

A Zope CMF

Működtessük weblapunk munkafolyamat-rendszerét tartalomszolgáltatóink és felhasználóink meglegedésére. Fejlesszünk Python alapú tartalomkezelő keretrendszerrel!



Az elmúlt néhány hónapban megtudhattunk néhány dolgot a tartalomkezelésről, illetve egy kicsit részletesebben is megismerkedhettünk a nyílt forrású Plone tartalomkezelő rendszerrel (CMS). A Plone igen rátermett CMS rendszer, olyasmí, amit az ember telepítés után majdhogynem azonnal használni tud. A Plone legnagyobb előnye az egyszerűség. Könnyű telepíteni, könnyű használni és egyszerű beállítani.

A Plone (vagy bármely más CMS) testreszabásának azonban megvannak a határai. Ha mint nem kereskedelmi termék nem tetszik a Plone kinézete, természetesen megváltoztathatjuk egy másik modellel. De ha teljesen más munkafolyamatot szeretnénk, mint amit a Plone nyújt, valószínűleg csak az időnket vesztegetjük, ha ilyesfajta változtatást próbálunk megvalósítani. Sokkal több értelme van, ha megírjuk saját CMS rendszerünket, vagy (ha bonyolultabb a feladat) átszabjuk azokat, amelyeket rugalmasan építettek fel.

A legtöbb CMS-terjesztő mostanra már ráébredt, hogyha használható terméket szeretnének, annak testreszabhatónak kell lennie. Amennyiben J2EE alapú CMS rendszert vásárolunk (márpedig a kereskedelmi CMS rendszerek igen nagy százaléka a J2EE-n alapul), várhatóan lehetőségünk lesz a tartalomtípusokat és azok kezelési módját leíró, új Java-objektumokat készíteni. Bizonyos ponton túl elmosódik a határvonal a meglévő CMS testreszabása és a saját CMS rendszer létrehozása között.

Itt lép be a képbe a Zope tartalomkezelője (CMF), amit olyan módon terveztek meg, hogy hatékony háttérrel nyújtson saját CMS-kezelőink megírásához. Mivel a Zope-fejlesztések igen gyorsak és egyszerűek, és mivel a Zope meglévő háttéreszközzeit is felhasználhatjuk, könnyen elképzelhető, hogy legalább ugyanolyan gyorsan (viszont sokkal olcsóbban) ki tudjuk fejleszteni CMS rendszerünket, mint ahogyan azt valamilyen kereskedelmi CMS-megvalósításban megtehetnénk. A Plone nem önálló program, hanem a CMF-re épül. Ennek megfelelően, ahogy a CMF fejlődik, úgy a Plone is egyre jobba válik.

Ebben a hónapban a CMF lesz terítéken, ami a Zope-alkalmazások közül lassan a figyelem középpontjába kerül. A CMF jelenleg elérhető legutóbbi üzembiztos változata az 1.3.1.-es, az alfáváltozat pedig a 2.0-s számot viseli. Ha esetleg az elkövetkezendő években a Zope irányvonala még nem lenne teljesen tiszta, a CMF 2.0 egyértelműen „a Zope3 némileg újracsomagolt vezető alkalmazása”.

A CMF telepítése és beállítása

A CMF telepítését néhány hónappal ezelőtt már bemutattuk (lásd a Kovácsműhely rovatot a Linuxvilág 2003. májusi számában), úgyhogy most nem részletezem. Töltsük le a forrásfájlokat a <http://cmf.zope.org> lapról, majd csomagoljuk ki a Zope saját könyvtárunk alatti

`lib/python/Products` könyvtárba.

Ezt követően a CMF könyvtárból a `Product` könyvtár `CMFCore`, `CMFDefault`, `CMFTopic` és `CMFCalendar` fájljaira készítsünk közvetett hivatkozásokat. Indítsuk újra a Zope-ot, és a Zope kezelőablakában az `Add` menü alatt máris rengeteg új CMF-vonatkozású terméket fogunk látni.

Mielőtt bármilyen CMF-objektumot hozhatnánk létre, előbb CMF-oldalunkhoz kell készítenünk egy tárolót. Nyilván észrevettük az egyértelmű párhuzamot a Plone-oldal és a CMF-oldal létrehozása között. Az új CMF-oldal létrehozásához a weblapú Zope kezelőfelületben egyszerűen csak válasszuk az `Add` menü `CMF Site` pontját. Ezt követően CMF-oldalunkat el kell neveznünk, illetve meg kell adnunk a leírását.

A CMF-oldal létrehozásakor válaszolnunk kell arra a kérdésre, hogy új felhasználói könyvtárat szeretnénk-e létrehozni az adott CMF-oldalon belül, vagy egy már létező felhasználói könyvtárat szeretnénk felhasználni. Egyelőre tegyük az utóbbit, ami azt jelenti, hogy a legfelsőbb szintű Zope-oldalon megadott felhasználók lesznek egyúttal CMF-oldalunk felhasználói is. Amennyiben CMF-oldalunkat inkább önálló egységként szeretnénk létrehozni, és (a lap tulajdonosán kívül) nem szeretnénk a külső világra hivatkozni, nyugodtan létrehozhatunk saját felhasználói könyvtárakat is.

