

Bevezetés

A könyv segítségével megtanulhatjuk, hogyan írhatunk webes alkalmazásokat a Microsoft HTTP-kéréseket feldolgozó keretrendszere, az ASP.NET 2.0 segítségével. Az 1990-es években a legelső webhelyek megjelenése óta a webfejlesztés hosszú utat járt be, a fejlesztőeszközök területén igen széles választékkal rendelkezik. A HTTP-kérések kezelésének tekintetében az elmúlt néhány évben az ASP.NET az egyik leg-robusztusabb, legmegbízhatóbb és sokoldalú keretrendszerre fejlődött.

A Visual Studioval együtt az ASP.NET rengeteg hasznos szolgáltatás révén megkönnyíti a webfejlesztők munkáját. Például a Visual Studio több, hasznos projektsablonnal rendelkezik, amelyek segítségünkre lehetnek webhelyünk fejlesztése során. A Visual Studio különböző fejlesztési módokat is támogat, többek között az IIS közvetlen használatát a fejlesztés alatt álló webhely tesztelésére, illetve a beépített webkiszolgáló használatát vagy a webhely FTP-kapcsolaton keresztüli fejlesztését. A Visual Studio hibakereső programja segítségével a kód kritikus területein végiglépkedve kereshetünk esetleges problémákat. A Visual Studio Designer hatékony felhasználói felület fejlesztését teszi lehetővé, a vezérlőelem-elemeket vászonra helyezve megtekinthetjük megjelenésüket. Ez csak néhány példa az ASP.NET-keretrendszerbe épített szolgáltatások közül, amelyek rendelkezésünkre állnak, ha a keretrendszert Visual Studioval használjuk.

Bár az ASP.NET és a Visual Studio a webes alkalmazások fejlesztésének kiváló eszközei, a Microsoft-platfomon a webfejlesztés nem mindig volt ilyen. Az ASP.NET 2.0 létrejöttéhez majdnem egy évtizedre volt szükség.

Az ASP.NET 2.0 fejlesztése

Körülbelül 1993-ig világszerte igen kevés webkiszolgáló működött. Az első webkiszolgálók nagy része egyetemeken és kutatási központokban üzemelt. Az 1990-es évek elején a webhelyek száma drámai növekedésnek indult. Ha az 1990-es évek elején használtuk a webet, biztosan találoztunk a webhely úttörők HTML-oldalaival vagy fényképgyűjteményekkel, amelyek GIF- vagy JPEG-fájlok hivatkozásait tartalmazták. Akkoriban még nem létezett a Google, a Yahoo és az MSN Search sem. A webhely megtekintésének egyetlen módja az volt, ha ismertük a webhely URL-jét (**Uniform Resource Locator**, egységes forrásazonosító), vagy valaki más oldalán találtunk hivatkozást az adott webhelyre.

Ha a

`http://www.somesite.com`

URL-hez hasonló hivatkozást gépeltünk a böngésző navigációs ablakába, a kérés útválasztók labirintusán keresztül végül valahol megjelent egy kiszolgálón. Az első webkiszolgálók UNIX-gépeken működtek. Feladatuk az volt, hogy betöltsék a HTML-fájlt és azt visszaküldjék a kérelmező félnek (ami lehetett böngésző, például a Mosaic).

A CGI (**Common Gateway Interface**, közös átjáró interfész) megjelenésével a böngészők az interaktív webes alkalmazások standard interfészévé léptek elő. Az egyszerű, statikus HTML-dokumentumokat kiszolgáló webkiszolgáló bizonyos célokra (például belső hivatkozásokkal ellátott szótár létrehozására) kiválóan alkalmas, az összetettebb alkalmazások megkövetelik a felhasználó és a kiszolgáló közötti kommunikációt.

És a CGI itt jut fontos szerephez. A szabványos grafikus vezérlőelemeket megjelenítő HTML-címkék segítségével a CGI-alkalmazások a kérésekre dinamikusan válaszolnak. Más szóval, a CGI-alkalmazások a kérés állapotától és az alkalmazástól függően különböző kimenetet állítanak elő, ezzel is segítik az interaktív alkalmazásokat. Például egy CGI-alkalmazás képes arra, hogy megvizsgálja a bejövő kérést, és meghatározza, hogy a felhasználó adott információrészt (történetesen egy termékkódot) keres. A CGI-alkalmazás adatbázis-keresés segítségével megkeresi a terméket, és a termék leírását tartalmazó HTML-t visszaküldi az ügyfélnek.

