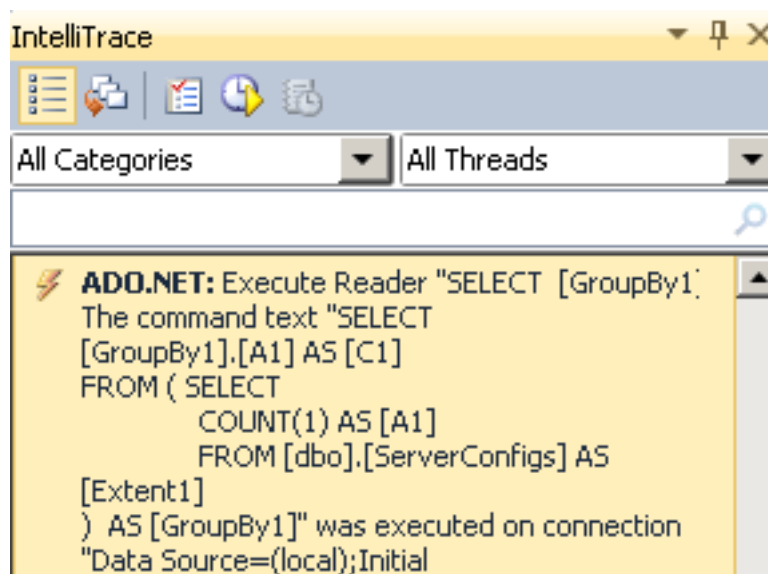
برنامه را در حالت دیباگ در VS.NET 2010 اجرا کنید. در هر لحظه‌ای می‌توان روی گزینه‌ی Break all کلیک کرد و خروجی SQL

تولید شده را نیز علاوه بر اطلاعات دیگر مشاهده نمود:



To view IntelliTrace data, you must break execution of your application.

 **Break All**



ج) استفاده از برنامه‌ی حرفه‌ای [entity framework profiler](#) این برنامه از هر دو مورد قبل کاملتر بوده و اساساً برای لاگ کردن کوئری‌ها، مدت زمان اجرا، گزارشگیری از وضعیت برنامه، کدامیک از کوئری‌ها سنگین‌تر هستند، حتی از طریق کدام متد فراخوانی شده‌اند، ارائه‌ی گزارشات و راهنمایی‌هایی در مورد چگونگی بهبود کارایی برنامه‌ی تهیه شده و امثال آن کاربرد دارد.

استفاده از آن هم بسیار ساده است. ابتدا ارجاعی را به اسمبلی `HibernatingRhinos.Profiler.Appender.v4.0` به پروژه‌ی ASP.NET خود اضافه کنید (همان پروژه‌ی هوست مربوط به WCF RIA Service ما). سپس به فایل `Global.asax.cs` برنامه مراجعه کرده و یک سطر ذیل را اضافه کنید:

```
protected void Application_Start(object sender, EventArgs e)
{
    HibernatingRhinos.Profiler.Appender.EntityFramework.EntityFrameworkProfiler.Initialize();
}
```

از این پس تنها کافی است برنامه‌ی پروفایلر در حال اجرا بوده و برنامه شما نیز اجرا شود. کلیه‌ی تبادلات با دیتابیس لاگ خواهند شد.

نظرات خوانندگان

نویسنده: محمود رمضانی
تاریخ: ۱۳۸۹/۰۵/۲۹ ۱۳:۳۹:۳۸

سلام. یک سوال در زمینه EF ؟
اینکه من همه کد ها رو به صورت Stored Procedure بنویسم و بعد فقط اون ها رو در EF استفاده کنم اشتباه؟
از این بابت میگم اشتباه چون فکر می کنم با این کار از قدرت EF استفاده نمی کنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۵/۲۹ ۱۴:۳۴:۴۴

سلام،
من یک زمانی از طرفداران نوشتن stored procedure بودم اما الان به دلایل ذیل نیستم:
- امنیت: تمام کوئری‌های تولیدی entity framework از نوع پارامتری هستند. این مورد یعنی عقیم سازی حملات تزریق اس کیوال . (بنابراین اگر کسی عنوان کند که با SP ما امنیت بیشتری داریم، باید عنوان کرد که اینجا هم به همین صورت است)
- سرعت: در نگارش‌های جدید اس کیوال سرور، با کوئری‌های پارامتری دقیقاً مانند SP رفتار می‌شود. همان کش شدن execution plan و غیره. بنابراین اینجا هم از همان مزایای SP برخوردار هستیم و سرعت سیستم مطلوب است.
- مشکلات نگهداری SP :
شما می‌تونید ساختار جداول رو تغییر بدید بدون اینکه اس کیوال سرور به شما پیغام خطایی در مورد غیر معتبر شدن یک SP بدهد. شما می‌تونید حتی یک SP غیر معتبر را تا زمانیکه syntax مربوط به آن صحیح است تولید کنید. هر دو مورد در زمان اجرا، سبب از کار افتادن برنامه می‌شوند. اما با EF این مشکلات را ندارید. ساختار را که عوض کنید برنامه دیگر کامپایل نمی‌شود. این مورد در برنامه‌های بزرگ خیلی خیلی خوب است!
مورد دیگر: یک برنامه بزرگ با چند صد SP رو در نظر بگیرید. جدا نگهداری این‌ها پیدا کردن کدها در یک برنامه‌ی بزرگ عذاب است. طبقه بندی آن‌ها یک طرف، اعمال تغییرات از طرف دیگر.
- مورد دیگر که من دیدم در یک سری از سایت‌ها در مورد آن بحث می‌کنند این است که نباید business logic برنامه را داخل دیتابیس طراحی کرد. این مورد باید با کد نویسی داخل برنامه باشد. در اینجا هم باز EF یا موارد مشابه بهتر هستند.
- مورد دیگری که در SP ها مشکل ساز می‌شود به اشتراک گذاری آن در بین برنامه‌های مختلف است. هم خوب است. بالاخره کد نویسی کمتر می‌شود. هم بد است، از این لحاظ که شاید این SP نیاز به تغییر داشت و اینجا است که برنامه‌های دیگر مشکل پیدا می‌کنند.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۵/۲۹ ۱۴:۳۸:۴۷

