

3P-2 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, wikioldalak megtekintését és szerkesztését lehetővé tevő serveralkalmazást kell kiegészíteni. (Az alkalmazás futtatásához 4.0-ás .NET Framework, módosításához Microsoft Visual Studio 2010 vagy Microsoft Visual C# 2010 Express Edition eszköz ajánlott¹.)

Egy wikioldal a `http://localhost:81/wiki/<wikioldal neve>` címen tekinthető meg (1. ábra).



1. ábra A `http://localhost:81/wiki/sorompó` oldal lekérdezésére kapott válasz egy részlete

A wikioldalak szerkeszthetők is, a szerkesztő a `http://localhost:81/edit/<wikioldal neve>` címen érhető el (2. ábra).

¹ Express Edition használata esetén a StyleCop eszköz nem integrálódik automatikusan a projektbe, hanem külön hozzá kell adni a projektfájlokhoz. Bővebben: <http://stylecop.codeplex.com/wikipage?title=Running%20StyleCop%20in%20VS2005%20or%20VS%20Express>



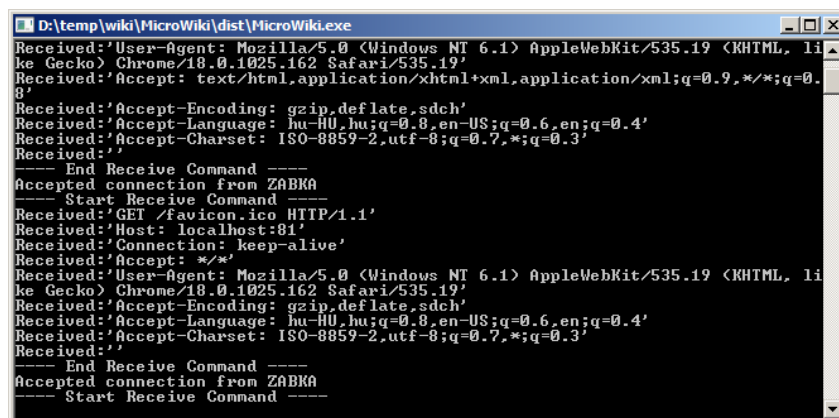
2. ábra A `http://localhost:81/edit/sorompó` oldal lekérdezésére kapott válasz egy részlete

Lehetőség van a wikilapok szövegének bizonyos formázására is:

- A következő szöveg a megjelenítés során *félkövér* lesz: `*szöveg*`
- A következő szöveg a megjelenítés során *dőlt* lesz: `_szöveg_`
- A következő szöveg a megjelenítés során *címsor* lesz: `==szöveg==`
- A következő szöveg a megjelenítés során *hivatkozás* lesz: `[szöveg]`
(A `[teszt]` a `teszt` nevű wikilapra fog hivatkozni, a `[http://www.bme.hu]` pedig a BME honlapjára fog mutatni.)

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

A kérések során a szerveralkalmazás konzolos kimenetén láthatók a fogadott kérések tulajdonságai diagnosztikai célokból (3. ábra).



3. ábra Az alkalmazás diagnosztikai kimenete

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 készítsünk egy UML-modellt (tipikusan komponens vagy osztálydiagramokkal), ami ábrázolja a rendszer főbb részeit, külső függőségeit 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 ismertessük.

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 soroljuk fel, hogy a kiszolgálónak milyen, a működésével kapcsolatos állapot és konfigurációs jellemzőket kellene rendszerfelügyeleti célokra elérhetővé tennie (pl. portszám)! Legalább öt jellemző definiálása kötelező. (Itt most ne teljesítmény jellegű metrikákat adjunk meg.)

A jellemzők definícióját jól áttekinthető formában adja meg (pl. táblázat), amely legalább a következő tulajdonságokat tartalmazza:

- név,
- leírás,
- írható / olvasható,
- lehetséges értékkészlet.

Ügyeljünk arra, hogy a jellemzők definícióját annyira részletesen adjuk meg, hogy azok további pontosítás nélkül implementálhatóak legyenek (ez nem feltételen azt jelenti, hogy konkrét változóneveket kell megadni, csak annyit, hogy leírásuk minél pontosabb legyen).

Ezen kívül adjuk meg és indokoljuk, hogy milyen távoli rendszerfelügyeleti beavatkozásokat érdemes megvalósítani az alkalmazáshoz (legalább hármat írjunk le).

3. feladat: Megvalósítás WMI segítségével

WMI-instrumentáció hozzáadásával és a szükséges egyéb módosításokkal tegyük lehetővé, hogy a fentiek közül három metrika WMI segítségével, távolról lekérdezhető legyen! A beavatkozások közül egyet implementáljunk WMI-metódusként úgy, hogy az pl. PowerShellből 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 a tesztelés dokumentálása során adjunk 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_MicroWiki` legyen.
- A WMI szolgáltatót megvalósító osztály neve `MicroWikiProvider` legyen.

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

További követelmények

- A feladat során kötelező használni a StyleCop² eszközt. A leadott megoldását ellenőrizzük vele, és kellőképp megalapozott esetektől eltekintve – a kiadott programban szereplő konfigurációval – a megoldás ne tartalmazzon figyelmeztetést!
- A forráskódban ne szerepeljenek „beégetett” abszolút elérési utak.
- Ne módosítsa a program parancssori paraméterezését (azaz ne vezessen be parancssori paramétereket).
- A megoldás során lehet, hogy módosítani kell az eredeti forráskódot. A forrásfájlok új verzióján kívül a leadott csomagba rakjunk be *diff* fájlokat is (unified diff formátumban³), minden módosított fájlhoz egy-egy külön diff fájl tartozzon. Diff fájlokat Linux alatt a `diff -u eredeti_fajl modositott_fajl` paranccsal tudunk létrehozni, Windows alatt pl. a GnuWin⁴ tartalmazza a diff parancsot.
- A leadott tömörített fájlban szerepeljen a dokumentáció (forrásfájlok és PDF export egyaránt), a diff fájlok, valamint a teljes solution (nem csak a .cs fájlok) fordítható állapotban.

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.

² A StyleCop eszköz elérhető a <http://stylecop.codeplex.com/> címen.

³ Wikipedia. Diff, Unified format, http://en.wikipedia.org/wiki/Diff#Unified_format

⁴ GnuWin, <http://gnuwin32.sourceforge.net/packages.html>