

## 3F Házi feladat

### Felügyeletre tervezés

**FIGYELEM:** A házi feladat megoldása előtt olvassa el a tárgy weblapján lévő HF tudnivalókat! A házi feladat leadása előtt nézze végig a HF tudnivalóknál szereplő ellenőrző listát!

### Környezet

A feladat során egy egyszerű, C# nyelven írt RSS-hírcsatornák tartalmát aggregáló szerveralkalmazást kell kiegészíteni<sup>1</sup>. (Az alkalmazás futtatásához legalább 2.0-ás .NET Framework, módosításához Microsoft Visual Studio 2010 vagy Microsoft Visual C# 2010 Express Edition eszköz ajánlott<sup>2</sup>.)

A program periodikusan meglátogatja a feeds.xml fájlban megadott hírcsatornákat, ezek tartalmát letölti. Amennyiben egy HTTP-kérés érkezik az alkalmazáshoz, úgy válaszul elküld egy HTML-oldalt, amelyet a tárolt hírekből és a programhoz mellékelt sablonfájlokból állít elő.

Egy ilyen kérés eredménye látható az alábbi ábrán:



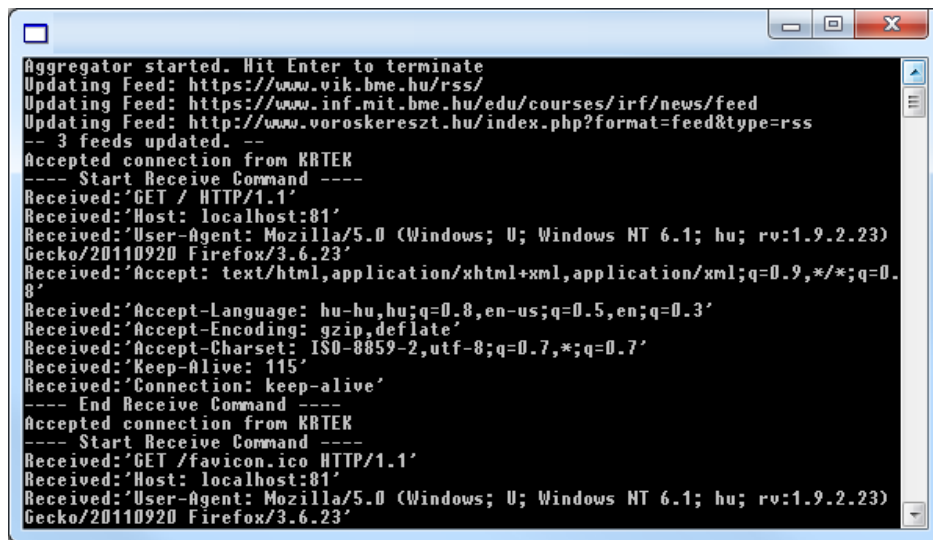
<sup>1</sup> A program alapjául Stephan Meyn és smallguy78 projektje szolgált, amely a The Code Project Open License (CPOL) 1.02 alatt került publikálásra.

<sup>2</sup> Express Edition használata esetén a StyleCop eszköz nem integrálódik automatikusan a projektbe, hanem külön hozzá kell adni a projekt fájljaihoz. Bővebben: <http://stylecop.codeplex.com/wikipage?title=Running%20StyleCop%20in%20VS2005%20or%20VS%20Express>

(A program alapértelmezés szerint a 81-es porton figyel, így a `http://localhost:81` címen érhető el. A port átállítható az RssAggregator projekt konfigurációs állományában.)

Bizonyos speciális kérésekre is van lehetőség a `http://localhost:81/?reverse` vagy a `http://localhost:81/client-info` cím meglátogatásával.

A kérések során a szerveralkalmazás konzolos kimenetén láthatók a fogadott kérések tulajdonságai diagnosztikai célokból.



```
Aggregator started. Hit Enter to terminate
Updating Feed: https://www.vik.bme.hu/rss/
Updating Feed: https://www.inf.mit.bme.hu/edu/courses/irf/news/feed
Updating Feed: http://www.voroskereszt.hu/index.php?format=feed&type=rss
-- 3 feeds updated. --
Accepted connection from KRTEK
---- Start Receive Command ----
Received: 'GET / HTTP/1.1'
Received: 'Host: localhost:81'
Received: 'User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; hu; rv:1.9.2.23)
Gecko/20110920 Firefox/3.6.23'
Received: 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8'
Received: 'Accept-Language: hu-hu,hu;q=0.8,en-us;q=0.5,en;q=0.3'
Received: 'Accept-Encoding: gzip,deflate'
Received: 'Accept-Charset: ISO-8859-2,utf-8;q=0.7,*;q=0.7'
Received: 'Keep-Alive: 115'
Received: 'Connection: keep-alive'
---- End Receive Command ----
Accepted connection from KRTEK
---- Start Receive Command ----
Received: 'GET /favicon.ico HTTP/1.1'
Received: 'Host: localhost:81'
Received: 'User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; hu; rv:1.9.2.23)
Gecko/20110920 Firefox/3.6.23'
```

Részletesebb fejlesztői és üzemeltetői dokumentációt sajnos a fejlesztők „elfelejtettek” készíteni az alkalmazáshoz.