- یک مورد دیگر را هم اضافه کنم. با SP نویسی شما برنامه خودتون رو به SQL Server گره می‌زنید. اما با استفاده از EF خالص به سادگی می‌تونید به بانک‌های اطلاعاتی دیگری که پروایدر EF دارند سوئیچ کنید.

نویسنده: محمود رمضانی
تاریخ: ۱۳۸۹/۰۵/۳۰ ۲۰:۳۴:۵۸

ممنون بخاطر جواب.
یک سوال دیگه اینکه EF موجب افزایش Round Trip نمی شه؟ منظورم در مقایسه با SP هست.
البته دلایلی که ذکر کردید به شدت وسوسه کننده بود. این رو فقط برای دونستن پرسیدم.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۵/۳۰ ۲۲:۵۰:۵۴

این مورد از چند دیدگاه قابل بررسی است:
- بحث lazy loading و eager loading : بارگذاری کردن یا نکردن اطلاعات وابسته به یک شیء تا زمانیکه به آن‌ها نیاز است یا

اینکه هر آنچه نیاز است به یکباره load شود. با eager loading می‌شود تعداد رفت و برگشت‌ها به دیتابیس را در شرایطی که به آن نیاز است، به حداقل رساند.

- بحث دیگر Deferred Execution است (که با lazy loading متفاوت است) و اساس LINQ را تشکیل می‌دهد، فارغ از اینکه LINQ to SQL است یا LINQ to Entities یا زمانیکه شما یک کوئری LINQ را می‌نویسید این کوئری بر روی دیتابیس اجرا نمی‌شود. این اجرا تا زمانیکه شما به اطلاعات آن نیازی نداشته باشید به تأخیر خواهد افتاد. به عبارتی اگر شما 20 عبارت LINQ را پشت سر هم تشکیل دهید حتی یک رفت و برگشت هم به دیتابیس نخواهید داشت. فقط expression tree آن تشکیل می‌شود. همینجا است که مفهوم وجودی SubmitChanges بهتر مشخص می‌شود. شما 10 عبارت تعریف رکوردهای جدید را با LINQ تشکیل دهید. هیچ اتفاقی بر روی دیتابیس رخ نخواهد داد (Deferred Execution). زمانیکه SubmitChanges فراخوانی می‌شود این expression tree نهایی به کدهای SQL متناظر ترجمه خواهد شد.

اطلاعات بیشتر: [+](#)

نویسنده: Meysam

تاریخ: ۱۳۸۹/۰۵/۳۱ ۰۰:۳۶:۰۱

- امنیت: دلیل امنیت فقط در تشخیص SQL-Injection نیست! اگر با ابزارهایی مثل acunetix سایت رو تست کنید، میبینید که همیشه یک threshold در حد MS درآورد که همیشه تشخیص اینجکشن رو داد. بله EF و SP این اشتراک رو دارن که همیشه اینجکت کرد ولی همیشه Vulnerability رو تشخیص داد.

- شخصا به خاطر مشکلات Deployment سعی میکنم BL بیشتر در SP باشه.

- در مورد تغییرات هم عرض کنم، ابزارهایی مثل SQL-ToolBelt مشکل رو حل کردن.

- اینکه برنامه به tightly couple نشه خوبه.

- Round Trip زیاده! به طور مثال فرض کنید یک مقدار تولیدی برای یک فیلد که کلید اصلی است (GUID یا Identity) البته فنون شرقی دارن ولی راه حل استاندارد نیستن!

- تمامی مباحث Loading درست ولی باید یک Trade Off بین بزرگی DB، سرعت نگاشت، قابل اطمینان بودن! (EF برخی قابلیت ها رو نداره اینجا رو نگاه کنید <http://blogs.msdn.com/b/alexj/archive/2009/03/26/index-of-tips.aspx> البته انتظار که ندارید سازنده ش مستقیم بگه نمیشه!)

بحث Change Tracking واقعا عالیه ولی به طور کلی، در یک برنامه Enterprise همیشه از EF استفاده کرد (به همراه OData)

مخلصیم

نویسنده: وحید نصیری

تاریخ: ۱۳۸۹/۰۵/۳۱ ۰۱:۴۰:۳۷

- اصلا راجع به تشخیص بحث نشد. نه SP و نه EF و نه L2SQL و نه NHibernate هیچکدام روش تشخیصی برای درک اینکه آیا ورودی فعلی خطرناک است یا خیر ندارند. کوئری پارامتری است که اینها را حفظ می‌کند: [+](#)

- در مورد توزیع از چه نوع؟ برنامه ویندوزی یا برنامه وب؟ برنامه‌های ویندوزی از نظر من اصلا برای یک سازمان مناسب نیستند:

[+](#)

- مثال Round Trip اصلا واضح نبود ...