Amikor világossá vált, hogy a web az informatika fontos része, a Microsoft is beállt az üzletbe. Piacra dobta az ISAPI-t (**Internet Services Application Programming Interface**, internetszolgáltatások alkalmazásprogramozási interfésze) és a HTTP-kéréseket figyelő programot: az IIS-t (**Internet Information Services**, internet információs szolgáltatások). Az első UNIX-webkiszolgálók (a klasszikus UNIX-modellhez igazodva) minden új HTTP-kérés számára új folyamatot indítottak. Ez rendkívül költséges. A Microsoft webes stratégiája DLL-eken alapul. Egy HTTP-kérésre válaszolva sokkal gyorsabb betölteni egy DLL-t, mint elindítani egy teljesen új folyamatot.

Microsoft-platfomon az IIS a 80-as porton figyeli a HTTP-kéréseket. Az IIS néhány kérést közvetlenül kiszolgál, míg más kéréseket az ISAPI-kiterjesztések DLL-jeihez küldi, és a kéréseket azok hajtják végre. A többi esetben az IIS a fájlkiterjesztést leképezi egy specifikus ISAPI DLL-re. Sok ISAPI DLL előre telepíthető a Windows operációs rendszerrel együtt. Az IIS bővíthető, és a különböző kiterjesztéseket bármely ISAPI DLL-re leképezhetjük, még saját DLL-jeinkre is. Az IIS-sel és ISAPI-val működő webhely készítése során a fejlesztők ISAPI DLL-eket alkalmaznak. Ezek a DLL-ek észlelik a kérést, elemeire szedik, és a kérésre válaszolva visszaküldenek valamit (rendszerint HTML-t) az ügyfélnek.

Noha az IIS/ISAPI platform a webes alkalmazások készítésének rugalmas és funkcionálisan gazdag módját biztosítja, ez a platform sem tökéletes. Az ISAPI DLL-ek C++-ban készülnek, ezért a DLL-eknek a C++-programozás csapdáit kell kikerülniük (olyan gyenge pontokat, mint például a rossz mutatók hivatkozásának törlése, a memória felszabadításának figyelmen kívül hagyása, valamint a hagyományosan hosszadalmas fejlesztési ciklusok). Az ISAPI DLL-ekkel kapcsolatos másik probléma, hogy manapság egyre nehezebb C++-programozókat találni.

A klasszikus ASP

A Microsoft annak érdekében, hogy a webfejlesztést még elérhetőbbé tegye Microsoft-platformon, kifejlesztette az ASP-t (**Active Server Pages**, aktív kiszolgálóoldalak). A klasszikus ASP alapja az, hogy egyetlen ISAPI DLL, az ASP.DLL értelmezi az ASP-kiterjesztéssel rendelkező fájlokat (például a MYSITE.asp-t). Az ASP-fájlok HTML-kódot, illetve esetenként szkriptkódot tartalmaznak, amelyet a kiszolgálón kell végrehajtani. Az ASP ISAPI DLL szükség szerint végrehajtja a szkriptkódot, és az ASP-fájlból található HTML-t visszaküldi az ügyfélnek. A szkriptkód rendszerint COM-objektumokat hív, és azok végzik el a piszkos munkát (például megkeresik az elemeket az adatbázisban, és az eredmények alapján összeállítják a kimenetet), miközben az oldal megjelenését az ASP-fájl HTML-kódja határozza meg.

Az ASP, mivel egy jóval szélesebb körben használt programozási nyelvet (Visual Basic, VBScript) biztosított, rengeteg új programozó vonzott, de mégsem jelentett mindenre megoldást. A klasszikus ASP hátrányai közül érdemes néhányat megemlíteni:

- a felhasználói felület kódjának keveredése a programozási logikával,
- teljesítményproblémák az *IDispatch* interfésznek köszönhetően,
- az állapotkezelés (munkamenet-állapot és alkalmazásállapot) következetlen eszközei,
- ad hoc biztonsági modell.

A lista messze nem teljes, de kiemeli a klasszikus ASP-vel kapcsolatos legfontosabb problémákat. Ezek a problémák hívták életre az ASP.NET-et.

ASP.NET 1.0 és 1.1

A Microsoft .NET-keretrendszere a Microsoft-platform programozásának teljesen új módját vezeti be. A Microsoft fejlesztőket elsősorban a szál- és a memóriakezelés foglalkoztatja (ami nagyjából az API programozási modellt takarja). Ez a modell, a webfejlesztést is beleértve, a programozás minden területén elterjedt, és nagyon megnehezítette a programozók munkáját.

A .NET a kezelt típusok fogalmára épít. A klasszikus Windows-kódot (és webkódot) készítő fejlesztők az osztályokat C++ vagy Visual Basic segítségével írták. A típusok több szempontból is hasonlítanak a C++ osztály fogalmához, mivel a típusok is önálló funkcióval rendelkező állapotegységeket jelentenek. Azonban a hasonlóságnak itt vége is szakad. Míg korábban a fejlesztőre hárult az osztálypéldányok kezelésének feladata, a típusokat teljes mértékben a .NET futásidejű szolgáltatásai, azaz a CLR (**Common Language Runtime**, közösnyelvi futtatórendszer) kezeli. Mivel a CLR magára vállalja a szál- és memóriakezelés feladatát, a fejlesztők sokkal inkább a konkrét alkalmazásra összpontosíthatnak (és nem kell téves mutatók, memóriaszivárgások és megmagyarázhatatlan összeomlások után nyomozniuk).

