

استفاده از Log4Net جهت ثبت خروجی‌های SQL حاصل از NHibernate

هنگام استفاده از NHibernate، پس از افزودن ارجاعات لازم به اسمبلی‌های مورد نیاز آن به برنامه، یکی از اسمبلی‌هایی که به پوشه build برنامه به صورت خودکار کپی می‌شود، فایل log4net.dll است (حتی اگر ارجاعی را به آن اضافه نکرده باشیم) که جهت ثبت وقایع مرتبط با NHibernate مورد استفاده قرار می‌گیرد. خوب اگر مجبوریم که این وابستگی کتابخانه NHibernate را نیز در پروژه‌های خود داشته باشیم، چرا از آن استفاده نکنیم؟! شرح مفصل استفاده از این کتابخانه سورس باز را در سایت اصلی آن می‌توان مشاهده کرد:

[Log4Net](#)

برای اینکه از این کتابخانه در برنامه خود جهت ثبت عبارات SQL تولیدی توسط NHibernate استفاده کنیم، باید مراحل زیر طی شوند:

الف) ارجاعی را به اسمبلی log4net.dll اضافه نمائید (کنار اسمبلی NHibernate در پوشه build برنامه باید موجود باشد)
 ب) فایل app.config برنامه را (برنامه ویندوزی) به صورت زیر ویرایش کرده و چند سطر مربوطه را اضافه نمائید (در مورد برنامه‌های وب هم به همین شکل است). configSections فایل web.config تنظیم شده و سپس تنظیمات log4net را قبل از بسته شدن تگ configuration اضافه نمائید) :

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <section name="log4net"
      type="log4net.Config.Log4NetConfigurationSectionHandler,log4net" />
  </configSections>

  <connectionStrings>
    <!--NHSessionManager-->
    <add name="DbConnectionString"
      connectionString="Data Source=(local);Initial Catalog=HelloNHibernate;Integrated Security =
true"/>
  </connectionStrings>

  <log4net>
    <appender name="rollingFile"
      type="log4net.Appender.RollingFileAppender,log4net" >
      <param name="File" value="NHibernate_Log.txt" />
      <param name="AppendToFile" value="true" />
      <param name="DatePattern" value="yyyy.MM.dd" />
      <rollingStyle value="Size" />
      <maxSizeRollBackups value="10" />
      <maximumFileSize value="500KB" />
      <staticLogFileName value="true" />
      <layout type="log4net.Layout.PatternLayout,log4net">
        <conversionPattern value="%d %p %m%n" />
      </layout>
    </appender>
    <logger name="NHibernate.SQL">
      <level value="ALL" />
      <appender-ref ref="rollingFile" />
    </logger>
  </log4net>
</configuration>
```

ج) سپس باید فراخوانی زیر نیز در ابتدای کار برنامه صورت گیرد:

```
log4net.Config.XmlConfigurator.Configure();
```

در یک برنامه ASP.Net این فراخوانی باید در Application_Start فایل Global.asax.cs صورت گیرد.
یا در یک برنامه از نوع WinForms تنها کافی است سطر زیر را به فایل AssemblyInfo.cs برنامه اضافه کرد:

```
// Configure log4net using the .config file
[assembly: log4net.Config.XmlConfigurator(Watch = true)]
```

یا این سطر را به فایل Global.asax.cs یک برنامه ASP.Net نیز می‌توان اضافه کرد. Watch=true آن، با کمک FileSystemWatcher تغییرات فایل کانفیگ را تحت نظر داشته و هر بار که تغییر کند بلافاصله، تغییرات جدید را اعمال خواهد کرد.

د) هنگام استفاده از کتابخانه Fluent NHibernate حتما باید متد ShowSql در جایی که دیتابیس برنامه را تنظیم می‌کنیم (Fluently.Configure()).Database) ذکر گردد (که نمونه آن را در مثال‌های قسمت‌های قبل مشاهده کرده‌اید).

توضیحاتی در مورد تنظیمات فوق:

configSections حتما باید در ابتدای فایل app.config ذکر شود و گر نه برنامه کار نخواهد کرد.

سپس کانکشن استرینگ مورد استفاده در قسمت کانفیگ برنامه ذکر شده است.

در ادامه تنظیمات استاندارد مربوط به log4net را مشاهده می‌کنید.

در تنظیمات این کتابخانه، appender مشخص کننده محل ثبت وقایع است. زمانی که از RollingFileAppender استفاده کنیم، اطلاعات را در یک سری فایل ذخیره خواهد کرد (امکان ثبت وقایع در EventLog ویندوز، ارسال از طریق ایمیل و غیره نیز میسر است که جهت توضیحات بیشتر می‌توان به مستندات آن رجوع نمود).

سپس نام فایلی که اطلاعات وقایع در آن ثبت خواهند شد ذکر شده است (برای مثال NHibernate_Log.txt)، در ادامه مشخص گردیده که اطلاعات باید هر بار به این فایل Append و اضافه شوند. سطرهای بعدی مشخص می‌کنند که هر زمانیکه این لاگ فایل به 10 مگابایت رسید، یک فایل جدید تولید کن و هر بار 10 فایل آخر را نگه دار و مابقی فایل‌های قدیمی را حذف کن. در قسمت PatternLayout مشخصات می‌کنیم که خروجی ثبت شده با چه فرمتی باشد. برای مثال یک سطر خروجی مطابق با تنظیمات فوق به شکل زیر خواهد بود:

```
2009-10-18 20:03:54,187 DEBUG INSERT INTO [Student] (Name) VALUES (@p0); select SCOPE_IDENTITY();@p0 = 'Vahid'
```

در قسمت Logger یک نام دلخواه ذکر شده و میزان اطلاعاتی که باید درج شود، از طریق مقدار level مورد نظر، قابل تنظیم است که می‌تواند یکی از مقادیر ALL, DEBUG, INFO, WARN, ERROR, FATAL باشد. اینجا level در نظر گرفته شده ALL است که تمامی اطلاعات مرتبط با اعمال پشت صحنه NHibernate را لاگ خواهد کرد.
توسط appender-ref آن appender ای را که در ابتدای کار تعریف و تنظیم کردیم، مشخص خواهیم کرد.

اگر هم با برنامه نویسی بخواهیم اطلاعاتی را به این لاگ فایل اضافه کنیم تنها کافی است بنویسیم:

```
log4net.LogManager.GetLogger("NHibernate.SQL").Info("test1");
```

[اطلاعات بیشتر](#)

ادامه دارد ...