- مطلبی که لینک دادید بیشتر مربوط به نگارش اول EF بود و الان خیلی تغییر کرده و حالا اگر با EF مشکل دارید از NHibernate استفاده کنید یا موارد دیگر.

- مورد آخر بیشتر یک نظر شخصی است (و محترم) ولی دلایل آن قانع کننده نبودند.

نویسنده: Meysam

تاریخ: ۱۳۸۹/۰۵/۳۱ ۱۳:۳۹:۰۰

همونطور که قبلا عرض کردم، فقط تشخیص اینجکشن نیست (یا به عبارتی اینکه نشه اینجکشن کرد). نکته اصلی اینجاست که مقدار زمان لازم در زمانی که اینجکشن شده تفاوت داره و در SP هیچ تفاوتی بین حالت عادی و حالتی که اینجکشن شده وجود نداره و این مهم از طریق برخی ابزارها قابل تشخیصه (ولو با زمان زیاد).

- بله توزیع ویندوزی مناسب نیست ولی خب گاهی اوقات اجبار از طرف کارگزار وجود داره!
مثال:- <http://blogs.msdn.com/b/adonet/archive/2010/06/28/performance-impact-of-server-side-generated-guids-in-ef.aspx>
-اینکه EF از تمامی مشابه های موجود بهتره شکی ندارم.

نویسنده: Msanaei
تاریخ: ۱۳۹۰/۰۱/۲۲ ۰۸:۰۱:۰۱

سلام
ممکنه بفرمایید نسخه مجانی پروفایلر EF هم وجود داره یا نه؟ اگه بله ازکجا میشه دانلود کرد ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۱/۲۲ ۰۸:۱۶:۲۸

مجانی نیست ولی می‌تونید از آدرس زیر یک مجوز رایگان یک ماهه دریافت کنید. به آدرس ایمیل شما ارسال خواهد شد
<http://efprof.com/Download>

اصلی ترین مزیت این پکیج ،امکان جداکردن dataModel و Metadata در پروژه یا اسمبلی جداگانه است . در حالیکه WCF RIA Service استاندارد فاقد این قابلیت میباشد و باید dataModel و Metadata در یک پروژه و در یک namespace تعریف شوند. برای استفاده از FluentMetadata :

1) ابتدا فرض می کنیم که کلاس Product ما در یک اسمبلی دیگر به نام DataModel تعریف شده است ، بصورت زیر :

```
public class Product
{
    public int ProductId { get; set; }
    public string ProductName { get; set; }
    public long ProductPrice { get; set; }
}
```

2) حال یک پروژه جدید به نام DataModelsMetadata تعریف می کنیم و ارجاعی به اسمبلی بالا یعنی DataModel نیز به آن اضافه می کنیم .

2-1) ابتدا باید پکیج FluentMetadata را توسط Nuget نصب کرد. [راهنمای نصب](#)
2-2) سپس کلاس های Metadata موردنظر خود را برای کلاس Product تعریف میکنیم .

```
public class ProductMetadata : MetadataClass<Product>
{
    public ProductMetadata ()
    {
        this.Validation(x => x.ProductName).Required("عنوان محصول وارد نشده است");
        this.Validation(x => x.ProductPrice).Range(1000,100000,"قیمت محصول باید بین هزار تومان تا صد هزار تومان باشد");
    }
}
```

2-3) سپس یک کلاس MetadataConfiguration که برای نمونه سازی تمام کلاس های متادیتا ایجاد می کنیم.

```
public class FluentMetadataConfiguration : IFluentMetadataConfiguration
{
    public void OnTypeCreation(MetadataContainer metadataContainer)
    {
        metadataContainer.Add(new ProductMetadata());
    }
}
```

2-4) در آخر اضافه کردن MetadataConfiguration ایجاد شده به Domain Service توسط ویژگی FluentMetadata.

```
[EnableClientAccess()]
[FluentMetadata(typeof(FluentMetadataConfiguration))]
public class FluentMetadataTestDomainService : DomainService
{
    ...
}
```

Entity framework 5 نسبت به نسخه‌های پیشین شاهد تغییرات بسیاری بوده است و مانند هر تغییر دیگری اینجا نیز ممکن است تغییرات ؛ باعث بروز مشکلاتی در روند توسعه نرم افزار شوند. EF در نسخه جدید خود در کدهای پشت صحنه Model به جایObjectContext از DbContext که نسخه محدود شدهObjectContext می‌باشد استفاده می‌کند. همین امر به خودی خود باعث محدود شدن متدهای شئی Context شده است. متدها و خواصی که گاهی برای انجام اعمال خاصی به آنها نیاز پیدا می‌کنیم ولی دیگر در دسترس نیستند. برای مثال برای یک برنامه خاص می‌خواستیم مقدار CommandTimeout را به صورت دستی برای شئی Context تنظیم کنیم ؛ ولی کد زیر دیگر قابل استفاده نبود:

```
context.CommandTimeout = 180;
```

همچنین این استفاده از DbContext در هنگام استفاده از WCF Ria در سیلورلایت باعث بروز مشکل شده و کلاس‌های مدل در هنگام تعریف Domain Service Class توسط WCF Ria قابل شناسایی نیستند. یعنی WCF RIA به صورت خودکار قادر به تشخیص کلاس‌های Model نمی‌باشد.

