Cikkgyűjtés RSS használatával

A hírgyűjtő rendszerek nem csak weblogok figyelemmel kísérésére alkalmasak. Az RSS segítségével a látogatókat könnyen tájékoztathatjuk egy-egy webhely vagy alkalmazás változásairól is.

Amikor elkezdtem használni a webet, még az volt a szokás, hogy minden új weboldal fenntartója küldött egy levelet ***Tim Berners-Leenek***, amiben megadta az ***URL***-t és az oldal témájának rövid leírását. ***Tim*** válaszában röviden összefoglalta személyes megjegyzéseit, majd frissítette a weboldalakat tartalmazó főlistát, amit bárki szabadon megnézhetett. A webes közösség aktív résztvevői rendszeresen átnézték a listát – majd utódját is, amelyet mellesleg a ***Mosaic*** böngésző készítői állítottak össze –, és kimazsolázták belőle az új vagy frissített oldalakat, nehogy lemaradjanak valami érdekesről.

A web alig egy évtized alatt túl nagyra nőtt ahhoz, hogy az új webhelyek listáját kézi munkával fenn lehetne tartani. Még ha találnánk is rá embereket, a látogatók csak egy kis töredékét tudnák befogadni a minden egyes nap nyilvánosságra kerülő új tartalomnak. Ha figyelembe vesszük, hogy napjainkban már több százezer weblog, röviden blog üzemel, és sokat közülük elég gyakran frissítenek, akkor nyilvánvalóvá válik, hogy a feladat rendkívül nehezen oldható meg.

Az egyik megoldás, hogy böngészőnkben könyvjelzőket használunk, ám ezek rendszeres végiglátogatása rendkívül vesződséges, főleg, ha naponta többször kívánjuk elvégezni. Milyen jó lenne, ha mindegyik oldal maga jelezné változásait, így csak akkor kellene meglátogatnunk őket, amikor valóban érdemes!

Az ötlet persze nem új, a tartalmukat hirdető weboldalak elképzelése már évekkel ezelőtt megszületett. Sajnos be kell vallanom, csak néhány hónappal ezelőtt jöttem rá, hogy mennyire elmaradott módszert alkalmazok, amikor könyvjelzők alapján látogatom végig a kedvenc oldalaimat. Egy ***RSS*** összesítő segítségével – vagyis egy olyan programmal, amely összegyűjti a különféle helyek ***RSS*** cikkeit, és jelzi, ha valahol frissítettek – mindezt sokkal kevesebb idő alatt el tudom végezni.

Ebben a hónapban a népszerű ***RSS*** (***Really Simple Syndication; valóban egyszerű cikkgyűjtés***) vagy ***RDF Site Summary*** (***RDF webhely-összegzés***) formátumcsaládot tárgyaljuk, megvizsgáljuk, hogy tagjai milyen célokra használhatók, illetve a szabványoknak megfelelő cikkek hogyan állíthatók elő.

Egyszerű RSS

Az ***RSS*** tulajdonképpen a ***Netscape*** gyermekének mondható – az internetes céget azóta az ***AOL*** felvásárolta, gyakorlatilag eltüntette. A ***Netscape*** ötlete az volt, hogy a felhasználóknak több forrásból származó híreket kínál egyetlen oldalon. Ebből a célból született meg az ***RSS 0.90***. Aki a ***Netscape*** portálján keresztül híreket szeretett volna közzétenni, ***RSS*** segítségével tehette ezt meg. A ***Netscape*** rendszere lekérte a megfelelő ***RSS*** dokumentumot a kérdéses webhelyről, majd közzétette a kapott anyagot.

Bár az ***RSS 0.90*** kisebb forradalmat indított el, rendkívül bonyolult volt. ***Dave Winer***, a ***Userland Software*** későbbi vezetője az ***RSS***-t sokkal egyszerűbb szabálygyűjteménnyé dolgozta át, nevét ***RSS 0.91***-re változtatta, majd elkezdte saját weblogjában, a ***scripting.com*** oldalon népszerűsíteni.

Az ***RSS 0.91*** szinte azonnal megjelent a web összes szegletében, és ***Dave*** narancsszínű ***XML*** gombjai, amelyek az egyes helyek ***RSS***-képességét jelezték, hatalmas népszerűségre tettek szert. Néhány év alatt további ***RSS***-változatok is megjelentek. Az ***RSS 1.0*** fejlesztését egy webes csoportosulás végezte, míg az ***RSS 2.0*** – ***Dave*** vezényletével – a ***0.9x*** továbbfejlesztéseként jelent meg.

