

Bloglines Webszolgáltatások

Egyre több weblap ajánlja fel szolgáltatásainak gépbarát verzióját. Az ilyen egyszerű de hasznos szolgáltatásra jó példa az új weblap tartalmakról küldött frissítés.

Az elmúlt hónapban azokat a módszereket vizsgáltuk amelyekkel begyűjthetjük vagy összerendezhetjük a különböző weblapok tartalmát majd a nap híreiből egyetlen összefoglalót készíthetünk. Bár izgalmas volt látni, mire jutunk mindössze pár sornyi kóddal, az általam bemutatott alkalmazás egyszerű kis játék a tényleges összesítőkhöz képest. A mintaalkalmazás egyetlen felhasználót kezel, primitív beállítássalományon alapszik, nem csoportosítja a blogokat, csak akkor frissíti a weblogokat, amikor kifejezetten utasítjuk rá. Ezen kívül nem figyel és kezeli a hibákat sem.

Figyelembé véve a szükséges technikai és tervezési kérdéseket egy igazán felhasználóbarát, komoly gyűjtőprogram megírása már meghaladná cikkünk kereteit. Azonban néhány nappal az előtt, hogy leültem volna megírni ezt a cikket, valami lenyűgöző dolog történt. Az ingyenes *Bloglines.com* gyűjtőszolgáltatás, amit igen sok ember használ saját kedvenc weblogjainak összegyűjtésére, elérhetővé tette webszolgáltatás-programozási felületét, melynek segítségével független fejlesztők használhatják a *Bloglines* adatait és alkalmazásait saját alkalmazásaik létrehozásához és telepítéséhez. A *Bloglines API* megjelenése jól mutatja a webszolgáltatások növekvő népszerűségét az nagyobb honlapok között és ajtót nyit az új *Bloglines* infrastruktúrára épülő új alkalmazások előtt.

E hónapban a *Bloglines API*-t vesszük szemügyre majd létrehozunk egy rá épülő egyszerű alkalmazást. Az *API* a cikk írásakor (2004 október eleje) teljesen újdonságnak számított és nyilvánvalóan fejlődni fog ahogy egyre több ember használja. Azok számára akiket érdekelnek a weblogok és eddig csak a webszolgáltatások praktikus felhasználási lehetőségeire vártak, az események ilyen fordulata éppen időben történt.

Mi is az a webszolgáltatás?

A webszolgáltatások mögött rejlő alapötlet nagyon egyszerű: A web sikerének titka nem kis részben annak köszönhető, hogy az ügyfél és a kiszolgáló operációs rendszere nem lényeges. Tehát amíg az ügyfél és a kiszolgáló is tartja magát, a *HTTP* és *HTML* szabályokhoz zökkenőmentesen tudnak beszélgetni egymással. A *Linux* is éppen ez által volt képes helyet biztosítani magának a kiszolgáló piacon.

A webszolgáltatások még egy lépéssel még tovább mennek, és azt mondják, hogy a web legnagyobb használói nem az

emberek hanem a számítógépek. Bár a számítógépek *HTTP*-n keresztül kommunikálnak, az adatokat *XML jelölőnyelven (markup language)* vagy metanyelven küldik és fogadják, amely robbanásszerűen terjed az utóbbi években. Ha az én számítógémem képes *XML*-t küldeni a számítógépnek küldött *HTTP* kérelemben, a géped pedig *XML*-t ad vissza a *HTTP* válaszában, operációs rendszertől és nyelvtől függetlenül tudunk információt cserélni.

A szolgáltatás *XML-RPC* néven ismert eredeti formája még ma is létezik és kiválóan használható a gyors könnyű kapcsolattartásra. Az ötletet azóta továbbfejlesztették és rengeteg fajta adattípus, hibakezelő módszer és objektum sorbafejtő megoldás jelent meg, amelyek az *XML-RPC* nyelvből hiányoztak. Ez a kiterjesztést *SOAP*-nak (*Simple Object Access Protocol*) nevezték el. A *SOAP* elméletileg sokféle protokoll felett képes futni, de a legtöbbször *HTTP* felett küldik.

