

چقدر خوب می‌شد اگر IDE ویژوال استودیو دات نت اشیایی از نوع disposable را برای ما مشخص می‌کرد و در ادامه پیشنهاد استفاده از عبارت [using](#) را برای آن‌ها می‌داد؟
سؤال: این مساله چه اهمیتی دارد؟
به مثال متداول زیر دقت بفرمائید:

```
string connString = "...";
SqlConnection conn = new SqlConnection(connString);
string cmdString = "select * from author where name=N'test'";
SqlCommand cmd = new SqlCommand(cmdString, conn);
conn.Open();
SqlDataReader reader = cmd.ExecuteReader();
while(reader.Read())
{
    ...
}
conn.Close();
```

این مثالی است که در اکثر سایت‌های آموزشی می‌توان یافت. باز کردن کانکشن به دیتابیس، اجرای کوئری و دریافت اطلاعات حاصل و در آخر بستن کانکشن. بسیار هم خوب! در این مثال اگر name مساوی test وجود نداشته باشد، روند اجرای برنامه در اولین reader.Read با یک exception متوقف خواهد شد. یعنی عملاً ممکن است به بستن کانکشن نرسیم (هیچ الزامی وجود ندارد که کدهای ما سطر به سطر اجرا شوند). این مورد چه مشکلی را ایجاد خواهد کرد؟ فقط کافی است سایت ما 100 نفر بازدید کننده داشته باشد. در مدت زمان کوتاهی کل سایت با خطای زیر از کار می‌افتد و پیغامی که از کاربران دریافت خواهید کرد این است: "باز هم برنامه کند شد، "بازهم" سایت کار نمی‌کند ..."

Exception Details: System.InvalidOperationException: Timeout expired. The timeout period elapsed prior to obtaining a connection from the pool. This may have occurred because all pooled connections were in use and max pool size was reached.

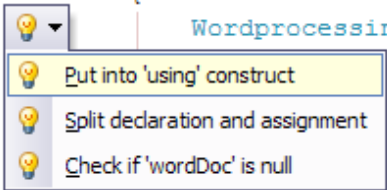
روش صحیح برای مقابله با این نوع خطاها قرار دادن قسمت بستن کانکشن در قسمت finally یک try/finally است تا مطمئن باشیم که حتماً این قسمت اجرا خواهد شد. روش زیباتر این کار استفاده از عبارت using است. عبارت using نیز در نهایت به یک try/finally توسط کامپایلر [بسط داده خواهد شد](#) اما کد حاصل در این حالت خواناتر است. بنابراین در این حالت حتی اگر خطایی نیز رخ دهد، شیء مورد نظر حتماً dispose خواهد شد. (از دات نت 2 به بعد عبارت using به VB.Net نیز [افزوده شده](#) است)

```
using (SqlConnection connection =
    new SqlConnection(connectionString))
{
    ...
}
```

عبارت using را تنها در مورد اشیایی از نوع IDisposable می‌توان بکار برد. یکی از توانایی‌های پلاگین Agent Johnson تشخیص این مورد و گوشزد کردن آن است (شکل زیر):

```
18 2 public static void CreateDocument2(string documentFileName, string text)
19 {
20     WordprocessingDocument wordDoc = WordprocessingDocument.Create(document
21     ... Wordproc
```

```
18 2 public static void CreateDocument2(string documentFileName, string text)
19 {
20     WordprocessingDocument wordDoc = WordprocessingDocument.Create(document
21     ... Wordproc
22
23
24
25
```



Agent Johnson در حقیقت پلاگینی است برای پلاگین دیگری به نام [Resharper](http://resharper.com) . این پلاگین را از آدرس زیر می‌توان دریافت کرد (احتمالا به یک پروکسی نیاز خواهد بود ...):

<http://code.google.com/p/agentjohnsonplugin/>

البته بحث تشخیص اشیاء disposable یکی از توانایی‌های آن است. سایر موارد را در آدرس فوق می‌توانید مشاهده نمایید.

نظرات خوانندگان

نویسنده: مهدی معنوی
تاریخ: ۱۳۸۷/۰۹/۱۷ ۲۲:۰۷:۰۰

جالب بود