Aki jól figyelt, észrevette, hogy jelenleg három különböző, ám egyaránt ***RSS***-nek nevezett cikkgyűjtő formátum is létezik. Bár vannak közöttük hasonlóságok, a különféle változatok erősen eltérő formátumokat adnak meg.

Az ***RSS*** sok tekintetben hasonlít a ***HTML***-re és a ***HTTP***-re, és eleinte egy kisebb csoport által fejlesztett, könnyen megérthető és könnyen megvalósítható szabványról volt szó. Az elmúlt évek során azonban mindhárom szabvány hatalmas fejlődésen ment keresztül, ami rugalmasságuk és egyszerűségük részleges elvesztéséhez vezetett.

A legegyszerűbb – és nem kevéssé népszerű – változat az ***RSS 0.91***. Minden tartalmat `<rss>` címkék közé kell helyezni, ez tartalmazza a változat megjelölését, és egy darab `<channel>` (csatorna) elem szerepelhet benne. A kötelező címkéket `title`, `link`, `description`, `language` és `image` (rendre cím, hivatkozás, leírás, nyelv és kép) egy vagy több `<item>` (tétel) elem követi. Minden tétel saját címmel, hivatkozással és leírással rendelkezik. Példaként lássunk egy egyszerű ***RSS***-cikket saját weblogomból (*1. lista*).

1. lista

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE rss PUBLIC
"-//Netscape Communications//DTD RSS 0.91//EN"
"http://my.netscape.com/publish/formats/rss-
0.91.dtd">

<rss version="0.91">

<channel>
  <title>Altneuland</title>
  <link>http://altneuland.lerner.co.il/</link>
  <description>Reuven´s Weblog</description>
  <item>
    <title>Independence Day</title>
     ➥<link>http://altneuland.lerner.co.il//
     ➥40</link>
  </item>
  <item>
    <title>Linux desktops for the masses?
     ➥Ha!</title>
     ➥<link>http://altneuland.lerner.co.il//
     ➥39</link>
  </item>
</channel>
</rss>
```

Ha megvizsgáljuk a fenti *RSS*-cikket, láthatjuk, hogy a korábban említett ***RSS 0.91*** szabálygyűjtemény előírásainak nem felel meg, ugyanis hiányzik belőle a kötelező nyelv és kép elem, illetve nem mindegyik tételhez tartozik leírás. Rutinosabbaknak ez aligha okoz meglepetést, régebben a *HTML* esetében is láthattunk ilyesmit: a programok készítői megelégednek olyan kimenet előállításával, amely hiányos ugyan, ám az alkalmazási területek túlnyomó részén megállja a helyét. Ezt a divatot követi a ***COREBlog*** is (jelenleg ezt használom saját weblogom készítéséhez), segítségével használható, ám az ***RSS 0.91***-nek csak részben megfelelő cikkeket állíthatunk elő.

A cikkek létrehozása

Ha szabványos *RSS*-cikket szeretnénk előállítani, próbálkozzunk meg a népszerűbb nyelvek mindegyikéhez elérhető nyílt forrású modulok valamelyikével. A Perl-hívők számára például az `XML::RSS` modul jelent segítséget, ezt bármelyik ***CPAN*** tükörről le lehet tölteni (lásd az internetes források részt).
Ha a modul segítségével *RSS*-cikket szeretnénk létrehozni, az alábbihoz hasonló egyszerű programot kell írnunk.

```
#!/usr/bin/perl

use strict;
use diagnostics;
use warnings;

use XML::RSS;

my $url = "http://altneuland.lerner.co.il/";

my $rss = new XML::RSS (version => ´0.91´);
$rss->channel(title    => ´Altneuland´,
       link            => $url,
       language        => ´en´,
       description     => "Reuven Lerner´s
                          ➥Weblog");