A *SOAP* sok problémára kiváló megoldás, eltekintve attól, hogy iszonyú bonyolult, elég lassú és nehéz megvalósítani. Az *XML-RPC* és a *SOAP* egyaránt megköveteli, hogy a *HTTP* üzenetben a lekérdezést hibátlan *XML* alakban adjuk meg. A növekvő összetettségre válaszul született meg a *REST (representational state transfer)*, ahol minden műveletet egyszerű *HTTP GET* kérelmek indítanak és az összes paramétert maga az *URL* tartalmazza. Válaszként a kérelemnek megfelelő bejegyzéseket és mezőket tartalmazó *XML* dokumentumot kapjuk vissza. A *Bloglines API* valamennyi hívását *REST*-en keresztül végezzük, bár nehéz megmondani, hogy a fejlesztők a viszonylag egyszerű lekérdezések vagy a tervezési irányelvek miatt döntöttek így. Bár a webszolgáltatások nem sokáig maradnak a cégek zárt ajtó mögött, csak néhány nagyobb webhely tette nyilvánossá terveit és *API* felületét. A legismertebb példák a web legnagyobb és legjövődélmezőbb vállalkozásai az *Amazon*, az *eBay* és a *Google*. Az *eBay* éves díj és tranzakciónkénti díj formájában pénzt kér webszolgáltatásainak eléréséért. Ezzel ellentétben az *Amazon* és a *Google* ingyenesen elérhetővé tették felületüket, igaz felhasználhatósági megkötésekkel és a későbbi elérhetőség ígérete nélkül.

Az *API* nyilvánossá tételével a *Bloglines* is jelzi, hogy az *Amazon*, *Google* és *eBay* által létrehozott fejlesztői közösséghez hasonló tábort szeretne felállítani. Lépésével egyúttal azt is jelzi, hogy továbbra is szeretne a világ első webloggyűjtő és alkalmazás rendszere maradni. Tekintve, hogy

a Google megvásárolta a *Blogger*-t néhány évvel ezelőtt, valamint ha megnézzük, milyen széleskörű keresési megoldásokat kínál *API* felületén a *Bloglines*, lehet hogy egy új alkalmazás csatának lehetünk szemtanúi a figyelemért versengő *Google* és a *Bloglines* felület között.

A Bloglines API

A *Bloglines* nagy számú weblogból és gyakran frissített hírforrásokból gyűjt adatot. A *Bloglines* több különféle formátumban is szívesen fogad adatokat, ideértve az *Atom* és az *RSS* néhány verzióját. Sőt, a *Bloglines* megkérdezi feliratkozótól, hogy melyik típust szeretnék használni amennyiben egynél több is elérhető. A *Bloglines* ezután archiválja a tartalmat, keresőfelületet biztosítva az érdeklődőknek. A *Bloglinesban* néhány fontossági képességet is találunk, amelyek felhívják a feliratkozók figyelmét rá, hogy mely további weblogok érdekelhetik őket. Végül a *Bloglines* lehetővé teszi, hogy más felhasználók feliratkozásába is beleskussunk; ha valaki kíváncsi rá, mely weblogok érdekelnek engem, elolvashatja a adatlapomat és megtekintheti a feliratkozásaimat. Egyelőre legalábbis, ezen funkciók nagy része még nincs leplezve így csak a *Bloglines* weblap felületén keresztül érhetők el. Három képesség azonban már a *Bloglines Webszolgáltatások API*-ból is elérhető:

- **Figyelemfelhívás (Notifier):** ha *Bloglines* feliratkozóként tudni szeretnénk, mikor érkezett új hír egy vagy több kiválasztott weblogunkra, ezzel a képességgel meg tudhatjuk. Ez a *Bloglines Webszolgáltatások* legmegalapotottabb része, számtalan eszköz több különféle

operációs rendszer és ablakozó eszközkészlet alá áll rendelkezésünkre amelyek ezt a felületet használják a frissítésekhöz.

- **Szinkronizálás (Sync) API:** ezzel egy adott felhasználó feliratkozásairól gyűjthetünk információkat, a feliratkozások legfrissebb változataival egyetemben. Ezt úgy képzelhetjük el, mint a *Bloglines* által létrehozott fő weblog lista *HTML* kódja mögötti található adatokat.
- **Blogroll API:** módszer, amellyel a lekérhetjük és megjeleníthetjük egy adott felhasználó feliratkozási listáját.