×

Add New Domain Service Class

Domain Service class name:

☒ Enable client access
☐ Expose OData endpoint

Available context classes:

Some Entity Framework context classes may have been excluded.
[More information can be found in the knowledge base.](#)

Entities	Enable editing

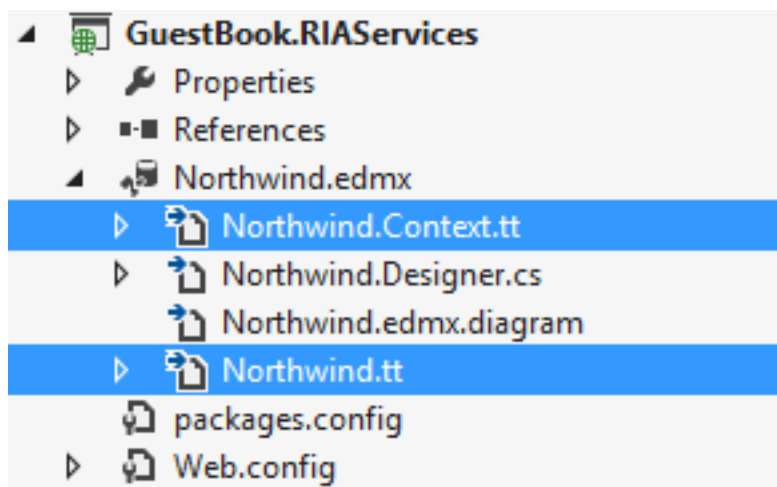
☐ Generate associated classes for metadata

OK

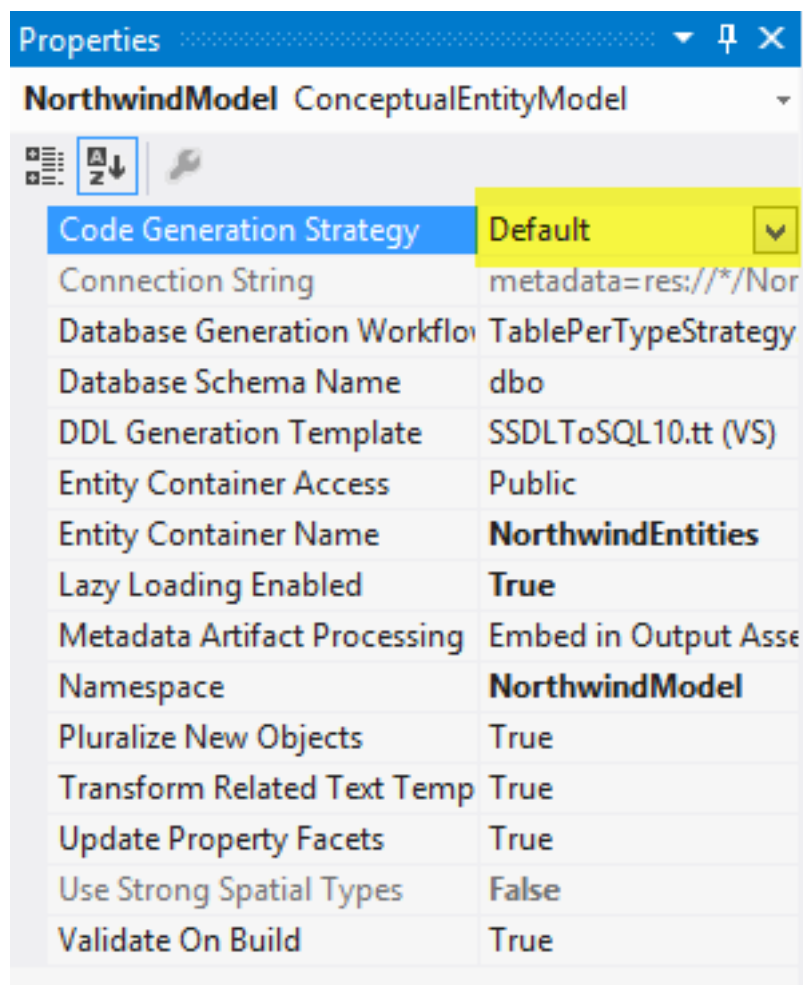
Cancel

برای رفع این مشکل مراحل زیر را انجام دهید:

دو فایل tt موجود در مدل را حذف نمایید.



2. مدل را در حالت Designer باز کنید و در بخش خصوصیات مدل مقدار Code Generation Strategy را از None به Default تغییر دهید.



3. پروژه را Rebuild نمایید. مشکل به همین سادگی رفع می‌شود.
حالا با خیال راحت می‌توانید کلاس‌های مدل را در پنجره Add New Domain Service Class مشاهده نمایید.

Add New Domain Service Class

Domain Service class name:

NorthwindDomainService

☒ Enable client access

☒ Expose OData endpoint

Available context classes:

NorthwindEntities (Entity Framework)

Some Entity Framework context classes may have been excluded.
[More information can be found in the knowledge base.](#)

Entities	Enable editing
<input type="checkbox"/> Category	<input type="checkbox"/>
<input type="checkbox"/> Customer	<input type="checkbox"/>
<input type="checkbox"/> CustomerDemographic	<input type="checkbox"/>
<input type="checkbox"/> Employee	<input type="checkbox"/>
<input type="checkbox"/> Order	<input type="checkbox"/>
<input type="checkbox"/> Order_Detail	<input type="checkbox"/>
<input type="checkbox"/> Product	<input type="checkbox"/>
<input type="checkbox"/> Region	<input type="checkbox"/>
<input type="checkbox"/> Shipper	<input type="checkbox"/>
<input type="checkbox"/> Supplier	<input type="checkbox"/>
<input type="checkbox"/> Territory	<input type="checkbox"/>

☐ Generate associated classes for metadata

OK Cancel

نظرات خوانندگان

نویسنده: سعید

تاریخ: ۱۹:۱۵ ۱۳۹۱/۰۹/۳۰

سلام، [از به روز رسانی جدید](#) ria services استفاده می‌کنید؟

dbcontext در code first هست. شما از db first استفاده کردید؟

این مدل‌ها قبلاً هم ظاهر نمی‌شدند. یعنی چون بر اساس reflection کار می‌کند برنامه یکبار باید کامپایل شود.

