© Kiskapu Kft. Minden jog fenntartva

Bemutatkozik a Zope

A Zope alkalmazáskiszolgáló birodalmában tett villámlátogatásunk során a Zope-ot először mint webfejlesztőrendszert vizsgáljuk meg.

Mindegy, milyen programnyelvet vagy operációs rendszert használunk, a webfejlesztés alapvetően HTML-fájlok, képek, önálló programok, hibrid HTML, illetve -kódvázlatok és adatbázis-kapcsolatok tervezéséről szól. A tapasztalt fejlesztők könnyedén tudnak Perlről PHP-re vagy ASP-re váltani, hiszen az egyik nyelvről a másikra váltás az elgondolás szintjén csak a lehető legkisebb változást okozza. A paradigmák effajta közeledése a programozóknak igen kényelmes, igaz, ennek az az ára, hogy a webfejlesztői közösség berkeiben egyre gyakrabban tapasztalhatjuk az önelégültség és a lustaság jeleit.
Szerencsére itt van nekünk a nyílt forráskódú Zope alkalmazáskiszolgáló, amely felrázhat minket langymeleg elégedettségünkből, és a webfejlesztés teljesen új irányába vezethet bennünket. A Zope, amelyet a Zope Corporation (korábbi nevén Digital Creations) működtet, szinte mindent megváltoztat, amit a webfejlesztésről valaha is tudni véltünk. Ugyan továbbra is dinamikusan készülő tartalommal és relációs adatbázisokkal dolgozunk, de a Zope ezt a piacon ma megtalálható alkalmazáskiszolgálóktól teljesen eltérő módon teszi.
Ebben a hónapban a Zope-ot webfejlesztési szemszögből vizsgáljuk meg. Remélem, sikerül rávilágítanom, miben is különbözik a Zope minden más környezettől. Az általa képviselt finomság és hatékonyság bizony igen esélyes versenyzővé emeli a nyílt forrás rajongóinak kegyeiért folytatott harcban.

Mi is a Zope valójában?

A Zope megértésének egyik fő nehézsége, hogy valójában nem egy, hanem több dologról van szó. Egyszerűbben fogalmazva: a Zope minden, amire csak a webalkalmazások fejlesztéshez szükségünk lehet. A Zope telepítésével egy egyszerű HTTP-, FTP- és Web-DAV-kiszolgálót (amely Zserver néven ismert), egy objektum-adatbázist (ZODB), illetve egy kiszolgálóoldali alkalmazások fejlesztéséhez szükséges keretrendszert kapunk.
A Zope legnagyobb része Pythonban íródott, ami azt is jelenti, hogy a különféle felületek között könnyedén mozgatható. Bár köztudott, hogy a Perl, a Python vagy a Java nyelveken írt programok Linux- és Windows-rendszereken egyformán könnyen futtathatók, még mindig elég szokatlan, hogy egy kiemelkedő nyílt forrású rendszer Windows alatt is fut.
Egy átlagos honlap statikus HTML-fájlok, sablonok és grafikus állományok keveréke. Mikor a HTTP-kiszolgáló kérelmet fogad, először is meghatározza, hogy milyen típusú fájlkérésről van szó (ezt vagy a fájl kiterjesztéséből, vagy a MIME-típus alapján tudja eldönteni), majd ha szükséges, a megfelelő programsorokat végrehajtja, végül visszaküldi a HTTP-választ.
A Zope esetében azonban semmi sem tárolódik a fájlrendszeren. Ehelyett a teljes laptartalom és a programok a ZODB objektum-adatbázisban tárolódnak (az adatbázis maga általában egyetlen hatalmas lemezállomány formájában létezik, amelyet menteni lehet, vagy le lehet másolni, így készítve biztonsági mentést a Zope-kiszolgálóról.)
Ha például egy egyszerű HTML-fájlra van szükségünk, létre kell hoznunk egy „file” objektumot a ZODB-ban. Ha a honlapon képet szeretnénk megjeleníteni, a ZODB-ban egy képobjektumot kell létrehoznunk. Ugyanígy ha egy bizonyos URL meghívása esetén egy adott kódot szeretnénk futtatni, ezt a kódot is a ZODB-ban tároljuk.
Szerencsére az elemek tárolása és lekérése a ZODB-ban igen egyszerű. A Zope csaknem mindent elvégez helyettünk, és néhány magas szintű kapcsoló beállítását már a telepítéskor lehetővé teszi. Továbbá a Zope-ot úgy fejlesztették ki, hogy webböngésző segítségével is tökéletesen vezérelhető legyen. Alkalmazhatjuk a Zope webalapú eszközeit, amelyek segítségével a honlapot anélkül szerkeszthetjük és karbantarthatjuk, hogy egyetlen böngészőn kívül bármi mást használnunk kellene. Tehát lehetőségünk nyílik rá, hogy a honlapon bármely tartalmat (HTML, képek vagy könyvtárak) mindössze a böngészőnket használva készítsünk el, módosítsunk vagy töröljünk. Ha szöveget (és kódot) inkább az Emacsban szeretünk szerkeszteni, semmint a Web szövegmezőiben, kapóra jön a ZServer FTP-felülete, amely a ZODB objektumait úgy jeleníti meg, mintha fájlrendszer-hierarchiát néznénk. Ha mi (illetve az általunk kitanított avatatlan csoport) a fájlok webkiszolgálóra való felviteléhez és lehozatalához korábban már használt FTP-t, akkor a ZServer FTP-megoldásával a megszokott lépéseket szinte változtatás nélkül folytathatjuk.

