

Kiadás Bricolage-alapokon

Működtessük weboldalunkat a profi híroldalak modellje alapján: a Bricolage segítségével írásokat hagyhatunk jóvá, és nyomon követhetjük, hogy ki mit szerkeszt.



A ki a hálózaton híroldal vagy magazint üzemeltet, az valószínűleg hallott már a tartalomkezelő rendszerekről (CMS). A CMS program leegyszerűsíti a webhely számtalan oldalának a kezelését, ugyanis segít a felhasználók, a csoportok, a jogosultságok, a szerkesztési felelősség és az egész honlapon közös sablonok rendszerezésében. Minél több oldal van a webhelyünkön, minél több ember dolgozik a rajta, annál valószínűbb, hogy sokat nyerünk egy CMS telepítésével.

Sok cég csapott le a kínálókozó lehetőségre és kínál CMS programokat, amelyek ígéreteik szerint szinte minden szolgáltatást készen nyújtanak – a webtartalom megírását kivéve. (Bár egy-két oldal írásainak a minőségét elnézve az ember néha egészen elbizonytalanodik.) Igaz, beletelt némi időbe, de mára számos nyílt forrású CMS csomag is elérhető, amelyek a legösszetettebb honlapokkal szemben támasztott igényeket is kielégítik. Az egyik leghatékonyabb és legnépszerűbb ilyen csomag a Bricolage, amely az Apache kiszolgáló `mod_perl` modulja és a PostgreSQL relációs adatbázis-kezelő, valamint a `HTML:Mason` sablonrendszer erejét ötvözi egyetlen elegáns csomagban, lásd a <http://sourceforge.net/projects/bricolage> címen. Sorozatunk korábbi részeiben megismerhettük a Mason rendszer néhány alapbeállítását és a bizonyos események bekövetkeztekor induló riasztásokat. Egészen mostanáig azonban elhanyagoltuk minden CMS legfontosabb képességét, nevezetesen a tartalomnak a végleges

weboldalon történő megjelentetését. Ebben a hónapban a cikkek útját a megszületésüktől egészen a kiadásukig követjük végig, azt, ahogyan végighaladnak a Bricolage kiadási útvonalán.

A webhely létrehozása

Mindenekelőtt létre kell hoznunk a weboldalt, amelyen aztán a kiadandó tartalom megjelenhet. A gépemem készítettem egy virtuális webhelyet `output.lerner.co.il` néven, külön könyvtárral a hiba- és elérési naplónak. Azután a `listán` látható VirtualHost-utasításokat vettem fel az Apache beállításfájlba.

Mostantól, ha a kiszolgálóm az `output.lerner.co.il` címre érkező kérelmet fogad, az `output.lerner.co.il/www` könyvtárban fog keresni a fő dokumentumgyökér helyett, amelyet egyébként általában a `/usr/local/apache/htdocs` könyvtárban találunk.

Mielőtt a Bricolage-dzsel cikkeket tehetnénk közzé a weben, meg kell mutatnunk a CMS-nek, hogy hol találja az új fájlokat. A Bricolage rendszerében ezt a `Distributions` fejléc alatt található `Destinations` menüpontban tehetjük meg. Egy időben több kimeneti célt is készíthetünk, így egyetlen Bricolage-példánnyal több weboldalt is kezelhetünk. Ilyen eset könnyen előállhat olyan kiadónál, amelynek a csapata egyszerre több különböző újságon is dolgozik. A kimeneti célok lehetnek helyi vagy valamilyen FTP-vel elérhető távoli fájlrendszeren.

Készítsünk új kiadási célt a `New Destination` pontra

VirtualHost-utasítások az Apache-beállításfájlban

```
<virtualHost 69.55.225.93>
  ServerName output.lerner.co.il
  ServerAdmin reuven@lerner.co.il
  DocumentRoot /usr/local/apache/v-sites/output.lerner.co.il/www
  CustomLog /usr/local/apache/v-sites/output.lerner.co.il/logs/access-log combined
  CustomLog /usr/local/apache/v-sites/output.lerner.co.il/logs/referer-log referer
  ErrorLog /usr/local/apache/v-sites/output.lerner.co.il/logs/error-log
</virtualHost>
```



1. kép Új cím beállítás



2. kép A Edit Desken ellenőrizhetjük a szerkesztésre kész történeteket

való kattintással. A legtöbb kis lapnak elegendő egyetlen csatorna, ami lehetővé teszi, hogy valamennyi dokumentumot a webhely *DocumentRoot* könyvtárába helyezzünk el a lemezen. Ezt követően be kell állítanunk még néhány dolgot:

1. A weboldal és a Bricolage rendszer azonos rendszeren található-e vagy a fájlokat FTP segítségével távoli rendszerre szükséges másolni.
2. Az ehhez tartozó kimeneti csatornát személyesen elérhetővé (private) kell tennünk. Feltételezzük, hogy a Bricolage kiszolgáló és a végleges webkiszolgáló azonos számítógépen található – nagyobb méretű környezetben, különösen ott, ahol a teljesítmény lényeges, bölcs dolog a kettőt elválasztani egymástól. Az első cél képernyőjét az 1. képen láthatjuk.