نویسنده: حسین مرادی نیا

تاریخ: ۱۹:۲۴ ۱۳۹۱/۰۹/۳۰

خیر

از این به روز رسانی استفاده نکردم.

این مورد برای حالت Database First ذکر شده و مشکل به این روش حل می‌شود.

کنجاکو شدم بدونم که این مشکل در این به روز رسانی حل شده یا نه! اگر کسی استفاده کرده به ما هم بگه.

نویسنده: محسن

تاریخ: ۱۹:۲۴ ۱۳۹۱/۰۹/۳۰

در code first هم همیشه به object context دسترسی پیدا کرد:

```
public class YourContext : DbContext
{
    public YourContext()
        : base(".....ConnectionString.....")
    {
        //ObjectContext of DbContext
        var objectContext = (this as IObjectContextAdapter).ObjectContext;
        objectContext.CommandTimeout = 120;
    }
}
```

نویسنده: ناصر فرجی

تاریخ: ۱۷:۵۰ ۱۳۹۲/۰۱/۱۵

این مشکل توی یک پروژه جدید با 2012 vs کلی وقت من رو هدر داد! بالاخره هم به صورت شانس حلش کردم. کاش زودتر این مطلب رو دیده بودم (:

نویسنده: احسان

تاریخ: ۱۰:۲۱ ۱۳۹۲/۰۱/۱۹

راه حل ارائه شده باعث می‌شه که شما قابلیت‌های DbContext رو از دست بدهید و با همون اشیاء Object Context کار کنید. در این حالت دیگه فکر نکنم استفاده از EF5 ضرورتی داشته باشه.

راه حل صحیح اینه که RiaServices.EntityFramework را با استفاده از Manage Neget Packages رو نصب بفرمائید تا Ria service بتونه اشیاء DbContext رو هم ساپورت کنه. شاد و پیروز باشید.

نویسنده: محسن خان

تاریخ: ۱۳:۲ ۱۳۹۲/۰۱/۱۹

DbContext مطابق کدی که چند سطر بالاتر نوشته شده فقط یک Wrapper است برایObjectContext. چیز جدیدی این وسط اختراع نشده. گل همان گل است.

نویسنده: احسان
تاریخ: ۱۳۹۲/۰۱/۱۹ ۱۳:۱۳

منظور از اختراع نمی‌دونم چیه؟ ولی DbContext دارای قابلیت‌های اضافی بسیاری نسبت بهObjectContext هست. مثل تابع Find و
بله حرف شما متین هست و DbContext ازObjectContext ارث می‌بره.
شاد و پیروز باشید

نویسنده: ناصر فرجی
تاریخ: ۱۳۹۲/۰۱/۲۱ ۱۳:۴۹

میشه بگید توی یک برنامه asp.net ساده که توی vs2012 نوشته شده راه حل چیه؟ چون در صورتی که Code Generation Strategy رو تغییر ندیم دیگه به مدل دسترسی نداریم؟ (روش database first)

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۱/۲۱ ۱۳:۵۹

روش database first همان روش استفاده مستقیم ازObjectContext است.

ضمناً چرا در یک برنامه ASP.NET ازRIA Services استفاده کردید؟ RIA Services بهینه شده برای استفاده در Silverlight و فناوری‌های مانند اون.

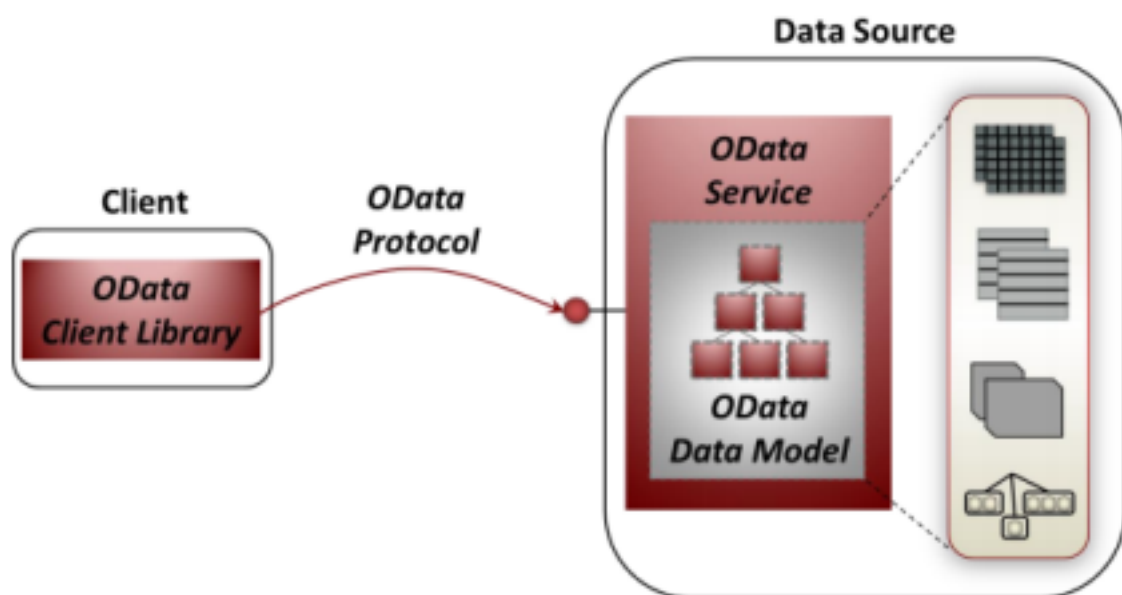
نویسنده: ناصر فرجی
تاریخ: ۱۳۹۲/۰۱/۲۱ ۱۴:۰۴