A Zope telepítése

Mindezen összetett szolgáltatások ellenére a Zope-ot meglepően egyszerű telepíteni. Le kell töltenünk a legfrissebb változatot a ➲ http://www.zope.org-ról (lásd a ***Kapcsolódó címek*** részt). Miután a letöltendő változatot kiválasztottuk, váltsunk Zope könyvtárunkba (ez általában a */usr/local/zope/*, de bármi megteszi), és csomagoljuk ki a Zope fájlt:

```
mkdir /usr/local/zope
cd /usr/local/zope
tar zxvvf /downloads/Zope-2.4.3-linux2-x86.tgz
```

A Zope archívállomány megnyitásával számos könyvtár bukkan elő:

- ***bin***, ahol a Zope indító és leállító parancsfájljai rejtőznek;
- ***doc***, amelyben a teljes Zope-leírás megtalálható;
- ***lib***, ebben a Python- és minden egyéb Zope-alkalmazás (más néven termékek (products)) tárolódnak;
- ***ZServer***, amely a Zserverhez szükséges osztályokat tartalmazza;
- ***utilities***, amely néhány Zope-pal kapcsolatos segédeszközt foglal magában.

Mielőtt nekikezdhetnénk a Zope-pal való munkának, előbb a telepítő parancsfájllal telepítenünk kell. Ezt a parancsfájlt azonban csak akkor futtassuk le, ha a Zope-fájlokat már a

végleges helyükre mozgattuk, mivel a telepítés bizonyos teljes rendszerre érvényes értékek beállításához a működő könyvtár nevét használja fel.
Miután a Zope telepítését befejeztük, a fő Zope könyvtárban található `start` parancsfájl segítségével akár el is indíthatjuk. Ez a parancsfájl a Zope HTTP-kiszolgálót alapértelmezés szerint a 8080-as kapun indítja el (egy másik kapun pedig az FTP-kiszolgálót). Ezeket az értékeket parancssori kapcsolókkal megváltoztathatjuk; a kapcsolók teljes listáját a `start -help` begépelésével kaphatjuk meg.
Bár a Zope főként Pythonban íródott, nem feltétlenül szükséges, hogy Python telepítve legyen a gépünkön. A Zope saját Python-másolattal rendelkezik, és ezt fogja használni, bármi legyen is telepítve az operációs rendszeren. Ezáltal a változatütközések és hibák esélye mérséklődik. Jelenleg a Zope Python 2.1-et használ (azaz az utolsó megbízható változatot), de hamarosan megjelenik a Python 2.2 – érdekes lesz megfigyelni, mikor fogja a Zope beilleszteni.
A telepítő parancsfájl nemcsak a Zope alapbeállítását végzi el, hanem a rendszer `admin` felhasználója számára egy nehezen kitalálható jelszót is készít. Hamarosan szükségünk is lesz erre a jelszóra, úgyhogy ne felejtsük el feljegyezni valahová.

