

عنوان: SQL تولیدی در NHibernate از کدام متد صادر شده است؟

نویسنده: وحید نصیری

تاریخ: ۱۳۸۹/۱۰/۱۳ ۱۲:۴۴:۰۰

آدرس: www.dotnettips.info

برچسب‌ها: NHibernate

اگر مطلب "[ذخیره سازی SQL تولیدی در NH3](#)" را دنبال کرده باشید که یک مثال عملی از "[NHibernate 3.0 و عدم وابستگی مستقیم به Log4Net](#)" بود، خروجی حاصل از آن به صورت زیر است:

```
---+ 12/29/2010 05:35:59.75 +---
SQL ...

---+ 12/29/2010 05:35:59.75 +---
SQL ...
```

و پس از مدتی این فایل هیچ حسی را منتقل نمی‌کند! یک سری SQL که لاگ شده‌اند. مشخص نیست کدام متد در کدام کلاس و کدام فضای نام، سبب صدور این عبارت SQL ثبت شده، گردیده است. خوشبختانه در دات نت فریم ورک می‌توان با بررسی Stack trace، رد کاملی را از فراخوان‌های متدها یافت:

```
StackTrace stackTrace = new StackTrace();
StackFrame stackFrame = stackTrace.GetFrame(1);
MethodBase methodBase = stackFrame.GetMethod();
Type callingType=methodBase.DeclaringType;
```

با بررسی StackFrame ها امکان یافتن نام متدها، فضاهای نام و غیره میسر است. مثلاً یکی از کاربردهای مهم این روش، ثبت فراخوان‌های متدی است که استثنایی را ثبت کرده است. بر این اساس سورس مثال قبل را جهت درج اطلاعات فراخوان‌های متدها تکمیل کرده‌ام، که [از این آدرس](#) قابل دریافت است.

اکنون اگر از این ماژول جدید استفاده کنیم، خروجی نمونه‌ی آن به صورت زیر خواهد بود:

```
---+ 01/02/2011 02:19:24.98 +---
-- Void ASP.feedback_aspx.ProcessRequest(System.Web.HttpContext) [File=App_Web_4nvdip40.5.cs, Line=0]
-- Void Prog.Web.UserCtrls.FeedbacksList.Page_Load(System.Object, System.EventArgs)
[File=FeedbacksList.ascx.cs, Line=23]
---- Void Prog.Web.UserCtrls.FeedbacksList.BindTo() [File=FeedbacksList.ascx.cs, Line=43]
----- System.Collections.Generic.IList`1[Feedback] Prog.GetAllUserFeedbacks(Int32)
[File=FeedbackWebRepository.cs, Line=66]

SELECT ...
@p0 = 3 [Type: Int32 (0)]
```

به این معنا که عبارت SQL ثبت شده، حاصل از پردازش صفحه‌ی feedback.aspx، سپس متد Page_Load آن که از یوزر کنترل FeedbacksList.ascx استفاده می‌کند، می‌باشد. در اینجا فراخوانی متد BindTo سبب فراخوانی متد GetAllUserFeedbacks در فایل FeedbackWebRepository.cs واقع در سطر 66 آن گردیده است. اینطوری حداقل می‌توان دریافت که SQL تولیدی دقیقاً به کجا بر می‌گردد و چه متدی سبب صدور آن شده است.

ملاحظات:

این ماژول تنها در صورت وجود فایل pdb معتبر کنار اسمبلی‌های شما این خروجی مفصل را تولید خواهد کرد. در غیر اینصورت از آن صرف‌نظر می‌کند. (برای مثال نام فایل سورس فراخوان، شماره‌ی سطر فراخوان، حتی محل قرارگیری آن فایل بر روی کامپیوتر شما در فایل‌های pdb ثبت می‌گردند؛ به همین جهت توصیه اکید حین ارائه‌ی نهایی برنامه، حذف این نوع فایل‌ها است)

نظرات خوانندگان

نویسنده: Meysam

تاریخ: ۱۳:۲۱:۲۴ ۱۳۸۹/۱۰/۱۳

استفاده از StackTrace برای پیاده سازی INPC به نظرتون مشکل Performance ایجاد میکنه؟

<http://csharperimage.jeremylikness.com/2010/12/jounce-part-8-raising-property-changed.html>

نویسنده: وحید نصیری

تاریخ: ۱۶:۵۱:۰۲ ۱۳۸۹/۱۰/۱۳

بله، در تعداد رکورد بالا مثلاً در یک گرید در صفحه، حتماً مشکل‌زا است. کلاً روش در این مورد زیاد هست، منجمله روشی که در قسمت 5 آموزش MVVM در سایت جاری هست. یا روشی که شما ذکر کردید، یا یک روش دیگر هم استفاده از فریم ورک‌های AOP است. این‌ها روی کد IL نهایی تاثیر می‌گذارند.

ولی در نهایت همان روش سنتی استفاده از رشته‌ها، هر چند کمی طولانی‌تر است، اما بهترین کارایی و کمترین سربار را هم دارد.

مطلبی هم که من در اینجا عنوان کردم در مورد دیباگ یک سیستم مبتنی بر NHibernate هست و بدیهی است قرار نیست در محیط کاری از آن استفاده شود.

نویسنده: وحید نصیری

تاریخ: ۱۸:۰۴:۰۷ ۱۳۸۹/۱۰/۳۰

نگارش کامل شده‌ی این پروژه را از آدرس زیر دریافت کنید:

<http://nh3sqllogger.codeplex.com>