Az ASP.NET bevezeti a futásidejű szolgáltatásokat, valamint a webfejlesztés magas szintű továbbfejlesztését jelentő, kiválóan megtervezett osztálykönyvtárat. A klasszikus ASP beépült az IIS/ISAPI architektúrába annak megfontolása nélkül, hogy a korai tervezési döntések a későbbiekben milyen hatással lesznek a fejlesztők munkájára. Mostanra, egy kis idő elteltével már világosan látszanak az ASP.NET hibái.

Az ASP.NET a kezdetektől fogva a HTTP-kérések kezelésének bővíthető, sokoldalú megoldásának készült. Az ASP.NET befolyásolja az IIS működését, mivel az ASP.NET-szolgáltatások kéréseit leképezi egy ISAPI DLL-re. Ez a DLL az ASPNET_ISAPI.DLL. A végrehajtás innen az ASP.NET dolgozófolyamatához (IIS 5-ben az ASPNET_WP.EXE programhoz vagy IIS 6-ban a W3WP.EXE programhoz) kerül. Az alapvető kérelmfeldolgozást a dolgozófolyamat kezelt típusai végzik. A vezérlés a csatornához csatlakoztatott osztályok között vándorol, néhány osztályt a Microsoft és/vagy harmadik fél biztosít, néhányat pedig a fejlesztő. Az ASP.NET már alapjaiban is a webes alkalmazások készítésének átfogó keretrendszere. A keretrendszer minden része együttműködik a kérések kezelésében. Ezzel szemben a klasszikus ASP.NET-szkriptkód strukturálatlan volt, és az alkalmazás kódlogikája is ad hoc jellegű volt.

Az ASP.NET 1.0 és 1.1 rengeteg új szolgáltatással rendelkezett. Íme néhány:

- objektumorientált keretrendszer az alkalmazások definiálásához,
- a felhasználói felületbeli deklarációk (HTML) és az alkalmazáslogika elkülönítése,
- az alkalmazáslogika végrehajtásához szükséges lefordított kód,

- konfigurálható munkamenetállapot-kezelés,
- beépített adat-gyorsítótárzás,
- beépített tartalom-gyorsítótárzás,
- a felhasználói felület jól definiált komponensekre bontásható architektúrája,
- az adatformázás kezelésének magas szintű összetevői (rácsok, listák, szövegmezők),
- beépített nyomkövetés és diagnosztika,
- beépített szolgáltatás a felhasználói bevitel ellenőrzésére,
- könnyen kezelhető egyedi hitelesítési mechanizmus,
- szilárd integráció az ADO.NET-tel (a .NET és az adatbázisok története),
- a webszolgáltatások kiváló támogatása,
- a komponensobjektum-modell figyelmen kívül hagyása;
- bővíthető futószalag, amelynek több pontján is beérkezhetnek a kérések.
- Az ASP.NET 1.0 jó alap volt a webfejlesztés és a Microsoft-platform felé forduló fejlesztőknek.

ASP.NET 2.0

És ezzel elérkezett az ASP.NET 2.0 ideje. Az ASP.NET 2.0 az ASP.NET 1.0 és 1.1 verzióira épül, és az ASP.NET 1.0 szolgáltatásain kívül sok új lehetőséget biztosít. Az újdonságok közül néhány:

- főlapok és felszínek,
- deklaratív adatkötés,
- szolgáltatói mintamodell,
- új gyorsítótár-szolgáltatások,
- tagsági vezérlőelemek,
- személyre szabási vezérlőelemek,
- webkijelzők támogatása,

- programozható konfiguráció,
- felügyeleti eszközök,
- új fordítási modell.

Ezekon kívül az ASP.NET 1.0/1.1 összes szolgáltatása is a rendelkezésünkre áll. Ezek az új szolgáltatások még ellenállhatatlanabb platformmá teszik az ASP.NET-keretrendszert egy webhely létrehozásához. Az ASP.NET 2.0 bemutatásakor mindezeket a szolgáltatásokat alaposan megvizsgáljuk.

Néhány szó a .NET-futtatórendszeréről

Az ASP.NET 2.0 a Microsoft CLR-re épül. Kezdetben a Windows-programozás az operációs rendszerrel való rendkívül szoros együttműködést jelentette. Például egy ablak megjelenítéséhez elég hosszú C-kódot kellett írni. Ezen kívül a Windows igen kezdetleges összetevő technológiával rendelkezett, nyers dinamikus csatolású könyvtárakkal. A DLL-ek (**D**ynamic **L**ink **L**ibrary, dinamikus csatolású könyvtár) a rendszerek dinamikus felépítésének szükséges kellei, azaz, az alkalmazások különböző bináris komponenseit állítják össze. Azonban pusztán a DLL-ek segítségével nem lehet a rendszereket megbízhatóan felépíteni, elsősorban azért, mert egyetlen komponens (DLL) több verziójának kezelése nem egyszerű feladat.