### 1. feladat: Ismerkedés az alkalmazással

Hogy jobban megértsük az alkalmazás működését, a forráskód tanulmányozásával készítsen egy UML-modellt (tipikusan komponens vagy osztálydiagramokkal), ami ábrázolja a rendszer főbb részeit és azok kapcsolatait. A modell tartalmazza, hogy melyik komponens milyen adatot tárol és mi a feladata, hogyan kommunikálnak egymással a komponenseink, stb. A modell elemeit, és így az alkalmazás felépítését és működését, egy rövid szöveges leírásban is ismertesse.

**Figyelem:** nem egy reverse engineering eszközzel előállított részletes osztálydiagramot kérünk, hanem egy saját modellt, ami csak az alkalmazás fontosabb részeit ábrázolja!

### 2. feladat: Felügyeleti modell elkészítése

Megfelelő érveléssel alátámasztva sorolja fel, hogy a kiszolgálónak milyen, a működésével kapcsolatos kvantitatív jellemzőket kellene rendszerfelügyeleti célokra elérhetővé tennie (pl. uptime)! Legalább öt metrika definiálása kötelező.

A metrikák definícióját valami jól áttekinthető formában adja meg (pl. táblázat), mely legalább a következő tulajdonságokat tartalmazza:

- metrika neve,
- metrika leírása,
- metrika számításának módja.

Ügyeljen arra, hogy a metrikák definícióját annyira formálisan és részletesen adja meg, hogy azok további pontosítás nélkül implementálhatóak legyenek (ez nem azt jelenti, hogy konkrét változóneveket kell megadni, csak annyit, hogy a számolásuk pontos mikéntjét leírjuk).

Ezen kívül adja meg és indokolja, hogy milyen távoli rendszerfelügyeleti beavatkozásokat tart szükségesnek a kiszolgálóra (legalább hármat írjon le).

### **3. feladat: Megvalósítás WMI segítségével**

WMI-instrumentáció hozzáadásával és a szükséges egyéb módosításokkal tegye lehetővé, hogy a fentiek közül három metrika WMI segítségével, távolról lekérdezhető legyen! A beavatkozások közül egyet implementáljon WMI-metódusként úgy, hogy az pl. PowerShell-ből végrehajtható legyen!

A megoldás dokumentációja a szokásos elemeken kívül (tervezési döntések, új osztályok/módosítások dokumentációja, tesztelés menete, környezet leírása, telepítés menete...) tartalmazza az 1. feladatban elkészített modell kiegészített változatát, illetve adjon példát a lekérdezésre és beavatkozásra (praktikusan PowerShell parancs formájában)!

### **Szerkezeti és elnevezési konvenciók**

A következők betartása **kötelező**, egyéb esetben a házi feladatot nem fogadjuk el.

Mivel a monitorozandó alkalmazásunk nem fut folyamatosan, és egyszerre több szerver példány is futhat ugyanazon a gépen, ezért:

- használjon lecsatolt (decoupled) hosting modellt,
- a WMI szolgáltató (provider) ne singleton legyen, hanem úgynevezett multi-instance típusú.

Kötelező továbbá a következő elnevezések használata:

- A kiszolgálóval kapcsolatos osztályok a `root\irf` névtér alá kerüljenek.
- A WMI szolgáltató neve `IRF_RssAggregator` legyen.
- A WMI szolgáltatót megvalósító osztály neve `RssAggregatorProvider` legyen.

A feladat megoldása során a szerver komponens futhat rendszergazdai jogosultságokkal rendelkező felhasználó nevében.

### **További követelmények**

- A feladat során kötelező használni a StyleCop<sup>3</sup> eszközt. A leadott megoldását ellenőrizze vele, és kellőképp megalapozott esetektől eltekintve – a kiadott programban szereplő konfigurációval – megoldása ne tartalmazzon hibát!
- A forráskódban ne szerepeljenek „beégetett” abszolút elérési utak.
- A megoldás során lehet, hogy módosítani kell az eredeti forráskódot. A forrásfájlok új verzióján kívül a leadott csomagba rakjon be *diff* fájlokat is (unified diff formátumban<sup>4</sup>), melyek a módosításokat jelzik.

---

<sup>3</sup> A StyleCop eszköz elérhető a <http://stylecop.codeplex.com/> címen.

***Plusz feladatok***

- A megoldás tesztelésében mutassa be azt az esetet is, amikor az alkalmazás kliens és kiszolgáló komponense külön számítógépen fut, valamint a WMI lekérdezéseket is a kiszolgáló komponenst futtató géptől különböző gépen adjuk ki.

---

<sup>4</sup> Wikipedia. Diff, Unified format, [http://en.wikipedia.org/wiki/Diff#Unified\\_format](http://en.wikipedia.org/wiki/Diff#Unified_format)