Figyelemfelhívó API

Mint korábban írtam, a *Bloglines* a *REST* rendszerét használja az összes webkiszolgáló *API*-jához. Ennek megfelelően minden lekérés egyetlen *URL*-ből áll, ahol az összes paraméter és azok értékei az *URL*-be kódolva találhatók.

A visszakapott információ olyan alakú, amelyet a kiszolgáló megfelelően talál. Ez szöges ellentétben áll a *SOAP* megközelítésével, ahol minden egyes paraméter és visszaadott érték nevét és típusát előre meg kell határoznunk. A szabály alól egy kisebb kivétel találmunk; amikor ugyanis az *API* azonosítás céljából felhasználónévvel és jelszóval kér, az alap *HTTP*-n keresztül és nem az *URL*-ben érkezik. A *Bloglines* világában, a feliratkozókat e-mail címük és felhasználó által választható jelszavuk azonosítja.

A legkönnyebben megérthető és használható *API* a *Notifier*. A *Notifier* meghívásához egyszerűen csak nyissuk meg

© Kiskapu Kft. Minden jog fenntartva

Kapu a Linux világába

- cikkek
- hírek
- fórum
- címtár

Több mint 1000 ingyenesen letölthető cikk!

The screenshot shows the Linuxvilag.hu website interface. At the top, there's a search bar and navigation links like 'Nyitó', 'Hírek', 'Magazin', 'Címtár', 'Fórum', 'Súgó', 'Médiaajánlat', 'E-mail'. Below the search bar, there's a main article titled 'Szavazz a CD-mellékletről!' with a sub-header 'Tavaszzal, Szerkeszd te is a Linuxvilagot!'. The article text discusses a survey about the CD-melléklet (CD supplement) and asks for user feedback. To the right of the article, there's a sidebar with a 'Bejelentkezés' (Login) section, a 'Szavazás' (Vote) section, and a 'MEGJELENT!' (Subscribe) section. The bottom of the page shows a 'Híreink:' (Our News) section with a link to 'München mégis vár az átállással'.

www.linuxvilag.hu

a `rpc.bloglines.com/update?user=reuven@lerner.co.il&ver=1` URL-t. A válasz, amelyet a kiszolgáló (helytelenül) `text/html` MIME típusként azonosít, a következő formátumú sima szöveges állományt tartalmazza:

```
|A|B|
```

A figyelemfelhívók a következőképpen értelmezhetik a választ:

- Az A általában a felhasználó feliratkozásában található olvasatlan weblog-bejegyzések számát mutatja.
- Amennyiben a megadott email cím nincs regisztrálva, az A értéke -1.
- Amennyiben B nem üres, a frissítő lap címére mutat. A dokumentáció nem túl bőbeszédű a frissítő lappal kapcsolatban. Feltételezem ez a lap inkább az embereknek mint programoknak szól, hiszen lehetetlen, de legalábbis nagyon nehéz lenne beazonosítani az összes *Notifier API*-t használó programot, amelyeknek frissítésre lehet szüksége.

Egy modern magas szintű nyelven nagyon könnyen fel tudjuk használni a *Notifier API*-t. Az írás születésekor *Bloglines* ügyfélkönyvtárak állnak rendelkezésre *Perl*, *Python* és *Ruby* nyelveken. Én a *Perl* változatot használok (a CPAN-on *WebService::Bloglines* található), de előfordulhat, hogy szívesebben használunk saját verziót, egy másik verziót, esetleg mindkettőt.

Nézzünk egy egyszerű parancssoros programot, amely kiírja, hogy „Új blog érkezett!” ha a *Bloglines* szerint új üzenetek várakoznak, vagy „Nincs új blog” ha már mindent elolvastunk:

```
#!/usr/bin/perl
use Webservice::Bloglines;
my $username = 'reuven@lerner.co.il';
my $password = 'MYPASS';
my $bloglines = Webservice::Bloglines->new(
    username => $username,
    password => $password);
my $unread_blogs = $bloglines->notify();
if ($unread_blogs)
{
    print "'$unread_blogs' új blog üzenete van!\n ";
}
else
{
    print "Nincs új blog.\n"
}
```

A `$bloglines->notify()` érték az olvasatlan küldemények számát tartalmazza és a nem az olvasatlan weblogok számát. Amennyiben 15 olvasatlan üzenetünk van 5 különféle weblogban, a `$bloglines->notify()` 15-öt és nem 5-öt ad vissza. Továbbá, a belső *Bloglines* adatbázis állapotát tükrözi. Azaz, ha a *Keep New* dobozra bökünk a weblog bejegyzés végén, ez is bekerül a `$bloglines->notify()` számlálóba. Amennyiben hibás e-mail címet adunk meg, programunk végzetes hibával tér vissza, jelezve, hogy hibás felhasználónevet adtunk meg. Hibás jelszó megadásnak nincsenek következményei a *Notifier API*-ban, hiszen ez az információ nyilvánosan elérhető.