$rss->add_item(title => ´Being scared´,
        link         => "$url/43/index_html",
        description  => ´Blog entry´
       );

print $rss->as_string;
```

A program első lépéseként egy új `XML::RSS` objektumot hozunk létre, egyúttal azt is kinyilvánítjuk, hogy az ***RSS*** szabvány ***0.91***-es változatát kívánjuk használni. Ezután az egyedi tételeket adjuk meg. Az `image` címkét elhagyhatjuk. Bár az `XML::RSS` modul a csatorna leírásából bármelyik címke elhagyását lehetővé teszi, például a cím és a hivatkozás kihagyásával az egész cikk értelmét veszti.
Következő lépésünk a csatorna feltöltése az egyes tételekkel. Amikor ezzel végeztünk, készen állunk az *RSS* kimenet előállítására, amely a következőképpen fog alakulni:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE rss PUBLIC
"-//Netscape Communications//DTD RSS 0.91//EN"
"http://my.netscape.com/publish/formats/rss-
0.91.dtd">

<rss version="0.91">

<channel>
<title>Altneuland</title>
<link>http://altneuland.lerner.co.il/</link>
<description>Reuven Lerner´s Weblog </description>
<language>en</language>

<item>
<title>Being scared</title>
<link>http://altneuland.lerner.co.il/43/index_html
➥</link>
<description>Blog entry</description>
</item>

</channel>
</rss>
```

Az *RSS*-cikkek előállítására szolgáló programok nagy része nem fogja ismételten meghívni az `$rss->add_item()` függvényt, ahogy én tettem. Ha weblog, kereskedelmi hírlap vagy egyéb gyakran frissített oldal változásait szeretnénk jelezni, akkor célszerűbb egy olyan *RSS*-cikket készítenünk, amely egy könyvtár fájljain vagy – még jobb megoldás – egy relációs adatbázis sorain halad végig újra és újra.

Az alábbi kódrészlet például az utolsó 24 órában megjelent weblog bejegyzéseket gyűjti össze egy *PostgreSQL* alatti, `weblog_entries` nevű táblába.

```
# Az összes az utolsó 24 órában készült bejegyzés
# kigyűjtése
my $sql = "SELECT entry_id, title, link,
description
          FROM weblog_entries
          WHERE when_entered >= (NOW() - interval
                              ➥'1 day')";

# Az SQL-utasítás összeállítása
my $sth = $dbh->prepare($sql);

# Az SQL-utasítás végrehajtása
my $result = $sth->execute;

# Végiglépkedés a kapott sorokon
while (my $rowref = $sth->fetchrow_arrayref)
{
my ($id, $title, $link, $description) = @$rowref;

$rss->add_item(title => $title,
            link => $link,
            description => $description
           );
}
```

A fentiekből azonnal nyilvánvalóvá válik az is, hogy miért érdemes relációs adatbázisban tárolni a weblogokat. Ha a bejegyzések bekerülnek valamilyen adatbázisba, az új szolgáltatások, például a cikkgyűjtés megvalósítása már egyszerű. Noha az `XML::RSS` biztosít lehetőséget a begyűjtött cikkek számának korlátozására (erre példakódot is találunk), erre a feladatra az adatbázisok sokkal alkalmasabbaknak tűnnek, hiszen esetükben egyszerűen a `LIMIT` módosítóval be tudjuk állítani a kapott sorok számának felső határát.

Áttérés az RSS 1.0 változatra

Az ***RSS 1.0*** egyfajta válasz volt az ***RSS 0.91***-re, célja a közelítés volt a ***World Wide Web Consortium (W3C)*** különféle szabványaihoz, többek közt az ***RDF***-hez. A változatszám alapján azt hihetnénk, hogy az ***1.0*** a ***0.91*** frissítése, ám a jelölés rendkívül szerencsétlen, hiszen két teljesen független megoldásról van szó. A ***0.91*** (és utódja, az ***RSS 2.0***) fejlesztését a közösség visszajelzései alapján ***Dave Winer*** végezte, az ***1.0*** változat viszont fejlesztők egy nyitott közösségének munkája nyomán állt elő. Az ***RSS 0.91*** és ***2.0*** között több hasonlóságot fedezhetünk fel, mint az ***1.0*** és a másik két változat bármelyike között, ami nem meglepő módon számos félreértés forrása.

