© Kiskapu Kft. Minden jog fenntartva

Bricolage-riasztások

Kövessük nyomon honlapunk eseményeit kulcseseményekhez rendelt riasztásokkal.

A múlt hónapban elkezdtük nézegetni *David Wheeler* Bricolage nevű, hatékony, `mod_perl`, `HTML::Mason` és PostgreSQL alapokra épülő, nyílt forrású tartalomkezelő rendszerét (CMS). A tartalomkezelő program viszonylag új típusú webalkalmazás, célja a méretesebb honlaprendszerek kezelési nehézségeinek a kiküszöbölése. A Bricolage megjelenését széles körben új nyílt forrású sikertörténetként ünnepelték, amely fényesen bizonyítja, hogy egy üzleti program nem feltétlenül rugalmasabb vagy hatékonyabb ingyenes társainál. Nem szabad elfelejtenünk, hogy bár a CMS olyan webalapú alkalmazás, amely dinamikusan létrehozott kiszolgálóoldali alapoktól függ, a CMS kimenete általában statikus oldal lesz. Ezért aztán annak ellenére, hogy a Bricolage viszonylag nagy web-, illetve adatbázisprogram, amelynek testreszabásához nem árt némi kiszolgálóoldali programozási ismeret, mégis olyan alkalmazás, amelyet nap mint nap nem programozók használnak. A Salon.com-tól kezdve a MacWorld On-line-ig rengeteg szervezet használja már most is a Bricolage-t. Ugyanakkor további weboldalak ezrei, a Cnettől a LinuxJournal.com-ig, tartalomkezelő megoldásokat használnak a Vignette rendszerétől (összetett, üzleti és drága) a PHPNuke-ig (egyszerű, nyílt forrású és ingyenes).

Ebben a hónapban az egyik kedvencemnek, a Bricolage-nak a riasztásképességét nézzük meg. Ez lehetővé teszi, hogy a felhasználó különböző eseményeket kövessen nyomon a Bricolage rendszerén. A riasztások nemcsak tájékoztatnak bennünket az események bekövetkeztéről, hanem egyúttal képet kaphatunk belőlük a teljes rendszer működéséről is – megtekinthetjük az objektumok listáját, feladatokat rendelhetünk ezekhez az objektumokhoz, valamint láthatjuk, hogy pontosan mikor hívódnak meg az egyes riasztások.

Riasztások

A Bricolage, csakúgy mint a legtöbb CMS-program, a cikkeket különféle állomásokon keresztül mozgatja. A Bricolage alatt ezeket az állomásokat asztaloknak (desks) nevezik, utalva ezzel a program újságírói gyökereire. A történetet ennek megfelelően a szerkesztőasztalon (edit desk) kezdik, majd átkerülnek a másolóasztalra (copy desk), onnan a jogi asztalra (legal desk), végül a kiadói asztalra (publish desk), ahonnan már a hálózatra jutnak. Maroknyi csapatok által vezetett honlapok esetében még könnyű nyilvántartani, hogy melyik cikk éppen hol található. De ha egyszer túlléptünk a kis szerkesztőgárdán vagy a csekély cikkszámon, igen nehézkessé kezd válni az egyes cikkek nyomon követése. Az egyik lehetőség, amivel ellenőrizhetjük a munkafolyamatot, ha a munkaasztalokat egyesével rendszeresen végignézzük, hogy lássuk, éppen milyen történetek vannak rajtuk, majd ennek megfelelően megtesszük a szükséges lépéseket. Ez a nézegetés azonban idővel elég fárasztó, ráadásul könnyen előfordulhat, hogy az összes cikk helyett nekünk csak a hírek kategóriát kellene átnéznünk, esetleg csak egy adott szerző műveit. Továbbá nem lenne rossz a munkafolyamat-eseményekről elektronikus levélben értesítést kapni.

A Bricolage rendelkezik ilyen lehetőségekkel, mégpedig igazi nyílt forrású stílusban, lenyűgözően testreszabható módon. A riasztások létrehozásához és módosításához válasszuk a képernyő jobb alsó sarkában található rendszermenü alól a ***Alert types*** menüt (riasztástípusok). Ez a menü a rendszergazda menüi közé tartozik, tehát csak a rendszergazdai jogosultságokkal rendelkező felhasználóknak jelenik meg. A kapott képernyőn – mint azt már a Bricolage karbantartó menüinél megszokhattuk – lehetővé teszi, hogy rákeressünk egy riasztás nevére, vagy kikeressük az összes, az adott betűvel kezdődő riasztást.

Riasztások létrehozása

Új riasztástípust a ***create new alert type*** hivatkozásra kattintva készíthetünk. Itt egy rövid HTML-űrlaphoz jutunk, ahol meg kell adnunk azt a Bricolage-objektumot, amelyikre a riasztást kérjük. Tehát például akkor szeretnénk riasztást kapni, ha valami, a történeteket (story) érintő dolog történik, ekkor kérjük a ***Story*** objektum riasztását. Ugyanígy, ha új felhasználó felvételekor szeretnénk riasztást, kérjük a ***Users*** objektumhoz tartozó riasztását. Tömören, szinte minden Bricolage-objektumhoz rendelhetünk riasztásokat.