بنده ازRIA Services استفاده نکردم. در vs2012 هنگام کار با asp.net web forms پیش فرض هنگامی که یک مدل رو به برنامه اضافه می‌کنید به مدل توی کلاس‌ها و فرم‌ها دسترسی ندارید. تنها راه اینه که برید Code Generation Strategy رو از none به default تغییر بدید تا برنامه به درستی کار بکنه و بتونید مدل رو توی namespace‌ها داشته باشید. میخاستم ببینم این کاری که من انجام میدم درسته یا باید راه دیگه ای رو انجام بدم؟

مقدمه

OData

قراردادی برای دسترسی به داده‌ها است که مایکروسافت آن را تحت مجوز [Microsoft Open Specification Promise](#) منتشر کرده است. این قرارداد استاندارد [CRUD](#) ایی را برای دسترسی به منبع داده از طریق وب سایت طراحی نموده است که از JDBC و ODBC ساده‌تر بوده و محدودیت ارتباط فقط با پایگاه داده‌های SQL ایی را ندارد. OData از روی Atom Publishing Protocol و JSON ساخته شده و از مدل REST برای همه در خواست‌های خود استفاده می‌نماید. OData در واقع یک راه مشترک برای هر نوع کلاینت برای دسترسی به هر نوع داده ای است.



OData چهار قسمت اصلی دارد :

OData data model که یک راه عمومی برای مدیریت و توصیف داده‌ها را فراهم می‌نماید
 OData protocol که به کلاینت اجازه ایجاد درخواست و پاسخ از سرویس دهنده OData را می‌دهد.
 OData client libraries که امکان ساخت ساده‌تر نرم افزارها برای دسترسی به داده‌ها با قرارداد OData را می‌دهد.
 OData service سرویس دهنده و امکان دسترسی به داده‌ها را فراهم می‌سازد.

از مزیت‌های OData می توان به موارد زیر اشاره نمود:

ساده و انعطاف پذیر

سورس باز بودن

امکان استفاده در سیستم‌های با داده‌های رابطه ای و غیر رابطه ای

امکان استفاده از داده‌ها با منابع ای که آدرس پذیر هستند یعنی دسترسی از طریق Uri

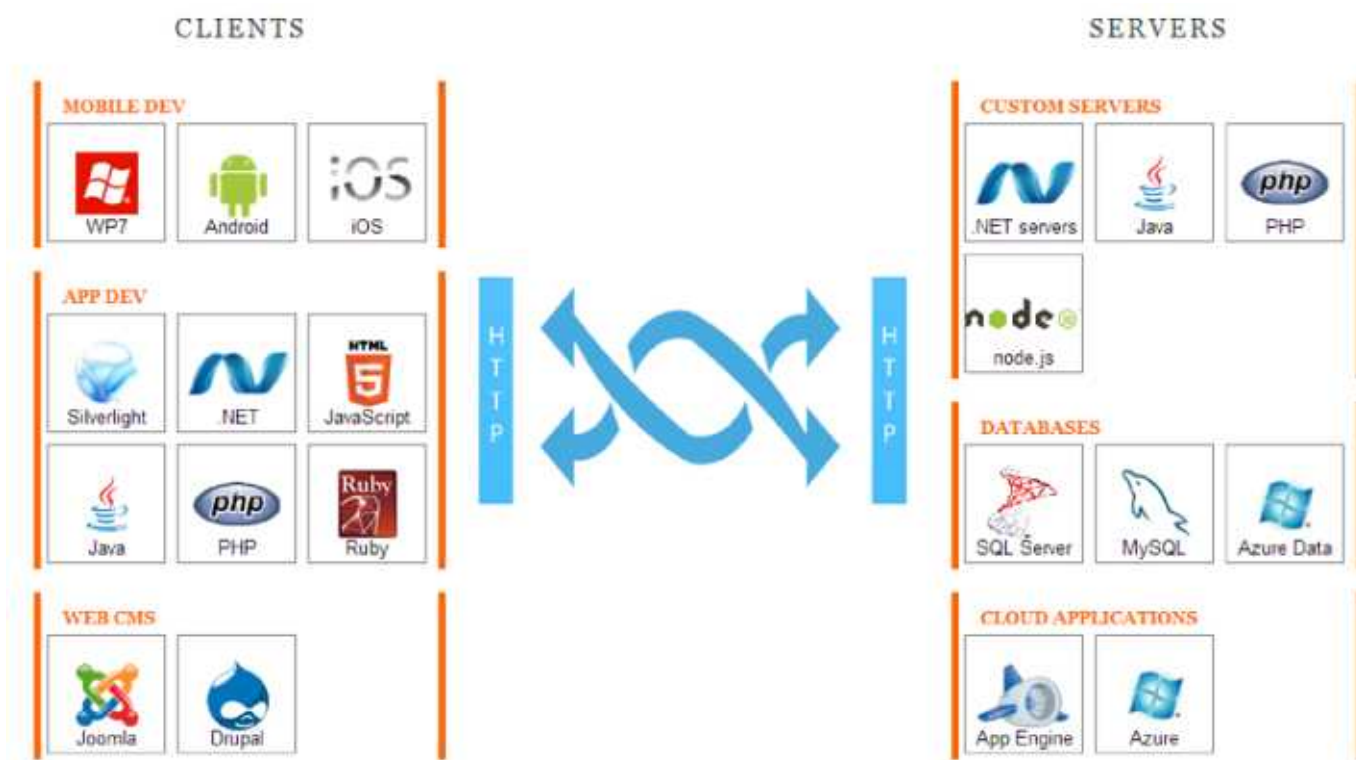
امکان دسترسی هر نوع گیرنده ای به داده ها