Az 1990-es évek közepén megjelent COM (**C**omponent **O**bject **M**odel, komponensobjektum-modell) az egyes összetevők különböző verziójának kezelését segíti. Az ügyfelek és a komponensek együttműködésének szigorú szabályozásával a COM a különböző bináris összetevőkből készülő alkalmazások összeállításához megfelelő technológiát jelentett. A COM néhányszor zsákutcába futott. A zsákutcák a nagyobb rendszerek fejlesztése során váltak nyilvánvalóvá.

A COM az emberre bízta a dolgok együttműködéséhez szükséges szabályok betartását. Például a COM előírja, hogy egy programozott interfész közzététele után nem módosulhat. Egy közzétett COM-interfész módosítása az után, hogy az ügyfelek megkezdik annak kódolását, egészen biztosan hibát okoz a rendszerben. Továbbá, a COM az erőforrás-kezelés terén kissé bárgyú szabályokra támaszkodott. A COM-nak a kegyelmdöfést valószínűleg az egyetlen típusrendszerek adták meg, a COM a különböző adattípusokat különbözőképpen jelenítette meg a fejlesztők három különböző kategóriájának: a C++, a Visual Basic és a szkriptírásképzőknél. A különböző adattípusrendszerek rendkívül kényelmetlenné tették a különböző nyelvek felhasználásával készült rendszerek építését. A feladat nem kivitelezhetetlen, de a fejlesztőknek nagyon körültekintőnek kellett lenniük a komponensek közötti hívások kezelése során.

A .NET és a CLR a COM 1990-es évek végére nyilvánvalóvá vált zsákutcáinak kiküszöbölésére készült. Amikor úgy határozunk, hogy megvásároljuk a .NET-futtatórendszert, képzeljük azt, hogy kódunkat futás közben egy szállodai lakosztályban helyezzük el. Például a .NET-futtatórendszer futás közben betölti és kezeli kódunkat. A nyilvánvaló memóriaszivargások már csak rossz emlékek, mert a futtatórendszer szükség szerint hulladékgyűjtést is végez. A tömbök túlcímzésének problémája is a múlté, mivel a .NET-futtatórendszer felügyeli a memóriát, és tudja, hogy mikor mi nincs rendjén. Ezen kívül a .NET-futtatórendszer új biztonsági modellt is magában foglal, amely megnehezíti a .NET alapú szoftverek illetéktelen hozzáférését. És végül a .NET-futtatórendszer bevezeti az új csomagolási és telepítési modellt, a .NET-szerelvényeket, amelyek kötelezővé teszik az összetevők verziókezelését.

Az ASP.NET a .NET-futtatórendszeren alapul. Mint azt a következő fejezetekben látni fogjuk, az ASP.NET teljes mértékben a CLR támogatása alatt fut. Miután az IIS átadja a HTTP-kérést az ASP.NET-nek, a kérés az ASP.NET-futószalagra kerül. A folyamat során a kérés több helyen elfogható, és a lehetőségek széles választéka áll rendelkezésünkre, hogy kivallassuk a kérést és módosítsuk a választ, mielőtt az elhagyná az ASP.NET-futtatórendszert. A HTTP-kéréseket feldolgozó gépezet és a vállalat tartományobjektumai közötti COM-rétegnek ezzel befellegzett. A .NET alatt futó tartományobjektumokat a jobb teljesítmény és a szigorúbb biztonság érdekében a kéréseket feldolgozó futószalaghoz lehet kapcsolni. Továbbá, mivel a .NET-összetevők megegyeznek az átadott adattípusokról, többé nem merülnek fel bizarr adatkonverziós problémák (mint annak idején a klasszikus ASP-ben).

Az ASP.NET-alkalmazások készítése során .NET-szerelvényeket fejlesztünk majd, az idő javarészában implicit módon, de néha explicit módon is. Az ASP.NET webalkalmazás-keretrendszer oldalának vizsgálata közben elmélyülhetünk a .NET-futtatórendszer tanulmányozásában is. Az ASP.NET-alkalmazásokban használt osztályok igen gyakran hasonlóak a parancssori alkalmazások, a Windows-alkalmazások vagy akár a komponenskönyvtárak osztályaihoz.

