

# A BLOG PÉLDAALKALMAZÁS

A 2006. február 9-10-én „Webes megoldások ASP.NET 2.0 alapokon” címen megrendezett előadássorozat moduljainak prezentációiban demóként megtalálható Blog rendszert azzal a céllal készítettük, hogy egy komplett alkalmazáson keresztül tudjuk bemutatni az ASP.NET 2.0 újdonságait. Éppen ezért próbáltunk egy teljes rendszert megvalósítani, de ennek ellenére az elkészült Blog alkalmazást (melynek a teljes [forráskódja](#) is letölthető a [devPORTAL](#)-ról) inkább szánjuk példa-, mint mintaalkalmazásnak.

Az elkészült példaalkalmazáson keresztül szeretnénk szemléletesen bemutatni az ASP.NET 2.0 platform olyan újdonságait, melyek drasztikusan lerövidítik a webalkalmazások fejlesztésének idejét azáltal, hogy rengeteg olyan megoldást kapunk készen, amit eddig mindig saját kezűleg implementálni kellett.

A blog rendszer specifikációja szerint lehetőséget biztosít blogok írására és olvasására az arra jogosult felhasználóknak. A blogokat mindenki olvashatja, de blogot írni csak a „Blogger” szerepkör tagjai tudnak, miután bejelentkeztek a megadott nevükkel és jelszavukkal. Hogy mely regisztrált felhasználók tagjai az előbb említett „Blogger” csoportnak, azt az „Admin” csoportba tartozó adminisztrátorok tudják beállítani. A felhasználók, szerepkörök, jogosultságok kezeléséhez és menedzseléséhez az ASP.NET 2.0 beépített Membership és RoleManagement szolgáltatásait vesszük alapul, és kihasználunk minden, a felhasználó-kezeléssel kapcsolatos beépített felületi vezérlőelemet a bejelentkezéstől a regisztráción át a jelszóváltoztatásig. A regisztrált felhasználók alap adatain kívül nyilvántartunk róluk egy egységes profil információt is, amiben egyszerű (szöveg), és összetett (kép) típusú tartalmat is tárolunk.

Az egységes webhely kialakításakor az újonnan megjelent mester- és tartalom-oldal koncepcióra építkeztünk, felhasználva a Microsoft által készített [Small Business Design Template](#)-et. A megoldás során többször találkoztunk olyan problémával, melyek igényelték egy saját PageBase osztály megvalósítását, ami logikailag a mester oldal és a tartalom oldal között helyezkedik el, és olyan dolgokat tartalmaz, amit minden oldalon meg kellene valósítani, viszont a mester oldalban ezt nem tudjuk megtenni, mert ott nincs rá lehetőség.

Annak érdekében, hogy szemléltethessük a webkijelzők használatának egyszerűségét és praktikusságát, az „Akik nálunk bloggolnak”, „A szerző kategóriái”, illetve a „Havi bontás” című modulokat egy-egy webkijelzőként valósítottuk meg. Így a példa bemutatja, hogy minimális kódolással milyen személyre szabhatósági lehetőségeket nyújt az új keretrendszer a bejelentkezett felhasználók számára.

A stílusokat két, jól elkülöníthető témába szerveztük, a legtöbb stílusbeállítást visszavezettük CSS-be. Így meglehetősen könnyű megoldással tudtunk a Blog rendszerünknek két különböző színvilágot kölcsönözni az ASP.NET 2.0 témáinak támogatásával.

A dinamikus adatok SQL adatbázisból jönnek, a felületi leképzés során mindig a tervezői nézetben alkalmazható összekattintgatós megoldással, a beépített varázslók használatával készítettük el a megjelenítéseket. A példa során használunk SQL és Object DataSource-ot is adatforrásként. Mivel az ObjectDataSource előállító varázsló csak olyan osztályokat fogad el adatforrásként, amik az App\_Code könyvtárban vannak, ezért a Membership és RoleManagement API-k ObjectDataSource-on keresztül történő használatához szükségünk volt egy csomagoló osztály elkészítésére (userManager.cs).

A providerek használatának szemléltetéséhez készítettünk egy olyan osztályt, amivel biztonságosan lehet QueryString-et felolvasni, és BuildProvider alapokon működik.

A rendszer menüje szintén az újonnan megjelent navigációs lehetőségekre támaszkodik, mind a logikai webhely-térkép, mint adatforrás, mind pedig a felületi megjelenítés, a menü vezérlőelem kapcsán. Ezen túlmenően azt a tulajdonságát is kihasználtuk a beépített navigációs lehetőségeknek, hogy képesek vagyunk egy-egy menü-részt csak bizonyos felhasználói csoportok számára definiálni, azaz a menü dinamikusan az aktuális felhasználói szerepkörhöz igazodva jelenik meg, és csak azokat a menüpontokat tartalmazza, amiket az aktuális felhasználó csoporttagságai számára elérhetővé tesznek.

Hogy bemutassuk annak szépségét, hogy mindent mindennel tudunk kombinálni, és ettől nem lesz nehezebb a megvalósítás az életünk, alkalmazásunkat minimális szinten lokalizáltuk is. Nevezetesen az előbb említett menüt készítettük el magyar és angol nyelvre egyaránt. Így egy sima statikus menü helyett képesek voltunk egy dinamikus, szerepkörhöz és nyelvhez adaptálódó menüt megvalósítani kódsor írása nélkül, mindent pusztán deklaratívan megadva.