Bár a Bricolage új kimeneti csatornák meghatározását is lehetővé teszi, egyelőre tartuk meg az eredeti webcsatornát. A kimeneti csatornák és a célok közötti kü-

lönbség elsősorban meglehetősen zavaró lehet. A kimeneti csatornákat képzeljük el logikai, míg a célokat fizikai célpontként, és rögtön látni fogjuk, hogy rengeteg különféle kombinációban ötvözhetjük a kettőt. Egyszerre több kimeneti csatornánk is lehet, amelyek egyetlen célra irányulnak, vagy egyetlen kimeneti csatornánk, amely több célra irányul.

Az új célunkkal kapcsolatos alapvető adatok megadása után meg kell adnunk legalább egy műveletet (move), majd meg kell határozni a legfontosabb részt: a kiszolgálószakaszt. A kiszolgálószakaszban adhatjuk meg, hogy melyik kiszolgáló tartalmazza a fájlokat. Mivel korábban már jeleztük, hogy a fájlokat az FTP használata helyett csak másolni fogjuk, elegendő a cél elérési útját megadnunk – ennek azonosnak kell lennie az Apache-ban megadott virtuális gép *DocumentRoot* bejegyzésével, s ezután innen érhetjük el a fájlokat.

Írás és kiadás

Miután létrehoztuk a kimeneti csatornát, visszamegyünk, feltesszük a riportersapkánkat és létrehozunk egy új cikket. (Ne felejtsük el, hogy ezeket a menüket egyébként a magazin csapatának különféle emberei vagy csoporttagjai használhatják.) Kattintsunk a *Story* menüben a *New Story* menüpontra, adjuk meg legalább a címet és a kategóriát (/), ha csak korábban nem határoztunk meg egyéb kategóriát), valamint a fontosságot (*normal*). Ezenkívül beállíthatjuk a kiadási dátumot, amely nemcsak a cikk megjelenésének dátumát adja meg a weben, hanem azt az URL-t is, amellyel azután elérhetjük. A Salon.com régebbi olvasói valószínűleg megfigyelték már, hogy a címek a kiadás dátumát is tartalmazzák; így azután nem meglepő, hogy a Bricolage, amely a Salonnál fejlesztett CMS-ből alakult ki, követi ezt a hagyományt.

Miután ezeket az alapvető adatokat megadtuk, kattintsunk a *Create* gombra. Itt lehetőségünk lesz elkészíteni a cikk szövegét, új bekezdéseket felvenni (egy időben egyet, vagy a nagy *Edit* gomb használatával üres sorokkal elválasztva). Amikor befejeztük a szerkesztést, az oldal alján található *Check In* gomb lenyomásával átküldhetjük az *Edit Desk*-re.

Sajnos a webalapú felületek természetéből adódóan, illetve a Bricolage nyújtotta rugalmasság következtében a Bricolage felülete gyakran egy kicsit szövevényesnek tűnhet, és számos különféle küldőgombbal vagyunk kénytelenek dolgozni. Szerencsére a gombok felirata egyértelmű – esetleg előfordulhat, hogy a *Edit Desk*-re továbbítás helyett mentjük a történetünket, de véletlenségből biztosan nem fogjuk törölni. Azt mondják, az új Bricolage-felhasználóknak érdemes elolvasniuk a gombok feliratát, hogy ne is feltételezzék, hogy minden *Submit* gomb ugyanúgy működik.

A cikket tehát elküldtük a *Edit Desk*-re, úgyhogy ismét lecseréljük a kalapunkat, és úgy teszünk, mintha mi lennénk a szerkesztők, akiknek a cikkeket el kell bírálniuk. Kattintsunk a *Desks* alfejléc alatti *Edit* menüpontra a *Stories* menüben, és máris olvashatjuk a *Edit Desk*-re jelenleg elküldött valamennyi cikket (2. kép). A már kiadott írások címsorában P betűt találunk. A sorból kiemelt (checked out) írások (ezek azok, amelyeket szerkesztői változtatásra



3. kép A kiadásra váró történetek átnézése a Publish Desken

jelölünk ki) átszerkeszthetők. Mi azonban most nem szeretnénk átírni a történetet, hiszen tudjuk, hogy a szerző olyan ember, aki mindig befejezi a mondatait és módszeresen ellenőrzi a munkáját. Ezért tovább is küldjük a **Publish Desk**-re, ahol a cikkeket utoljára átnézik, végül kikerülnek a webre. Ehhez a **Move To** menü **Publish** pontját választjuk, majd a **Move Assets**-re kattintunk.

Most már átléphetünk a kiadói munkafelületre, ahol az elemeket utoljára átnézve kihelyezhetjük őket a webre. Figyeljük meg, hogy a Bricolage beépített támogatással rendelkezik az olvasószerkesztéshez és a jogi felülvizsgálat kezeléséhez. Amennyiben szervezetünk felépítése az alapértelmezettnél összetettebb, további munkafelületeket is készíthetünk.