امکان نمایش خروجی با فرمت Xml یا Json

...

کتابخانه‌های کار بار OData:

کتابخانه‌های بسیاری برای odata نوشته شده است که امکان استفاده آن را در اکثر زبان‌ها مهیا می‌سازد.



اما بهترین کتابخانه [WCF Data Services](#) است که از سوی مایکروسافت ارائه شده و در اکثر تکنولوژی‌ها و محصولات خود قابلیت استفاده را دارد. WCF Data Services با پیاده سازی قرارداد OData ، توسعه دهندگان را از سطح پایین این قرارداد رهایی ساخته و به راحتی می‌توانند از ساختار شی گرا برنامه خود، در سرویس دهی با OData استفاده نمایند.

کاربردهای OData:

OData یک قرارداد سرویس دهی بر روی وب است که به هر نوع گیرنده که امکان دسترسی به وب را داشته باشد، امکان سرویس دهی دارد. به همین خاطر در اکثر برنامه‌های تحت وب یا نرم افزارهای موبایل که می‌خواهیم اطلاعاتی را مابین سرویس دهنده و گیرنده رد و بدل کنیم حتی زمانیکه platformهای مختلفی در کار باشند OData بهترین گزینه است. در مطالب بعدی با پیاده سازی مثال‌های با استفاده از WCF Data Services بیشتر با OData آشنا خواهید شد. در این اینجا هدف آشنایی اولیه با OData و کاربردهای آن بود که امیدوارم مفید واقع شده باشد.

نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۲۱:۱۶ ۱۳۹۱/۱۰/۰۳

ممنون، لطفا ادامه بدید ...

نویسنده: پرهام
تاریخ: ۷:۵۷ ۱۳۹۱/۱۰/۰۴

ممنون. منتظر مطالب بعدی هستیم.

نویسنده: MegaMan
تاریخ: ۱۵:۴۷ ۱۳۹۱/۱۰/۱۷

جالب بود والبته مفید (:

عنوان: قراردادهای کوثری نویسی در OData و WCF Data Service - قسمت دوم

نویسنده: مجتبی کاویانی

تاریخ: ۱۳۹۱/۱۰/۲۵ ۱۶:۳۰

آدرس: www.dotnettips.info

برچسب‌ها: WCF RIA Services, OData, WCF Data Services

در مطلب قبلی [قراردادهای کوثری نویسی در OData و WCF Data Service - قسمت اول](#) با قراردادهای کوثری نویسی آشنا شدید در این مطلب به جزئیات بیشتر این قراردادها می‌پردازیم.

عملگرهای منطقی

در OData نه عملگر منطقی داریم که امکان مقایسه منطقی در عبارات‌های شرطی را در اختیارمان قرار می‌دهد.

eq عملگر برابری

ne عملگر مخالف

lt عملگر کوچکتری

le عملگر کوچکتر یا مساوی

gt عملگر بزرگتری

ge عملگر بزرگتر یا مساوی

and

or

not

عملگرهای ریاضی

پنج عملگر ریاضی وجود دارد که امکان انجام عملیات ریاضی در کوثری را میسر می‌سازد.

add جمع دو عملوند

sub تفریق دو عملوند

mul ضرب دو عملوند

div تقسیم دو عملوند

mod باقیمانده تقسیم دو عملوند

در آخر () برای گروه بنده و اولویت دهی عبارات کاربرد دارد.

توابع عبارت‌های کوثری نویسی

در OData چهار دسته توابع داریم

String Functions در جدول زیر این توابع با توضیح آن آورده شده است:

	آیا رشته p0 شامل رشته p1 هست؟
(bool substringof(string p0, string p1	مثال: http://services.odata.org/Northwind.svc/Customers?filter=substringof('Alfreds', CompanyName) eq true شرح: مشتریانی که نام شرکت آنها شامل رشته Alfreds باشد
(bool endswith(string p0, string p1	آیا رشته p0 با رشته p1 پایان می‌یابد؟ مثال: http://services.odata.org/Northwind/Northwind.svc/Customerstomers?filter=endswith(CompanyName, 'Futterkiste') شرح: مشتریانی که نام شرکت آنها با رشته FutterKiste پایان می‌یابد
(bool startswith(string p0, string p1	آیا رشته p0 با رشته p1 آغاز می‌شود؟ مثال: http://services.odata.org/Northwind.svc/Customers?