Zope Felügyelet

Ha a Zope-ot üzembe helyeztük, hogyan használhatjuk? Nos, az első és legfontosabb lépés a böngészőnk elindítása, majd a saját gépünk felkeresése a http://localhost:8080/ URL megadásával. Így néhány kezdeti Zope-adathoz jutunk, többek között például a leírásra és a honlapokra mutató hivatkozásokhoz (előfordulhat, hogyha esetleg más is fut a 8080-ason, az gondot okoz, ugyanis például a `wwwoffle` is azon keresztül fut – a ford.). Hagyjuk itt ezt a bemutatkozó lapot, és lépjünk tovább a Zope fő kezelőfelületére, a http://localhost:8080/manage címre. Itt meg kell adnunk a karbantartó felhasználói nevét és jelszavát – használjuk azt az `admin`-jelszót, amit a telepítő parancsfájl írt ki nekünk, amikor a Zope-ot először telepítettük.
Amennyiben a jelszót helyesen gépeltük be, böngészőablakunk két függőleges részre oszlik: a bal oldali rész a Zope-kiszolgáló szerkezetét mutatja, míg a jobb oldali részben az éppen kiválasztott objektumot vizsgálhatjuk meg részletesen.

Alapvető tartalom készítése

Nem nehéz megérteni, mi történik akkor, amikor a */foo/bar* dokumentumot lekérjük az Apache-kiszolgálótól. Az Apache megnézi, van-e ***bar*** nevű fájl a ***foo*** könyvtárban a dokumentumgyökér alatt. Ha létezik ilyen fájl, az Apache beolvassa a lemezről, és a tartalmát HTTP-válaszként visszaadja. Ha a fájl CGI-program, akkor végrehajtja a programot, és a kimenetet adja vissza HTTP-válaszként; és végül, ha a fájl nem létezik, az Apache a felhasználó böngészőjének hibaüzenetet ad.
A Zope az URL-eket másképpen értelmezi; a */foo/bar* URL például azt jelenti Zope alatt, hogy a foo-objektumban található barobjektum megjelenítő eljárását meg kell hívni. Ha nem létezik foo- vagy barobjektum, a Zope hibaüzenetet ad vissza, amelyben arra figyelmeztet bennünket, hogy a ZODB-ben nem talált ilyen objektumot.
Természetesen, ha minduntalan megjelenítő eljárások írásával és objektumok létrehozásával kellene vesződnünk, a Zope sem volna több egy különleges programozói játékszernél. Szerencsénkre a Zope lehetővé teszi, hogy számunkra a ZODB hierarchikus fájlrendszernek látszódjon, ahol HTML-fájljainkat és grafikáinkat elhelyezhetjük.
Például egy egyszerű HTML-fájlt akár a böngészőnk segítségével is felvihetünk. Lépjünk a Zope saját könyvtárába (ide bármikor visszatérhetünk, ha a bal oldali keret bal felső részére kattintunk), az új állomány létrehozásához válasszuk a DTML Document lehetőséget a jobb felső sarokban található ***Select type to add...*** listából (amint azt hamarosan látni fogjuk, a ***DTML/Document Template Markup Language*** a Zope kibővített HTML-nyelve; a Zope szinte mindig DTML-re hivatkozik HTML-dokumentumok helyett).

DTML-értékek beállítása

Böngészőnkön egy rövid, három részből álló HTML-űrlapot láthatunk:

- Az id szövegmező – a Zope alatt minden elemhez egy ID (azonosító) tartozik, ezeknek az ID-knek az adott mappában mindig egyedinek kell lenniük. Az ID-k itt ugyanazt a szerepet töltik be, mint a lemezen a fájlnevek, azaz arra szolgálnak, hogy az URL objektumait azonosítsák.
  A */foo/bar URL*-ben, a foo a könyvtárobjektum ID-je, a bar pedig ebben a könyvtárban található objektum azonosító ID-je. A Zope-dokumentumoknak hagyományosan nincsenek kiterjesztései (mint például a ***.html*** vagy a ***.gif***); mivel a rendszer pontosan tudja, hogy az egyes azonosítókhoz milyen típusú objektum tartozik, egy ilyen kiterjesztés használata felesleges lenne.
- A cím szövegmező – az URL-ekben az objektum ID-je jelenik meg, de a Zope belső kezelőfelületein már az objektum címével dolgozunk (az objektum címének semmi köze a HTML <title> (cím) taghoz, amit továbbra is nekünk kell elkészítenünk).
- A fájlelem lehetővé teszi, hogy a helyi gépről a HTTP-feltöltés használatával állományt töltsünk fel. Ez azt jelenti, hogy a DTML-állományt a helyi gépen kell létrehoznunk, majd amikor ellenőrizni akarjuk, fel kell töltenünk a Zope-ra.