A **Publish Desk** sok tekintetben hasonló a többihez: megjegyzéseket fűzhetünk a történetekhez, megtekinthetjük a történet szerkesztői útját, vagy belepillantunk a naplőbejegyzésekbe. Az írásokat a **Publish Desk**-en is szerkeszthetjük: emeljük ki (check out) őket, majd kattintsunk az **Edit** hivatkozásra. A **Publish Desk** legfontosabb tulajdonsága, hogy itt egy (nem meglepően **Publish** névre hallgató) további jelölőnégyzetet és egy **Publish Checked** feliratú gombot is találunk (3. kép).

Amint rákattintunk **Publish Checked** gombra, a Bricolage megkérdezi, hogy mikor szeretnénk az írást megjelentetni. Ez nem feltétlenül egyezik meg az írás kiadási dátumával. Elképzelhető, hogy egy február 1-jei dátummal ellátott művet már január 31-én elérhetővé szeretnénk tenni; de éppúgy lehetséges, hogy csupán február 7-én szeretnénk közzétenni. A **Publish Assets** gombra kattintva megindul a folyamat, a Bricolage pedig majd jelenteni fogja, ha közreadta a kérdéses cikket (vagy történetet). Ekkor egészen biztosak lehetünk benne, hogy a dokumentumunk felkerült a korábban megadott címre – jelen esetben az **index.html** néven a **DocumentRoot** könyvtárba.

Kezelési kérdések

Ha egy nyomtatott újság vagy magazin valamilyen cikket megjelentet, többé nincs lehetőség a kiadott változat javítására. A weben azonban nem kell mást tennünk, mint hogy átszerkesztjük a kérdéses dokumentumot, és máris mindenki láthatja a változást. E technológiai lehetőséggel

kapcsolatos újságírói etikai kérdéseket sutba dobva a Bricolage a cikkek könnyű megváltoztatását és weboldalra való újraküldését teszi lehetővé.

A már kiadott cikkek megváltoztatásához válasszuk az **Active Stories** menüpontot, amely felhossa a jelenleg elérhető történeteket. Kiválaszthatjuk a kiemelő cikket és a pillanatnyi felhasználó számára szerkeszthetővé tehetjük. (A Bricolage változatkezelő rendszere biztosítja, hogy egy adott dokumentumot egy időben csak egyetlen felhasználó szerkeszthesse.) Azután átszerkesztjük a fájlt és elküldhetjük a **Edit Desk**-re, pontosan úgy, ahogy az eredeti cikkel is tettük. Szintén, akár csak az előbb, az **Edit Desk**-ről kell az írást továbbküldeni a **Publish Desk**-re, ahonnan aztán kikerülhet a webre.

Azok számára, akik ez idáig csak kis weboldalaknál dolgoztak, az ilyesmi talán egy kicsit bonyolultnak tűnhet. Végtegre is nem éppen az az egyik vonzereje a webnek, hogy a dokumentum szerkesztése során a változtatás azonnal láthatóvá is válik? Tény, hogyha weboldalunkhoz tartalomkezelő rendszert használunk, bizonyos engedményeket kell tennünk. Többé már nem módosíthatjuk egyszerűen a fájlokat a merevlemezen; kénytelenek vagyunk bejelentkezni, kiemelni a fájlokat, elvégezni a módosításokat, visszahelyezni őket, és elküldeni a **Publish Desk**-re, ahol a változtatásainkat esetleg el is utasíthatják. Azonban bármilyen összetett és bürokratikus is ez a módszer, még mindig sokkal jobb, mintha számtalan ember harcolna a fájlok birtoklásáért, különböző emberek szerkesztenék ugyanazt az állományt – néha eltérő módon – egy élesben működő honlapon. A CMS-t éppen arra tervezték, hogy lelassítson minket, ahogy a biztonsági öv is némiképp gátolja a mozgásunkat. Mindkét esetben az a megkötések célja, hogy ne üssük meg magunkat. Ráadásul a bürokrácia még némileg a hasznunkra is válhat, hiszen a dokumentum valamennyi előző változata elérhető. Ha valamelyik fontos történet felét véletlenségből letöröltük volna, könnyedén visszaterethetünk a korábbi változathoz, hiszen a Bricolage tárolta az adatbázisában.

Összegzés

A Bricolage teljes körű, lenyűgöző képességekkel rendelkező, kifinomult, nyílt forráskódú kezelőrendszer. Mint azt ebben a hónapban láthattuk, Bricolage alapú weboldalt létrehozni és az Apache beállításfájlban néhány egyszerű utasítással meghatározott kiadási csatornákon keresztül írásokat közreadni nem is olyan nehéz. A következő részben bricolage-es vizsgálódásaink végéhez érünk és megnézzük, hogyan formálhatjuk igényeink szerint a kimeneti sablonokat, és az egyszerű fekete-fehér kiadványt miként varázsolhatjuk egy kicsit izgalmasabbá.

Linux Journal 2004. január, 117. 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 a feleségével és lányaival nemrég költözött Chicagóba.