A könyv használatáról

A könyv célja, hogy megismertesse az olvasóval az ASP.NET 2.0 fejlesztést. Az egyes részek értelmezhető formátumban meghatározott ASP.NET-szolgáltatásokat mutatnak be, *példákkal*. A lépésenkénti útmutatók azonnal használható eredményeket adnak. A fejezetekben megtaláljuk az ASP.NET főbb szolgáltatásait, amelyeket tömör és könnyen másolható példák illusztrálnak. Igyekeztem a szolgáltatásokat jól szemléltető, de nem nehézkes példákat összegyűjteni. Az ASP.NET-szolgáltatások példákön keresztüli bemutatásán kívül az egyes szolgáltatások gyakorlati alkalmazásaival is találkozni fogunk. Tehát ezeket a technikákat a való életben is alkalmazhatjuk majd.

Kinek szól a könyv?

A könyv alapvetően fejlesztőknek szól:

- **Az ASP.NET-tel való ismerkedést most kezdő fejlesztőknek.** A könyv elegendő háttérinformációt biztosít a webfejlesztés történelméről még az asztali alkalmazást fejlesztők számára is.
- **Az ASP.NET 1.x-ről vagy a klasszikus ASP-ről migráló fejlesztőknek.** A könyv ismerteti az ASP.NET 2.0 és az ASP.NET 1.x különbségeit. A fejezetek az ASP.NET és a klasszikus ASP különbségeit részletező hivatkozásokat is tartalmaznak.
- **Az ASP.NET-ismereteket emészthető darabokban befogadni kívánó fejlesztőknek.** A fejezetek egymástól független egységek. A fejezeteket nem kell meghatározott sorrendben olvasni. A könyv e nélkül is értékes információs forrásnak bizonyul. Minden egyes fejezet többé-kevésbé önálló (az első fejezet kivételével, amely a webes alkalmazások alapjait részletezi; ha még nem merészkedtünk tovább az asztali alkalmazások fejlesztésén, tanácsos ezzel a fejezettel kezdeni). A kiszolgálóoldali vezérlőelemekről szóló fejezeteket (3–5. fejezet) érdemes, de nem feltétlenül szükséges egymás után áttanulmányozni.

A könyv felépítése

A könyvben az egyes fejezeteket nagyrészt egymástól függetlenül elolvashatjuk. Az 1. fejezet (a webes alkalmazások alapjait ismerteti), valamint a kiszolgálóoldali vezérlőelemeket részletező 3–5. fejezetek kivételével, amelyeket együtt érdemes elolvasni, a fejezetek egy-egy meghatározott ASP.NET-szolgáltatást bemutató, önálló információblokkok.

A kezdetek

Ha idáig eljutottunk, akkor készen állunk arra, hogy kódoljunk egy kicsit. Mielőtt elkezdenénk, győződjünk meg arról, hogy gépünkre telepítettük a Visual Studio 2005-öt. Ha a fejlesztői környezetet telepítettük, biztosak lehetünk abban, hogy egyidejűleg a .NET-futtatórendszer segítségét is telepítettük.

Az első néhány példához kizárólag szövegszerkesztőre lesz szükségünk, és egy működő IIS-telepítésre. Bemelegítésként néhány alapszintű példa segítségével illusztráljuk az ASP.NET objektumorientált természetét, valamint a fordítási modellt. Továbbá bemutatjuk, hogy az ASP.NET pontosan mit csinál egy-egy kérés kezelésekor.

Ez remek alkalom lesz arra, hogy magas szintről megvizsgáljuk az ASP.NET architektúráját. A webürlapok programozásával folytatjuk, majd a Visual Studiót hívjuk segítségül a kód megírásához. (Látni fogjuk, hogy mennyivel könnyebb a dolgunk!).

A webürlap-fejlesztés alapjainak elsajátítása után az ASP.NET fennmaradó részét darabokra szedjük. Példák segítségével vizsgáljuk meg az ASP.NET-szolgáltatásokat, például a kiszolgálóoldali vezérlőelemeket, a tartalom gyorsítótárazását, az egyedi kezelők írását, a kimenet és az adatok gyorsítótárazását, a hibakeresést és a diagnosztikát, és végül az ASP.NET webszolgáltatás támogatását.

Honnan kezdjük az olvasást?

A könyv segítségével több fontos terület tudásanyagát is elsajátíthatjuk. A könyvet akkor is érdemes fellapozni, ha most kezdjük a webes programozással való ismerkedést, vagy egy másik webfejlesztési platformról váltunk Microsoft-platfomra. A következő táblázat segítségével megkereshetjük a könyv számunkra megfelelő kiindulópontját.

