

Laborator 1: Logging folosind Log4net

Logging = jurnalizare

Intrebări ce trebuie avute în vedere de către programator:

- Ce faci când utilizatorul raportează o eroare, dar nu poate să spună ce anume s-a întâmplat, care a fost contextul?
- Cum faci detectarea erorilor și a contextelor de apariție *dupa* ce aplicația a fost livrată beneficiarului?
- Cum încerci să reproduci o eroare raportată de client?
- În general, cum îți asiguri premisele de a afla dacă ceva nu merge bine în aplicație (accese nepermise, erori de calcul, timpi anormal de mari pentru diverse operații)?

Exemple de utilizări ale jurnalizării:

- Urmărirea potențialelor probleme ce pot apărea în rularea unei aplicații
- Calibrarea performanțelor: prin analiza jurnalelor se pot detecta care sunt zonele din aplicație care consumă mult timp (bottlenecks)
- Obținerea de statistici legate de utilizarea aplicației: cât de des se ajunge într-o anumită secțiune? Câți utilizatori înregistrați/anonimi accesează aplicația într-o perioadă dată?
- Monitorizări de securitate: există încercări de atac de tip brute force (încercarea succesivă de nume și parole, cereri anormal de multe venind de la un anumit client etc.)
- Monitorizarea comunicărilor cu sisteme externe – unele mesaje esuează, altele au un timp de acces mare etc.
- Dacă vrei să știi ce se întâmplă cu aplicația în timpul funcționării ei, jurnalizarea este o necesitate; nu există înlocuitori pentru așa ceva;
- Monitorizarea se poate face și în timpul dezvoltării aplicației, nu doar la rularea ei

Tema de implementat în acest laborator: să se creeze o clasă simplă (model de calculator, sau stivă de întregi etc.) în care partea de procesare să fie însoțită de logging. Se va folosi biblioteca log4net (<http://logging.apache.org/log4net/>).

Pasi:

0. Mașina virtuală – prezentare
1. Se adaugă în proiect biblioteca log4net folosind NuGet.
2. De la adresa <http://logging.apache.org/log4net/> citiți: about, features, exemple;

3. Pasii de urmat: <http://www.justinrhinesmith.com/blog/2008/05/12/quick-and-easy-log4net-setup/> - cu corectia: numele fisierului in care se adauga configurarea de assembly este AssemblyInfo.cs

Nota: in cazul in care nu rezulta jurnalizare (fisierul de log nu se creeaza pentru un appender de tipul FileAppender) se va proceda ca la

<http://stackoverflow.com/questions/3618380/log4net-does-not-write-the-log-file>.

4. (Optional) Pentru a face log4net sa lucreze cu .NET 4+, urmariti <http://tseonet.blogspot.com/2010/07/making-log4net-run-on-net-40.html> (modificarea tinte de compilare) sau <http://www.codeproject.com/KB/dotnet/Log4NetWithClient.aspx?msg=3595219> (recompilare de biblioteca log4net) sau folositi biblioteca log4net din directorul net-cp (cp = client profile). **Se prefera totusi folosirea lui .NET 4+ full, nu client profile.**
5. Faceti log folosind un appender de tip fisier si unul de baza de date.

Nota: utilizati exemplele de configurare de pe site-ul Apache. Preluati schema tabeli in care se va face log de pe acelasi site. Folositi un server de baze de date SQL Server (instalat in masina virtuala) sau MySQL.

Intrebari:

1. De ce la logging in baza de date *nu apare instant* fiecare eveniment jurnalizat?
2. De ce la log.Info() se scrie in log, chiar daca in fisierul de configurare avem setarea:

`<level value="DEBUG"/>?`

3. Exista in log4net suport pentru utilizarea unui logger asincron? Care sunt beneficiile si riscurile unui astfel de appender? Sugerati o modalitate prin care mesajele de log sunt scrise in loturi (precum la append in baza de date), mai degraba decat la fiecare cerere de log – asa cum se procedeaza de exemplu la FileAppender.
4. De citit articolul "Log20: Fully Automated Optimal Placement of Log Printing Statements under Specified Overhead Threshold" de la <http://www.eecg.toronto.edu/~yuan/papers/p125-Zhao.pdf> ce discuta dispunerea automata a instructiunilor de jurnalizare din perspectiva efectelor costului de performanta.

Note:

1. Se sugereaza ca inaintea fiecarei cereri de jurnalizare sa se testeze daca log-ul permite tipul de jurnalizare dorit. Astfel, in loc de:

```
log.Info("S-a apasat click pe button1.");
```

este de preferat sa se scrie:

```
if (log.IsInfoEnabled)
{
    log.Info("S-a apasat click pe button1.");
}
```

```
}
```

Alternativ: se poate face o clasa wrapper care sa expuna un API similar cu cel al lui log4net, sau sa se foloseasca metode de extensie, care in interior sa faca testarea sugerata mai sus.

2. In fisierul de configurare, e bine ca sectiunea

```
<configSections>  
  <section name="log4net"  
type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />  
</configSections>
```

sa fie pusa cat mai la inceputul fisierului.