Miután a CMF-oldal létrehozásával végeztünk, a honlapra kerülünk, ahol megtudhatjuk, hogy ideje meglátogatnunk az alapbeállítások űrlapját. Minthogy a CMF eredetileg Portal Toolkit néven (PTK) volt ismert, sok lap CMF-oldalak helyett portálokat emleget. Az űrlapba írandó adatok meglehetősen általános természetűek. Itt állíthatjuk be például azt a levélcímet, amiről (látszólag) a létrejövő elektronikus levelek érkeznek, valamint az oldal SMTP-kiszolgálóját.

A dolgok mindjárt érdekesebbé válnak, ha az útmutatást követve átlépünk a CMF-kezelőfelületre, ami valójában a CMF-oldalhoz tartozó Zope-kezelőfelület. Más szavakkal, ha oldalunk a `/cmfdemo` címen érhető el, az oldal tartalmába a `/cmfdemo/manage` címen nézhetünk bele. A kezelőfelület, mint azt már Zope alatt megszokhattuk, a bal oldalon egy kis navigációs menüt tartalmaz. Ugyanakkor, mint azt a múlt hónapban láthattuk, a bal oldalon számos portáleszköz is találunk, amelyekkel a CMF-oldalunkat állíthatjuk be és módosíthatjuk.

A `portal catalog`-ra kattintva megjelenik a szótár (angolul „vocabulary”, a Zope világában ezt a szót használják az index helyett), amiből megtudhatjuk, hogy az oldal gazdája miképpen készíthet teljes körű szövegkeresést a CMF-oldalon – minden erőfeszítés nélkül. A `portal types`-ra (portáltípusok) kattintva a tartalom objektumosztályokat pillanthatjuk meg. Ezek az osztályok alkotják a CMF magját. Alább részletesebben is megvizsgáljuk a tartalomosztályokat, a jövő

hónapban pedig azt is megtudhatjuk, hogyan készíthetjük el a saját tartalomtípusainkat.

Végül kattintsunk a *portal_workflow*-ra, ahol az egyes tartalomtípusokhoz használni kívánt *workflow* (munkafolyamat) objektum címét gépelhetjük be. A munkafolyamat szabályozza, hogyan jut el a tartalom az írástól a kiadásig, illetve biztosítja, hogy az emberek bizonyos feladatok elvégzéséhez a megfelelő jogosultságokkal rendelkezzenek. A szerzők például történeteket írhatnak, de nem tehetik közzé írásait az oldalon. A jó munkafolyamat-rendszer lehetővé teszi, hogy a saját szervezeti igényeinknek megfelelően magunk állítsuk fel ezeket a szabályokat.

Egyszerű CMF-lap

A CMF-vezérlőpanel bemutatása után ideje megvizsgálnunk a lapunkat is. Ha nyers CMF-oldalunkra belépünk, középen a fő tartalomterületet, továbbá különféle helyeken eszköztárakat és dobozokat láthatunk. A legfelső menüben találjuk meg a fontosabb navigációs pontokat, amikkel az oldal elejére, a tagok oldalára, a hírek oldalára, illetve az oldal tartalmának a keresésére ugorhatunk. A menü alatt, de még mindig a jobb felső sarokban felhasználói menükre bukkanhatunk, első helyen a *My Preferences* (saját beállítások) ponttal. Itt a bejelentkezett felhasználók felvehetik a saját beállításait, hivatkozásokat adhatnak kedvenceik listájához, illetve kijelentkezhetnek. A még be nem jelentkezett felhasználóknak a rendszer felkínálja a bejelentkezést, amennyiben létezik azonosítójuk, illetve akiknek még nincs, lehetőséget kapnak a csatlakozásra. Bal oldalon találjuk a navigációs menüt, ebben láthatjuk az elérhető könyvtárak listáját, illetve beállíthatunk néhány képességet, például a hírügynökséget (syndication) és a helyi szabályokat.