Ha	Kövessük az alábbi lépéseket
most ismerkedünk	
a webes fejlesztéssel	<ol style="list-style-type: none"> 1. Telepítjük a példaprogramokat! 2. Olvassuk el az 1. és a 2. fejezetet, és végezzük el a példákat! A fejezetek segítségével elsajátíthatjuk a webfejlesztés alapjait. Az első két fejezetben megismerkedtünk az ASP.NET-tel és a Visual Basic-kel is. 3. A könyv többi fejezeteit igényeink szerint olvassuk el!
most ismerkedünk	
az ASP.NET-tel és a Visual Studio-val	<ol style="list-style-type: none"> 1. Telepítjük a példaprogramokat! 2. Olvassuk el a 2. fejezetet, és végezzük el a példákat! A fejezet az ASP.NET és a Visual Basic használatának alapjait ismerteti. 3. A könyv többi fejezeteit igényeinknek megfelelően olvassuk el!

Ha	Kövessük az alábbi lépéseket
----	------------------------------

migrálunk

- | | |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ASP.NET 1.x-ről
vagy klasszikus
ASP-ről | <ol style="list-style-type: none">1. Telepítsük a példaprogramokat!2. Lapozzuk végig az első két fejezetet, amelyek segítségével áttekintést kaphatunk a webfejlesztésről Microsoft-platfóron és Visual Basic 2005-ben.3. Igényeink szerint összpontosítsunk a 3–20. fejezetekre! Már járatosak vagyunk ezekben a témákban, csak arra van szükségünk, hogy lássuk, a meghatározott szolgáltatások miben különböznek az ASP.NET 1.x-ben és az ASP.NET 2.0-ban. Időnként az is előfordulhat, hogy az ASP.NET 2.0 egyik újdonságára vagyunk kíváncsiak. |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
-

**hivatkozásként
használjuk
a könyvet**

- | | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| miután elvégeztük
a gyakorlatokat | <ol style="list-style-type: none">1. A tárgymutató vagy a tartalomjegyzék segítségével keressük meg az adott témával kapcsolatos információkat!2. Olvassuk el a fejezetek végén található Gyorsreferenciákat, ahol a fejezetben tárgyalt szintaxis és a technikák rövid áttekintését találjuk. |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
-

A könyvben használt jelölések

A szöveg érthetőségének és követhetőségének megkönnyítésére jelöléseket alkalmaztunk. Mielőtt elkezdenénk a könyvet olvasni, tanulmányozzuk át a következő listát, amely a könyvben található jelöléseket és azok rövid leírását tartalmazza, és kiemeli a könyv hasznos különlegességeit, amelyeket a későbbiekben alkalmazhatunk.

Jelölések

- A fejezetek elején az egyes fejezetek céljainak összegzését találjuk.
- Minden gyakorlat feladatok sorozatából áll. Az egyes feladatok egymás után végrehajtandó lépések sorozatát tartalmazzák.

- A „Tipp” megjegyzések az adott lépés sikeres végrehajtásával kapcsolatos további információkat vagy alternatív módszereket ismertetnek.
- A beírt szöveg **félkövéren** jelenik meg.

```
class foo
{
    System.Console.WriteLine("HelloWorld");
}
```

- Az utasítások gyakran alternatívákat is tartalmaznak, amelyek segítségével ugyanarra az eredményre jutunk. Például a Visual Studio projekthez új elemet a főmenüből vagy a Solution Explorerben jobb egérgombbal kattintva is hozzáadhatunk.
- A könyv példáinak túlnyomó többsége C# segítségével készült. Néhány fejezetben C#- és Visual Basic példákat is találunk. Ezeknek a példáknak a segítségével megvizsgálhatjuk, hogy ugyanazokat a programozási idiómákat különböző programozási nyelveken hogyan fejezhetjük ki.

Egyéb jelölések

- A könyvben széljegyzeteket és megjegyzéseket találunk, amelyek az adott témáról részletes információkkal szolgálnak. A széljegyzetek háttérinformációkat, tervezési ötleteket vagy az aktuális témával kapcsolatos egyéb információkat tartalmaznak. Ezekben a széljegyzetekben arról is tudomást szerezhethetünk, hogy a meghatározott szolgáltatások a korábbiakhoz képest miben különböznek az ASP.NET e verziójában.
- A fejezetek végén összegzést és Gyorsreferencia részt találunk. A Gyorsreferencia a fejezetben ismertetett feladatok végrehajtásához szükséges tömör emlékeztetőket tartalmazza.

Rendszerkövetelmények

A könyv gyakorlati feladatainak végrehajtásához szükséges rendszerkövetelmények a következők:

Megjegyzés: A könyv *nem* tartalmazza a Visual Studio 2005 szoftvert! A könyvhöz tartozó CD-ROM a gyakorlatok végrehajtásához szükséges példaprogramokat tartalmazza. A Visual Studio 2005 szoftvert külön meg kell vásárolni.