Ebben a példában most a ***testdoc*** ID-t írjuk be, és a ***Test document*** címet adjuk meg. Bár ezt a dokumentumot is hozzáadhatnánk már a ZODB-hez, így még üresen maradna. Ezért az ***Add and edit*** gombra kattintunk, így szövegmezőhöz jutunk, ahol rögtön alapértelmezett tartalmat is találunk:

```
<dtml-var standard_html_header>
<h2><dtml-var title_or_id></h2>
<p>
This is the <dtml-var id> Document.
</p>
<dtml-var standard_html_footer>
```

Ha akarjuk, ezt a szöveget a szövegablakban át is szerkeszthetjük. Mikor végeztünk, kattintsunk a ***Save changes*** (változások mentése) gombra a képernyő alján. Ez a gomb visszavisz bennünket a szerkesztőablakra, de beilleszt egy emlékeztetőt, ami megmutatja, hogy a dokumentumot mikor módosították utoljára.

A DTML használata

DTML-dokumentumunk igen hasonló a HTML-hez, eltekintve attól, hogy néhány `<dtml-var>` kezdetű tagot tartalmaz.
A DTML tulajdonképpen programozási nyelv, ami sok mindent lehetővé tesz, de talán egyszerűbb (és jobb) úgy gondolnunk rá, mint egy igen erős kiszolgálóoldali beillesztő (include) szerkezetre. Ahogy a példadokumentum is bemutatja, a `<dtml-var>` alkalmazásával dokumentumunkba dinamikus értékeket illeszthetünk. Így a `<dtml-var standard_html_header>` a `standard_html_header`

© Kiskapu Kft. Minden jog fenntartva

© Kiskapu Kft. Minden jog fenntartva

ID-jű objektum tartalmát illeszti be, míg a `<dtml-var id>` a pillanatnyi dokumentum azonosítóját helyezi a tartalomba. Akár ki is találhatnánk: a `<dtml-var title_or_id>` a címet vagy az azonosítót jeleníti meg aszerint, hogy címet megadtunk-e vagy sem.
Mikor a Zope egy `<dtml-var standard_html_header>`-kifejezést talál, a `standard_html_header` azonosítójú objektumot először a pillanatnyi könyvtárban keresi. Ha létezik ilyen objektum, a `<dtml-var>` tagot ennek látható tartalmával helyettesíti. Ha nem létezik, a Zope a keresést a felsőbb könyvtárakban is megismétli. Így az egész szerkezetet végigjárja, míg végül el nem ér a saját könyvtárig.
Ez azt jelenti, hogy az önműködően beillesztett fejlécek és láblécek nemcsak a használt `<dtml-var>` tagoktól függenek, hanem DTML-dokumentumunk helyétől is. Ezt az igen lényeges és hasznos jelenséget, miszerint az objektum objektumhierarchiában elfoglalt helye befolyásolja a kimenetet, a Zope világában szerzeményezésnek (acquisition) nevezik. A szerzeményezés révén minden egyes könyvtárhoz más és más fej- és láblécet rendelhetünk. Ha dokumentumunkat az egyik könyvtárból a másikba helyezzük, megváltoztatjuk a Zope által használt keresési utat, amelyet a fejlécek és láblécek, illetve más objektumok azonosításához használ. Az objektum viselkedése tehát a Zope-ban nemcsak a meghatározástól függ, hanem az objektumhierarchiában elfoglalt helyétől is. Ezért a szerzeményezést gyakran írják le a biológiából ismerős „nature vs. nurture” (öröklött vagy tanult) vita megfelelőjeként: az objektum meghatározását tekinthetjük „öröklött tulajdonságnak” (nature), míg az objektumhierarchiában elfoglalt helyét „tanult képességnek” (nurture).
A dokumentum pillanatnyi tartalmának megtekintéséhez a keret tetején kattintsunk a ***View*** fülre. A dokumentumtartalmat pontosan úgy fogjuk látni, ahogy a felhasználó látná. Ha a dokumentumot ismét szerkeszteni szeretnénk, a legegyszerűbb, ha a bal felső sarokban található ***Root folder*** hivatkozásra kattintunk, kiválasztjuk a ***testdoc*** objektumot, majd a szövegmezőt tetszés szerint átszerkesztjük.
Ha a DTML szerkesztése közben hibát vétünk, nem árt tudni, hogy a Zope végtelen visszalépési lehetőséggel rendelkezik. Kattintsunk az ***Undo*** (visszalépés) fülre a DTML-szerkesztőablak jobb felső részében, és válasszuk ki azt a változatot, amelyikhez vissza szeretnénk térni. A végtelen visszalépés az egyik kedvenc szolgáltatásom a Zope-ban, nemcsak azért, mert lehetőséget ad hibáim kijavítására, hanem mert a nem programozók számára is könnyen elérhető, és bármilyen típusú objektummal működik.