Az ***RDF*** (***Resource Development Framework***, erőforrás-fejlesztési keretrendszer) a ***W3C*** fejlesztése, a szemantikai, jelentéstani web létrehozására irányuló tervezet része, amelynek célja az, hogy a webet az embereken túl a számítógépek számára is érthetővé tegye. Ehhez alapfeltétel a metaadatok, vagyis a webhelyek által átadott anyagokat kísérő, a felhasználók számára láthatatlan leírók szabványosítása. Az ***RDF*** is egy próbálkozás erre a szabványosításra. Az ***RSS 1.0*** tehát a cikkgyűjtést az ***RDF***-hez csatolja, gondoskodva az ***XML*** névterek használatáról is. Az ***XML*** névterek segítségével különböző ***XML***-megadásokat is össze tudunk fogni egyetlen dokumentumba.

Ha az ***RSS 1.0***-nak megfelelő cikket szeretnénk összeállítani, akkor a fenti programban egyetlen apró módosítást kell végrehajtanunk, mégpedig át kell írnunk az `XML::RSS` esetében megjelölt változatszámot. Lásd az *1. listát*.

2. lista

```
my $rss = new XML::RSS (version => '1.00');
A módosítást követően a létrejövő RSS-cikk picit
eltérően épül majd fel.
<?xml version="1.0" encoding="UTF-8"?>

<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/
          ➥22-rdf-syntax-ns#"
 xmlns="http://purl.org/rss/1.0/"

 xmlns:taxo="http://purl.org/rss/1.0/modules/
          ➥taxonomy/"
 xmlns:dc="http://purl.org/dc/elements/1.1/"

 xmlns:syn="http://purl.org/rss/1.0/modules/
          ➥syndication/"
 xmlns:admin="http://webns.net/mvcb/"
>

<channel
rdf:about="http://altneuland.lerner.co.il/">
<title>Altneuland</title>
<link>http://altneuland.lerner.co.il/</link>
<description>Reuven Lerner's Weblog
</description>
<dc:language>en</dc:language>
<items>
 <rdf:Seq>
  <rdf:li rdf:resource=
     "http://altneuland.lerner.co.il/
     ➥43/index_html" />
 </rdf:Seq>
</items>
</channel>

<item rdf:about=
 "http://altneuland.lerner.co.il/43/index_html">
<title>Being scared</title>
<link>http://altneuland.lerner.co.il/43/
      ➥index_html</link>
<description>Blog entry</description>
</item>

</rdf:RDF>
```

3. lista

```
<?xml version="1.0" encoding="UTF-8"?>

<rss version="2.0"
 xmlns:blogChannel="http://backend.userland.com/
                    ➥blogChannelModule">

<channel>
<title>Altneuland</title>
<link>http://altneuland.lerner.co.il/</link>
<description>Reuven Lerner´s Weblog
</description>
<language>en</language>

<item>
<title>Being scared</title>
<link>http://altneuland.lerner.co.il/43/
      ➥index_html</link>
<description>Blog entry</description>
</item>

</channel>
</rss>
```

A kimenetben több említésre méltó elem is található. Érdemes például észrevenni, hogy benne számos névteret adunk meg és használunk, ezek bevezetése az `xmlns` jellemzőkkel történik, de további, kifejezetten az *RDF*-hez kötődő jellemzőket is használunk, mint az `rdf:about` és az `rdf:resource`.
A fentiekben az ***RSS 1.0*** által kínált lehetőségeket lényegében kihasználatlanul hagyjuk, hiszen a szabvány nagy számú beállítás megadására kínál módot. Például, az `$rss->channel()` híváshoz egy `syn` részt adva beállíthatjuk oldalunk cikkgyűjtési frissítésének gyakoriságát.
Az ***RSS 1.0*** a ***Dublin Core***-t is támogatja, ez egy folyamatosan növekvő népszerűségű szabványos megoldás a dokumentumok címkézésére.

RSS 2.0

A jó hír az, hogy – amint láttuk – az ***RSS 1.0*** formátumú cikkek előállítása vagy feldolgozása nem különösebben nehezebb, mint az ***RSS 0.91*** formátumúaké, feltéve persze, hogy naprakész eszközökkel rendelkezünk. Csakhogy az ***RSS 1.0*** viszonylag bonyolult, néhányan úgy vélik, hogy túlzottan is.
Mivel a számos próbálkozás ellenére nem sikerült megegyezésre jutni az ***RSS 1.0*** tekintetében, fejlesztők egy csoportja Atom név alatt új tervezetet indított. Az Atomról most nem szólnék, azt viszont érdemes tudni róla, hogy felbukkanása ösztönözte a ***Winer*** által vezetett ***RSS*** tábort az ***RSS 2.0*** kifejlesztésére.
Ha ***RSS 2.0***-megfelelő cikkeket szeretnénk előállítani, akkor újfent a kívánt változatszámot kell módosítanunk:

```
my $rss = new XML::RSS (version => ´2.0´);
```