- Microsoft Windows XP Professional 2-es javítócsomaggal, Microsoft Windows Server 2003 1-es javítócsomaggal vagy Microsoft Windows 2000 4-es javítócsomaggal;
- Microsoft Internet Information Services (IIS) (a Windows operációs rendszer tartalmazza),
- Microsoft Visual Studio 2005 Standard Edition vagy Microsoft Visual Studio 2005 Professional Edition,
- Microsoft SQL Server 2005 Express Edition (a Visual Studio 2005 tartalmazza) vagy Microsoft SQL Server 2005,
- 600 MHz Pentium vagy kompatibilis processzor (1 GHz Pentium javasolt),
- 192 MB RAM (256 MB vagy több javasolt),
- SVGA (800×600 vagy nagyobb felbontású) monitor legalább 256 színnel (1024×768 16 bites High Color javasolt),
- CD-ROM vagy DVD-ROM meghajtó,
- Microsoft egér vagy kompatibilis mutatóeszköz.

Az SQL Server 2005 Express Edition konfigurálásához számítógépünkön rendszergazda jogosultságokra is szükség lesz.

A Microsoft Access használatáról

Az adatkötésekkel foglalkozó 13. fejezetben és az alkalmazásbeli adatok gyorsítótárzását ismertető 14. fejezetben is szükségünk lesz a Microsoft Access-re. Ha szeretnénk megtekinteni és módosítani az adatbázisokat, számítógépünkre telepíteni kell a Microsoft Access-t. Ha a Microsoft Office-t használjuk, előfordulhat, hogy a szoftvert már korábban telepítettük. A program nem igényel semmilyen speciális beállítást, és az adatbázisokat speciális lépések végrehajtása nélkül használhatjuk ASP.NET-alkalmazásainkban.

Példaprogramok

A könyvhöz tartozó CD-melléklet a C#-példaprogramokat tartalmazza, amelyeket a könyv gyakorlatainak végrehajtása során használunk majd. A példaprogramok segítségével nem kell arra időt pazarolnunk, hogy létrehozzunk a gyakorlat szempontjából lényegtelen fájlokat. A fájlok és a fejezetek lépésenkénti útmutatóinak segítségével gyakorlás közben tanulhatunk. Ez az új ismeretek megszerzésének könnyű és hatékony módja.

Megjegyzés: Ha szívesebben használjuk a Visual Basic példaprogramokat, akkor letölthetjük azok Visual Basic változatát is. További információkat „A Visual Basic példaprogramok telepítése” című részben találunk.



A C#-példaprogramok telepítése

Az alábbi lépések segítségével számítógépünkre telepíthetjük a C#-példaprogramokat, hogy azokat a könyv gyakorlati feladatainak végrehajtása során alkalmazni tudjuk.

Megjegyzés: A példaprogram-telepítő módosítja az IIS-t, tehát számítógépünkön rendszergazda jogok szükségesek a példaprogramok telepítéséhez.



1. A könyv belső borítójáról szedjük le a CD mellékletet, és helyezzük a CD-ROM meghajtóba!

Megjegyzés: A végfelhasználói licencszerződés automatikusan megnyílik. Ha a szerződés nem nyílik meg automatikusan, az asztalon vagy a Start menüből nyissuk meg a My Computer menüt.



2. Olvassuk át a végfelhasználói licencszerződést! Ha elfogadjuk a feltételeket, válasszuk a megfelelő lehetőséget, majd kattintsunk a Next gombra!
A könyvvel kapcsolatos opciók menüje jelenik meg.
3. Kattintsunk az Install Code Samples opcióra!
4. Kövessük a képernyőn megjelenő utasításokat!

Megjegyzés: Ha a gépen nincs IIS, a képernyőn üzenet tájékoztat arról, hogy a telepítő nem tud az IIS-hez kapcsolódni. Figyelmen kívül hagyhatjuk az üzenetet, és telepíthetjük a példaprogramok fájljait, de azok a példaprogramok, amelyek futtatásához szükség van az IIS-re, nem működnek.



A telepítő a példaprogramokat a számítógép következő mappájába másolja: C:\Microsoft Press\ASP.NET 2.0 Step by Step\

A telepítő aspnet2sbs névvel virtuális könyvtárat hoz létre az alapértelmezett webhely alatt. Az aspnet2sbs virtuális könyvtár alatt több webes alkalmazást is létrehoz. A beállítások megtekintéséhez nyissuk meg az Internet Information Services konzolt.

A Visual Basic példaprogramok telepítése

Az alábbi lépések segítségével számítógépünkre telepíthetjük a Visual Basic példaprogramokat, hogy azokat a könyv gyakorlati feladatainak végrehajtása során alkalmazni tudjuk.



Megjegyzés: A példaprogram-telepítő módosítja az IIS-t, tehát számítógépünkön rendszergazda jogok szükségesek a példaprogramok telepítéséhez.

1. Töltsük le a Visual Basic példaprogramok telepítőjét a könyv hálózati melléklet lapjáról:
<http://www.microsoft.com/mspress/companion/0-7356-2201-9/>
2. Indítsuk el a telepítőt!
3. Kövessük a képernyőn megjelenő utasításokat!



