### Modulul 19. Logging în C#.

#### Ce este procesul de logging? Scopurile și utilitatea

Logging reprezintă procesul de salvare a stării unei aplicații în unele fișiere textuale. Programatorii crează log-uri pentru a putea analiza cum se execută un program în mediu real. De asemenea aceste log-uri pot fi create și de calculatoarele utilizatorilor, și cu acordul lor, trimise developerilor care vor primi informație utilă despre modul în care funcționează aplicație.

Scopul final al procesului de logging este de a identifica unele erori neprevăzute/neidentificate și de a le elimina cât mai rapid.

Nu există o nomenclatură rigidă în privința execuției acestui proces, fiecare programator decide singur ce mesaje și când să stocheze în fișierele loguri.

#### Cel mai simplu program de logging

După cum am menționat anterior, logging-ul este procesul de stocare a unor mesaje în fișiere externe pentru analiză ulterioară. Cel mai simplu exemplu este stocarea unui mesaj.

**Exemplu:**

|  |
| --- |
| namespace Logging  {  class Program  {  static void Main(string[] args)  {  Console.WriteLine("Hello world!");  File.WriteAllText("log.txt", "Program executat cu succes.");  }  }  } |

În exemplul de mai sus am stocat mesajul simplu că programul s-a derulat cu succes.

Dar să considerăm alt **exemplu:**

|  |
| --- |
| namespace Logging  {  class Program  {  static void Main()  {  int z = 5;  int y = 0;  Divide(z, y);  }  private static void Divide(int \_a, int \_b)  {  try  {  int x = \_a / \_b;  Console.WriteLine(x);  File.WriteAllText("log.txt", "Executat cu succes.");  }  catch (Exception e)  {  File.WriteAllText("log.txt", e.Message);  }  }  }  } |

În programul de mai sus am creat o metodă care primește 2 numere ca parametri, pentru a ne asigura că operația se execută corect am implementat un try catch simplu, în cazul în care operația se execută cu succes vom afișa la consolă rezultatul, iar în fișierul log mesajul de succes.

În caz opus vom folosi catch pentru a extrage mesajul excepției, acest lucru ne va permite să identificăm care a fost problema și în care punct a programului a apărut.

#### Practici utile pentru procesul de logging

Deși am menționat anterior că nu există reguli stabilite în privința procesului de logging, este recomandat să logați un set de date care vă vor ajuta să identificați o problemă/excepție în cazul în care aceasta apare:

* **Mesajul excepției.** În mod evident acesta este primul lucru de care aveți nevoie pentru a identifica problema.
* **Numele clasei și metodei care a generat excepția.** Acestea vă vor ajuta să identificați locul exact de unde a provenit excepția.
* **Sistemul de operare.** În cazul în care aveți o aplicație care se derulează pe mai multe sisteme de operare această informație poate fi vitală.

|  |
| --- |
| namespace Logging  {  class Program  {  static int z = 5;  static int y = 0;  static void Main()  {  Program prog = new Program();  prog.Operatie();  }  private void Operatie()  {  try  {  int x = z / y;  Console.WriteLine(x);  File.WriteAllText("log.txt", "Executat cu succes.");  }  catch (Exception e)  {  //Numele metodei  var st = new StackTrace();  var sf = st.GetFrame(1);  File.WriteAllText("log.txt", String.Format("Exceptia: {0} \nNumele clasei:{1} \nNumele metodei:{2} \nSistemul de operare:{3}", e.Message, GetType().Name, sf.GetMethod().Name, Environment.OSVersion.ToString()));  }  }  }  } |

* **Limba dispozitivului.** În unele cazuri excepțiile pot fi provocate de faptul că dispozitivul nu poate citi/afișa caracterele pe care le-ați preconizat.
* **Data și ora apariției.** Vă ajută mai mult din punct de vedere organizatoric și cronologic, dar în unele cazuri utilizatorii pot schimba artificial data/ora pentru a obține beneficii.
* **Numele utilizatorului, email etc.** Datele de contact vă vor ajuta, evident, să contactați persoana care are nevoie de suport tehnic.
* **Valorile variabilelor de intrare.** Dacă excepția este generată într-o metodă, operație cu anumite variabile de intrare, salvarea valorilor acestor variabile poate ajuta.
* **Statutul conexiunii.** Dacă aveți o aplicație care necesită conexiune la internet, iar conexiunea nu a fost realizată, sau a fost realizată cu erori acest lucru poate rezulta în excepție.
* **Nivelul de importanță.** Dacă aveți nevoie să structurați fișierul de log în dependeță de tipul mesajelor, puteți adăuga un string în fața acestora(ex. MesajInfo, Warning, Exception etc.) și să le atribuiți în dependență de caz.
* **Informații despre dispozitiv.** În unele cazuri avem nevoie să știm ce resurse are dispozitivul care rulează aplicația.