Más DTML-eljárások

Számos DTML-tag létezik, valamennyit a dtml-kezdet különbözteti meg. A legtöbb DTML-tag egy vagy több értéket is elfogad, ahol minden érték az alapértelmezett működést változtatja meg egy kicsit. Példadokumentumunkat például átalakíthatjuk, hogy a következő egyszerű tagot használja:

```
<dtml-var expr>:
<p><dtml-var expr="5+10"></p>
```

A fenti kódban az `expr` egyszerű Python-kifejezés. A Zope kiértékeli a kifejezést, és az eredményt a dhtml-tag tag helyére illeszti. Egyszerű összeadás helyett valami érdekesebbet is kipróbálhatunk, például a `string` modulból használhatjuk a nagybetűsítő (capitalize) eljárást (ami önműködően importálódik a Zope-ba), ha az értékén belül nevezzük meg:

```
<p>
<dtml-var expr="_.string.capitalize(·abc·)">
</p>
```

Figyeljük meg, hogy a `string.capitalize()` függvényt nem hívhatjuk meg közvetlenül, csak a `_.string.capitalize()` -t.
Ugyan a DTML lehetővé teszi, hogy a `string`-modult ilyen módon használjuk, de soha nem lesz rá szükségünk, hogy közvetlenül hívjuk meg, olyan sok hasznos karakterkezelő függvényt találunk a DTML-be építve.
A DTML tervezőknek és nem programozó felhasználóknak készült azért, hogy saját dinamikus tartalmukat a kiszolgálóoldali ***include*** állományok gyenge írásmódjával és a kevés leírással való bajlódás nélkül készíthessék el. Természetesen egy nem programozónak a DTML-t nem olyan könnyű megtanulni, mint egyesek gondolnák, de minden bizonnyal sokkal könnyebb, mint egy teljes értékű programnyelvre tanítani meg őket. Ráadásul a Zope a DTML-dokumentum mentésekor bizonyos fokú hibaellenőrzést is végez, ami segít elkerülni néhány, a futásidejű nyelveknél gyakori hibát.

Összefoglalás

Ebben a hónapban villámlátogatást tettünk a Zope világának berkeiben, megismerhettük a Weben keresztüli szerkesztést, a karbantartó felületet, a szerzeményezést, és egy egyszerű DTML-dokumentumot is szemügyre vehettünk. A következő hónapban a Zope termékeit ismerjük meg – megnézzük, miképpen kell őket letölteni és telepíteni, illetve hogyan írhatjuk meg a saját változatainkat.

Reuven M. Lerner (reuven@lerner.co.il) kisebb webes és internetes módszerekkel foglalkozó tanácsadó cég tulajdonosa és vezetője. A cikk megjelenésének időpontjában valószínűleg már végleg elkészült Core Perl című könyvével, melyet idén jelentet meg a Prentice-Hall. Az ATF honlapon érhető el (➲ http://www.lerner.co.il/atf/).

Kapcsolódó címek

A nyílt forráskódú Zope-alkalmazást a Zope Corporation, régebbi nevén a Digital Creations kezeli és írja. Bővebb tájékoztatást a Zope Corporation honlapján találunk a ➲ http://www.zope.com címen.

A Zope hivatalos fejlesztői közösségének oldala a ➲ http://www.zope.org címen érhető el. Egyúttal ez az a honlap, ahonnan a legújabb változatokat letölthetjük, megvitathatjuk felmerült kérdéseinket, letölthetjük a Zope-termékeket, illetve tagjává válhatunk a Zope fejlesztőközösségnek.

Beehive Book of Zope c. könyve, melyet a No Starch Press és a Linux Journal Press közösen adtak ki, kitűnő bemutató írás Zope-programozóknak és -felhasználóknak egyaránt. A teljes körű ismertetés a következő havi Linuxvilágban jelenik meg.