Aki már hozzászólt a Plone-oldalakhoz, az az alapértelmezett CMF-oldalt egy kicsit talán spártainak, de mindenképpen ismerősnek fogja találni. Az egyszerűség oka, hogy az alapértelmezett CMF-oldal arra készült, hogy egy saját CMS rendszerben felhasználhassuk; igaz, hogy teljes mértékben működőképes, de nem önálló alkalmazásnak szánták. Új tartalomtípusok meghatározásával és a megjelenítő felületek módosításával pillanatok alatt életre kelthetjük a CMS-t. Mivel a megjelenítő logika a rendszer többi részétől teljesen elkülönül, viszonylag gyorsan megváltoztathatjuk a kinézetet. A CMF-oldal minden egyes tagjához egy vagy több szerepet rendelhetünk: Member (tag), Reviewer (bíráló), Manager (vezető) vagy Owner (tulajdonos). Ezek a bíráló kivételével valószínűleg ismerősen csengenek a tapasztalt Zope-felhasználók és rendszerfelügyelők fülének. Az új szerep a munkafolyamatok kezeléséhez szükséges, ahol a bírálok feladata hitelesíteni a tartalmat, még mielőtt az felkerülhetne a Webre. A rendszerfelügyelők a képernyő bal oldalán kiterjedt menüt láthatnak, megtekinthetik a pillanatnyi könyvtár tartalomnézetét, illetve módosíthatják a meglévő tartalmat, valamint új tartalmat hozhatnak létre. Amikor új objektumot hozunk létre, nemcsak tartalmat és azonosítót (ID) rendelünk hozzá (amit a hagyományos webrendszerek fájlneveknek neveznek), hanem egyúttal a tartalmat leíró metaadatokat is. Bár az objektum típusát létrehozás után már nem változtathatjuk meg, valamennyi paraméterét átírhatjuk, ha a tartalomnézetbe visszatérve megnyitjuk a kérdéses tartalmat. Minden egyes tartalmat ki kell adni (publish), mielőtt azt mások is láthatnák. Alapértelmezés szerint az új tartalom állapota *private* (magán), de a bal oldali menüsor *publish* (kiadás) hivatkozásával kiadhatjuk őket. Ugyanezt a felületet

használva a lap felügyelője cikkeket vonhat vissza a lap kiadott elemeinek listájából. Ez igen nagy fejlődés a hagyományos weblapokhoz képest, ahol el kellett távolítanunk a csatolásokat és le kellett törölnünk a fájlokat.

Továbbá a legtöbb tartalomtípus vitafórummal rendelkezhet. Ez a *Comment on This Posting* (Megjegyzés ehhez az íráshoz) képességhez hasonlít, ami igen népszerű a weblogok világában, mivel lehetővé teszi, hogy a látogató megjegyzéseket fűzzön az oldal hivatalos rendszerfelügyelője által feltett íráshoz. Amikor egy új tartalmat felrakunk, eldönthetjük, hogy elfogadjuk-e a vitafórumokra vonatkozó alapértelmezés szerinti oldalmeghatározást, vagy ennél az egy példánynál felülbíráljuk a hely globális beállításait.

Tartalomtípusok

Nézzük meg, hogy tulajdonképpen milyen tartalomtípusokat is hozhatunk létre.

A legtöbbjük a CMFDefault termékben van meghatározva, a *lib/python/Products/CMFDefault* könyvtár különálló *.py*-állományában. Ez a termék írja le a Zope kezelőfelületen korábban látott beállítóeszközöket, valamint az olyan CMF alatt példányosítható alap-tartalomtípusokat, mint a *NewsItem* (hírelem), a *Portal*, az *Image* (kép) és a *Link* (hivatkozás).

A hozzám hasonlókat minden bizonnyal meglepi és lenyűgözi a CMF-ben meghatározott alap-tartalomtípusok egészen kis mérete. A lista alján a *NewsItem* áll, mindössze száz sorral, de még a *Portal* elem is csak 350 sort számlál. Ez nemcsak azt jelenti, hogy ezeket a tartalomtípusokat szükség esetén könnyű megváltoztatni és a hibáikat felderíteni, hanem azt is, hogy az új tartalomtípusok felvétele is igen egyszerű. Igazság szerint mára már számos új tartalomtípus született a CMF-hez, és továbbiak is várhatóak. Ha ellátogatunk a CMF Collective Projekt honlapjára (☞ <http://collective.sf.net>), számtalan CMF-vonatkozású terméket találhatunk, amik az elmúlt hónapokban jelentek meg. Születőfélben lévő CMF-termékek találhatóak többek között e-kereskedelem, fotóalbum és weblog témakörben. Ahogy a CMF egyre népszerűbbé válik, várhatóan a CMF Collective népszerűsége is egyaránt növekedni fog.

Összefoglalás

Figyelembe véve, hogy a Zope Corp. többször kijelentette, hogy a Zope jövője a CMF, és tudván, hogy a CMS telepítése felhárítónan költséges is lehet, egyértelmű, hogy a Zope Corp. komolyan megpróbálja túlszárnyalni kereskedelmi vetélytársait, és azok árai alá ígér. Ugyanakkor, mivel a Zope és a CMF nyílt forrású, saját termékeinkben nyugodtan felhasználhatjuk, akár a tartalomkezelésről szeretnénk többet megtudni, akár különböző elemeket szeretnénk szerkeszteni és kiadni. A következő hónapban egy kicsit mélyebben elmerülünk a témában, és megnézzük, hogyan írhatjuk meg a saját CMF példa-tartalomtípusunkat.

Linux Journal 2003. július, 111. 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 lányával Izraelben, Modi'in-ben él.