Példaképpen készítsünk egy olyan riasztást, amelyik akkor figyelmeztet, amikor valamely, a címében a „Linux” szöveget tartalmazó cikk az egyik asztalról a másikra kerül át. Ne feledjük, a Bricolage segítségével szinte minden objektum megfigyelhető. A következő példa kétségtelenül adja magát. Válasszuk a ***create new alert type*** hivatkozást az ***Admin*** menüből, és készítsünk új riasztást a ***Story*** objektumhoz. Itt a Bricolage által megfigyelhető eseménylistához jutunk, a kategóriahozzáadástól kezdve a történetünk kiadásán át a történet valamely részének törléséig. Példánkban mi a ***Story moved to desk*** (a történet mozgatása asztalok között) lehetőséget választjuk, majd a ***Linux story moved*** nevet adjuk neki.

Minden riasztástípushoz tartozik egy tulajdonos; jelen esetben a tulajdonosok mi leszünk, hiszen a saját nevünkön léptünk be. A Bricolage egyetlen felhasználót készít el magától, a ***Bricolage Administrator***-t, mégpedig `admin` bejelentkezési néven. Az `admin` felhasználó a Unix-rendszerek `root` felhasználójának, vagyis a rendszergazdának felel meg. Nyilvánvaló módon riasztásokat is birtokolhat, de sokkal jobban járunk, ha készítünk magunknak egy másik felhasználót (a bal alsó sarokban található ***Admin/User*** menüpontban), majd megadjuk a rendszergazdai jogosultságokat és a továbbiakban így jelentkezünk be (a `root` helyett).

Valamely esemény bekövetkeztekor ennek a lapnak az alján található ***next*** gombra kattintva a fő riasztástípus-szerkesztő oldalra jutunk, ahol új riasztástípusokat hozhatunk létre, illetve módosíthatjuk a már meglévőket. Minden riasztás négy részből áll:

- ***Properties*** (Tulajdonságok): a riasztástípus neve és tulajdonosa, amit az előző oldalon gépeltünk be.

© Kiskapu Kft. Minden jog fenntartva

- ***Rules*** (Szabályok): megadja, hogy mikor kell a riasztásnak megszólalnia. Minden szabály tartalmaz egy változót (amit lenyíló menüből választunk ki, így elkerülhetjük az esetleges elgépeléseket), összehasonlító tesztet és egy szövegmezőt, ahová az összehasonlítandó értéket írhatjuk be.
- ***Content*** (Tartalom): az elektronikus levélben levő üzenet, amit a riasztás címzettjei megkapnak. Ebben több különféle változót is használhatunk, a cikk címétől kezdve a kiadás dátumáig.
- ***Recipients*** (Címzettek): azok a felhasználók és csoportok, akiknek a riasztást szánjuk. Tudunk például riasztást küldeni az összes szerkesztőnek, az összes szerzőnek vagy éppen Gyurinak és Ferinek, de Debinek és Máriának nem.

Riasztási szabályok

A Bricolage riasztási rendszerének talán legérdekesebb része a szabályok alkotása, hiszen olyan egyszerűen alkalmazhatjuk őket, hogy a programozáshoz semmit sem értők is elboldogulhatnak velük. Ugyanakkor a `=~` és `!~` műveleti jeleknek (amelyeket látva egy Perl-programozó rögtön szabályos kifejezésekre gondol) akad egy nagy veszélyük. Egyértelműen kétélű fegyverről van szó, hiszen a szabályos kifejezések rendkívül hatékonyak a felvilágosultak kezében, de hihetetlenül veszélyesek (és elkedvetlenítők) a tudatlanokra nézve.
Riasztásunk létrehozásához a bal oldali tulajdonságlistából kiválasztjuk a ***Story title*** elemet, majd a `=~` jelet az összehasonlító műveleti jelek (operator) listájából, végül a `Linux` szöveget gépeljük be a szövegmezőbe értékként. Így a következőt kapjuk:

```
Story title =~ Linux
```

A `=~` műveleti jelet és nem az egyszerű `=` kell használnunk, mivel a Linux szót a cím bármely részében keressük és nem csak a teljes címmel való egyezést fogadjuk el. Amennyiben a Linux és a Perl is érdekel minket, a következőre is kereshetnénk:

```
Story title =~ Linux|Perl
```