Blogroll API

A *Bloglines* másik eszköze, mint korábban említettük, a *Blogroll API*. A *blogroll* lényegében egy webloglista amit az adott szerző érdekesnek talál és gyakran olvas. Ha érdekesnek találjuk valakinek a weblogját olvasgatni, elképzelhető, hogy kíváncsiak vagyunk olvasmánylistájára is. A *Bloglines* esetében a blogroll egyszerűen a felhasználóhoz rendelt feliratkozások listája. Mostanáig azt állítottuk, hogy a *Bloglines* felhasználói név megegyezik felhasználó e-mail címével. Ez azonban nem teljesen igaz. Amennyiben a *Bloglines*-t csak saját magáncéljainkra használjuk, másokkal soha nem osztjuk meg a feliratkozásainkat, az e-mail címünkön kívül tényleg nincs szükségünk másra. Amennyiben azonban közzé szeretnénk tenni feliratkozásainkat, választanunk kell egy felhasználói nevet, amelyre hivatkozni lehet. Például az én regisztrációs e-mail címem `reuven@lerner.co.il` míg a felhasználói nevem reuven. Ez a megkülönböztetés az első néhány hónapnyi *Bloglines* használat során nem igazán volt világos számomra, igaz, úgy tűnik most már valamivel jobban reklámozzák a dolgot.

Amennyiben a felhasználó már szerzett magának egy felhasználói nevet valamint megosztotta feliratkozásait, a következő *HTML* és *JavaScript* felületen keresztül kérhetjük le a felhasználó blogrollját: `http://www.bloglines.com/public/reuven`. Ha a blogroll eredményét inkább *HTML* formátumban szeretnénk megnézni, azt a következő alakú URL segítségével tehetjük meg: `http://rpc.bloglines.com/blogroll?id=reuven&html=1`.

Ugyanakkor a webszolgáltatások alapötlete éppen az, hogy az adatokat gépek számára olvashatóvá tegyük és ezáltal a számítógépek tárolhassák és feldolgozhassák. A *Bloglines* feliratkozás listáinak exportálásához *Dave Winer* 2000-ben készített *OPML (Outline Processor Markup Language)* formátumát használja. Ez ugyan nem része a *Bloglines Webszolgáltatások* szabályainak, de a következő URL-ről könnyen begyűjthetjük: `http://www.bloglines.com/export?id=reuven`.

Valamennyi fenti példában a *Bloglines* felhasználónevet annak a felhasználónak a nevére kell lecserélnünk akinek a blogjegyzékére kíváncsiak vagyunk. Nem mindenkit teszi nyilvánossá a blogjegyzékét, így aztán letöltéskor könnyen találkozhatunk hibáüzenetekkel. Az *OPML*-t letöltés után fel kell dolgoznunk valamilyen eszközzel, például a *CPAN XML::OPML* moduljával.

Összefoglalás

Mint láthatjuk, a webszolgáltatásokhoz szánt *Bloglines API* szabad utat enged a harmadik féltől származó alkalmazásoknak. Egyre nagyobb az esély olyan alkalmazások létrehozására, amelyek ugyan *HTML*, *XML* és *HTTP* nyelveket használnak mégsem kötődnek webböngészőhöz. A *Notifier* és *Blogroll API* megjelenése csak a kezdet. Mint korábban láthattuk a *Sync API* lehetővé teszi, hogy a fejlesztők hatékonyan készítsenek új *GUI* felületet és alkalmazásokat a *Bloglines* által tárolt és begyűjtött tartalomhoz. Következő cikkben bepillantunk a *Sync API*-ba, és egy egyszerű alkalmazást építünk a *Bloglines* alapokra.

Linux Journal 2005. január, 129. szám

Reuven M. Lerner