	آیا رشته p0 شامل رشته p1 هست؟
(bool substringof(string p0, string p1)	مثال: <code>http://services.odata.org/Northwind.svc/Customers?\$filter=substringof('Alfreds', CompanyName) eq true</code> شرح: مشتریانی که نام شرکت آنها شامل رشته Alfreds باشد
	(<code>\$filter=startswith(CompanyName, 'Alfr</code> شرح: مشتریانی که نام شرکت آنها با رشته Alfer آغاز می‌یابد
int length(string p0)	محاسبه طول رشته دریافتی. مثال: <code>http://services.odata.org/Northwind/Northwind.svc/Customers?\$filter=length(CompanyName) eq 19</code> شرح: مشتریانی که طول نام شرکت آنها برابر 19 باشد
int indexof(string arg)	برگشت اندیس رشته ورودی مثال: <code>http://services.odata.org/Northwind.svc/Customers?\$filter=indexof(CompanyName, 'lfreds') eq 1</code> شرح: مشتریانی که نام شرکت آنها با رشته lfreds که از کارکتر دوم شروع می‌یابد
(string replace(string p0, string find, string replace	جایگزینی یک رشته در رشته دیگر مثال: <code>http://services.odata.org/Northwind.svc/Customers?\$filter=replace(CompanyName, ' ', '') eq "AlfredsFutterkiste"</code> شرح: مشتریانی که نام شرکت آنها بدون فاصله برابر AlfredsFutterkiste باشد با پرکردن فاصله با جای خالی
(string substring(string p0, int pos (string substring(string p0, int pos, int length	برگرداندن رشته ای از رشته دیگر از شماره اندیس ورودی یا از شماره اندیس تا طول رشته ورودی مثال: <code>http://services.odata.org/Northwind.svc/Customers?\$filter=substring(CompanyName, 1) eq 'lfreds Futterkiste'</code> شرح: مشتریانی که نام شرکت آنها از کاراکتر دوم برابر lfreds Futterkiste باشد <code>http://services.odata.org/Northwind.svc/Customers?\$filter=substring(CompanyName, 1,2) eq 'lf</code>
string tolower(string p0)	برگرداندن رشته ورودی با کارکتر بزرگ
string toupper(string p0)	برگرداندن رشته ورودی با کاراکتر کوچک
string trim(string p0)	حذف کارکترهای Whitespace از دو طرف رشته
(string concat(string p0, string p1	الحاق رشته به هم

int year(DateTime p0)	برگرداندن سال تاریخ ورودی مثال: http://services.odata.org/Northwind.svc/Employees?\$filter=year(BirthDate) eq 1971
int month(DateTime p0)	برگرداندن ماه تاریخ ورودی مثال: http://services.odata.org/Northwind/Northwind.svc/Employees?\$filter=month(BirthDate) eq 5
int days(DateTime p0) int day(DateTime p0)	برگرداندن روز تاریخ ورودی برگرداندن تعداد روز فاصله زمانی مثال: http://services.odata.org/Northwind.svc/Employees?\$filter=day(BirthDate) eq 8
int hour(DateTime p0)	برگرداندن ساعت تاریخ ورودی مثال: http://services.odata.org/Northwind.svc/Employees?\$filter=hour(BirthDate) eq 4
int minute(DateTime p0) int minutes(DateTime p0)	برگرداندن دقیقه تاریخ ورودی مثال: http://services.odata.org/Northwind.svc/Employees?\$filter=minute(BirthDate) eq 40
int second(DateTime p0)	برگرداندن ثانیه تاریخ ورودی مثال: http://services.odata.org/Northwind.svc/Employees?\$filter=second(BirthDate) eq 40

Math Functions

double ceiling(double p0) decimal ceiling(decimal p0)	برگرداندن سقف عدد ورودی مثال: http://services.odata.org/Northwind.svc/Orders?\$filter=ceiling(Freight) eq 32
double floor(double p0) decimal floor(decimal p0)	برگرداندن کف عدد ورودی مثال: http://services.odata.org/Northwind.svc/Orders?\$filter=floor(Freight) eq 32
double round(double p0) decimal round(decimal p0)	گرد کردن عدد ورودی مثال: http://services.odata.org/Northwind.svc/Orders?\$filter=round(Freight) eq 32

Type Functions

<pre>(boolisof(type p0 (boolisof(expression p0, type p1</pre>	<p>برگرداندن نوع داده ورودی</p> <p>مثال: http://services.odata.org/Northwind.svc/Orders?\$filter=ter=isof(Customer, NorthwindModel.MVPCustomer شرح: سفارشی که نوع مشتری آنها برابر MVPCustomer باشد</p>
<pre>(p0> cast(type p0> p1> cast(expression p0, type p1 ></pre>	<p>تبدیل نوع داده ورودی</p>

در آخر چند نکته

برای استفاده از رشته‌ها در عبارات از ' تک کوتشن استفاده نمایید
 برای دستیابی به مقادیر پروپرتی‌ها از \$value استفاده نمایید
 برای دستیابی به آدرس روابطی یک موجودیت از \$links استفاده نمایید
 مثال: [http://services.odata.org/OData/OData.svc/Categories\(1\)/\\$links/Products](http://services.odata.org/OData/OData.svc/Categories(1)/$links/Products):
 \$select برای محدود کردن پروپرتی‌های یک موجودیت استفاده می‌شود
 مثال: [http://services.odata.org/OData/OData.svc/Products?\\$select=Price,Name](http://services.odata.org/OData/OData.svc/Products?$select=Price,Name):
 از ستاره برای دستیابی به همه پروپرتی‌های یک موجودیت می‌توان استفاده نمود
 مثال: [http://services.odata.org/OData/OData.svc/Products?\\$select=*](http://services.odata.org/OData/OData.svc/Products?$select=*):
 ادامه دارد...