A tapasztalt Perl-programozók biztosan örömmel hallják, hogy a `=~` és a `!~` műveleti jelek itt nem figyelik a kis- és nagybetű közti különbségeket.
A riasztás témája és szövege tetszőleges szöveget tartalmazhat, ideértve Bricolage által előre meghatározott behelyettesíthető Perl-változókat is. Egy szép JavaScript-megoldással a Bricolage lehetővé teszi, hogy a változókat lenyíló menüből válasszuk ki, elkerülve ezzel a gépelési és egyéb hibákat. Egyben igen hasznos jegyzék, hiszen nem kell megjegyeznünk az összes elérhető, a cikkekhez tartozó változónevet. Ezzel a módszerrel könnyen összeállíthatjuk a következő alakú riasztáscímet:

```
A $title cikk átkerült a $desk asztalra
```

Bár a riasztásüzenet valószínűleg más szöveget tartalmaz, itt is használhatjuk a változóneveket. A szövegtestbe beírhatjuk például a következőket:
Ön kérte, hogy figyelmeztessük, amennyiben a Linux témájú cikkek új asztalra kerülnek át. Nos, $trig_full_name épp most helyezte át a "$title" című cikket a $desk asztalra. Reméljük, segítettünk!

A riasztások fogadása

Amikor egy riasztás életbe lép, a kiválasztott címzettek két különböző módon kaphatják meg az üzenetet. Valamennyi bejegyzett felhasználó levélben üzenetet fog kapni, amely a fent bemutatott levélben tájékoztatja őket a változás bekövetkeztéről. A Bricolage ugyanakkor fel is jegyzi ezeket az eseményeket a saját adatbázisában, ami igen megkönnyíti a riasztások webes követését. Amikor a fent leírt linuxos riasztás életbe lép, levélben jelzést kapunk. Ugyanakkor valamennyi riasztást megtekinthetjük a riasztási oldalunkon. Ezt a képernyő tetején található, a ***my alerts*** gombra kattintva tekinthetjük meg.
A riasztások egészen addig maradnak ezen az oldalon, amíg nem jelezzük, hogy tudomásul vettük őket. A riasztásoldalt nem hosszútávú tárolóhelynek szánták; sokkal inkább egyszerű üzenetküldő rendszernek, amely lehetővé teszi a szerkesztőknek, hogy egyszerre lássák a honlapon akkortájt történt összes lényeges eseményt.
A riasztásokat olyan módon vehetjük tudomásul (ezzel törölve őket az oldalról), ha a listában egy vagy több jelölőnégyzetet bekapcsolunk, majd a lista alján lenyomjuk az ***acknowledge checked*** gombot. A jelölőnégyzetek használata nélkül is tudomásul vehetjük az összes üzenetet egyszerre, ha az ***acknowledge all*** gombot választjuk.

Tanulságos riasztások

A riasztások nagyszerű és hasznos eszközök, amelyekkel a Bricolage képes a vonatkozó adatokat eljuttatni a megfelelő felhasználókhoz, és így nem kényszeríti rá őket, hogy maguk keresgéljék ki őket. Mindeközben áldásos mellékhatásként a friss Bricolage-rendszergazdáknak és -programozóknak segít megérteni, miképpen jönnek létre különféle események az egyes objektumokban. Egyszerű példa erre a ***User*** (felhasználó) objektum: készíthetünk olyan riasztást, amely akkor figyelmeztet, ha egy felhasználót létrehozunk vagy törlünk, illetve megváltoznak a beállításai vagy a jelszava. Az ilyesfajta riasztás egy szerkesztő szemében nem ér különösebben sokat, ugyanakkor elsődleges fontosságú lehet a rendszergazda számára. Valamivel összetettebb példa a ***Template*** (sablon) objektum, amellyel a következő hónapban ismerkedhetünk meg. A sablonok határozzák meg, hogyan jelenjenek meg a történetek, ezért igen fontos a megfigyelésük és a változásaik nyomon követése. Ezért aztán a sablonokhoz több beállítás tartozik, mint a felhasználókhoz – nyomon követhetjük a sablon telepítését, szerkesztését, akár azt is, hogy mikor kerül fel az asztalára. Amennyiben esetleg korábban nem értettük volna meg, hogy akárcsak a történeteknek, a sablonoknak is van saját asztaluk, a riasztórendszerből ez világosan kiderül.
Én elég sokat tudtam meg a Bricolage rendszerről, miközben különböző szemszögekből megvizsgáltam, ideértve a riasztásmegadásokat is. Ha még újak vagyunk a Bricolage rendszerben, és a dokumentumok végigolvasása és a menük kinyitogatása után (ami igen fontos, amennyiben tisztában szeretnénk lenni a Bricolage oldalszerkezetével és jogosultságrendszerével) sem igazán értettük meg a dolgok belső lelkivilágát, a riasztásmeghatározások végignézése rengeteg dologra rávilágít.
A cikkhez tartozó kapcsolódó címek megtalálhatóak az 53. CD Magazin/Bricolage könyvtárában.

Linux Journal 2003. október, 114. 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 Izraelben, Modi'inben él.