Ügyeljünk arra, hogy a változatszám ***2.0***; nem ***2*** és nem ***2.00***. Utóbbiak egyike sem fog működni, a változatszám ellenőrzése ugyanis karakterláncok összevetésével és nem számértékek összehasonlításával történik. Hogyan néz ki az ***RSS 2.0***? Nem fog túl sok meglepetést okozni (*3. lista*). Amit fent látunk, nagyon hasonlít a ***0.91***-es ***RSS***-re, de akár az ***RSS 1.0*** lecsupaszított változatának is tekinthető. Ha viszont felidézzük azt a tényt, hogy az ***RSS 2.0*** a ***0.91*** utódja, melynek feladata – a kis méret, az egyszerű megvalósíthatóság és rugalmasság megőrzése mellett – a felmerült hiányosságok pótlása volt, azonnal minden megvilágosodik.
Az ***RSS 2.0*** számos fejlesztést hordoz magában a ***0.91***-hez képest, a legfontosabb talán a névterek modulokként való használata, amivel új szolgáltatások valósíthatók meg.
Az ***RSS 2.0*** által megadott vagy használt névterek száma messze nem közelíti meg az ***1.0***-nál látottat, ám ennek fő oka az, hogy nem próbálkozik meg az ***RDF*** megvalósításával.
Winer részben a ***RSS 2.0*** szabálygyűjtemény szerzői jogának megtartása miatt őt ért kritikák hatására a jogokat a ***Harvard Universitynek*** adta át. Feltételezhető, hogy ***Winer*** továbbra is fontos szerepet fog játszani az ***RSS 2.0*** fejlesztésében, ám nem ő lesz az, aki végső soron dönt a használattal vagy a bővítésekkel kapcsolatos kérdésekben.
A szétválás mindettől függetlenül véglegesnek tűnik. Kialakult egy ***Atom csoport*** és egy ***RSS csoport***, én nem nagyon hiszem, hogy valaha is közös nevezőre jutnak. A fejlesztői táborok által kitűzött célokat szemlélve ez a legkevésbé sem meglepő – végül is nem várhatjuk el, hogy ugyanaz a szabálygyűjtemény egyszerre törekedjen az egyszerűségre és a rugalmasságra.

Összefoglalás

Ez alkalommal az ***RSS*** jelenleg is használatban lévő változatait tekintettük át, illetve összevetettük stílusukat és fejlesztőik célkitűzéseit. Szerencsére, ha valaki szeretne egyszerű, begyűjtésre alkalmas cikkeket készíteni, különösebb nehézségekre nem kell számítania. Bár a programozók az egyes változatokra egyedileg jellemző mezőket is hozzáadhatnak, az alap mindegyik ***RSS***-változatnál azonos, függetlenül attól, hogy az együttműködésnek még a szándéka is hiányzik. A létrejövő ***RSS***-cikkek természetesen egészen eltérő kinézetűek is lehetnek, függően a kiválasztott változattól.
Következő írásomban az ***RSS*** nemrég megjelent, ám egyre elterjedtebb vetélytársáról, az ***Atom*** cikkgyűjtő formátumról lesz szó. Ha azzal is végeztünk, akkor áttekintjük, hogyan készíthetünk saját cikkgyűjtőt, amellyel különböző forrásokból származó cikkeket tudunk értelmezni és kezelni. Megvizsgáljuk az ***RSS*** használatának különféle módjait is, illetve azt, hogy a cikkgyűjtők révén hogyan juthatunk hozzá a legfrissebb hírekhez és a legújabb véleményekhez.

Linux Journal 2004. október, 126. szám

Reuven M. Lerner (➲ http://www.lerner.co.il/atf) Nyílt forrású programokra, valamint web- és adatbázis-alkalmazásokra szakosodott tanácsadó. Könyve, a Core Perl, 2002 januárjában jelent meg a Prentice Hall gondozásában. Reuven feleségével és lányaival nemrég költözött Chicagóba.