Megjegyzés: Ha a gépen nincs IIS, a képernyőn üzenet tájékoztat arról, hogy a telepítő nem tud az IIS-hez kapcsolódni. Figyelmen kívül hagyhatjuk az üzenetet, és telepíthetjük a példaprogramok fájljait, de azok a példaprogramok, amelyek futtatásához szükség van az IIS-re, nem működnek.

A telepítő a példaprogramokat a számítógép következő mappájába másolja: C:\Microsoft Press\ASP.NET 2.0 Step by Step\

A telepítő aspnet2sbs névvel virtuális könyvtárat hoz létre az alapértelmezett webhely alatt. Az aspnet2sbs virtuális könyvtár alatt több webes alkalmazást is létrehoz. A beállítások megtekintéséhez nyissuk meg az Internet Information Services konzolt.

A példaprogramok használata

A könyv fejezeteiben részletes utasításokat találunk az adott fejezet a példaprogramjainak használatáról. Amikor elérkezik a példaprogramok alkalmazásának ideje, a könyvben megtaláljuk a fájlok megnyitásához szükséges lépések listáját. Több fejezetben is a semmiből kezdünk teljesen új projektet. Így megvizsgálhatjuk a teljes fejlesztési folyamatot. Néhány példában visszaköszönnek majd a korábbi példák kódjai.

Az alábbiakban a példaprogramprojektek összefoglaló listáját találjuk.

Projekt	Leírás
1. fejezet	
HelloWorld.asp, Selectnoform.asp, Selectfeature.htm, Selectfeature2.htm, Selectfeature.asp	A nyers HTTP-kérések különböző példáit bemutató webes erőforrások.
WebRequestor	Egyszerű alkalmazás, amely nyers HTTP-kéréseket készít.
2. fejezet	
HelloWorld, HelloWorld2, HelloWorld3, HelloWorld4, HelloWorld5, partial1.cs, partial2.cs	Webes erőforrások, amelyek bemutatják az ASP.NET fordítási modelljét és a részleges osztályokat.
3. fejezet	
BunchOfControls.htm, BunchOfControls.asp, BunchOfControls.aspx	A vezérlőelem-címkék megjelenését bemutató webes erőforrások.
ControlORama	Visual Studio projekt, amely bemutatja a Visual Studio és a kiszolgálóoldali vezérlőelemeket.
4. fejezet	
ControlORama	Bemutatja a megjelenített kiszolgálóoldali vezérlőelemek létrehozását és használatát.
5. fejezet	
ControlORama	Bemutatja az összetett kiszolgálóoldali vezérlőelemek és a <i>User</i> vezérlőelemek létrehozását és használatát.

Projekt	Leírás
6. fejezet	
ControlPotpourri	Bemutatja a vezérlőelem-ellenőrzést, a <i>TreeView</i> és a <i>MultiView/View</i> vezérlőelemeket.
7. fejezet	
UseWebParts	Bemutatja a webkijelzők használatát a webes alkalmazásokban.
8. fejezet	
MasterPageSite	Főlapokat, témákat és felszíneket alkalmazó webes alkalmazás oldalainak segítségével bemutatja a megjelenés és az elrendezés kialakítását.
9. fejezet	
ConfigORama	Bemutatja az ASP.NET-en belüli konfigurációt. Bemutatja a Web.Config fájl kezelését, új konfigurációs elemek hozzáadását és a konfigurációs elemek visszakeresésének módszereit.
10. fejezet	
SecureSite	Bemutatja az űrlap alapú hitelesítés és a webhelyen belüli engedélyezéseket.
Login.aspx, OptionalLogin.aspx, Web.Config, Web.ConfigForceAuthentication, Web.ConfigForOptionalLogin	Az űrlap alapú hitelesítést a lehető legegyszerűbben bemutató webes erőforrások.
11. fejezet	
DataBindORama	Bemutatja a különböző vezérlőelemek, például a <i>GridView</i> vezérlőelem adatkötéseit, valamint az adathalmazok XML-ként vagy XML-sémaként való betöltését és mentését.
12. fejezet	
MaketPersonal	Bemutatja az ASP.NET 2.0 új személyre szabási szolgáltatásait.

Projekt	Leírás
13. fejezet	
SessionState	Bemutatja a munkamenet-állapot használatát webes alkalmazásban.
14. fejezet	
UseDataCaching	Bemutatja, hogy az adatok gyorsítótárazása hogyan javítja a rendszer teljesítményét.
15. fejezet	
OutputCaching	Bemutatja, hogy a kimenet gyorsítótárazása hogyan javítja a rendszer teljesítményét.
16. fejezet	
DebugORama	Bemutatja a webes alkalmazások hibakeresését és nyomkövetését.
17. fejezet	
UseApplication	Bemutatja a globális alkalmazásobjektumok és a HTTP-modulok az alkalmazás találkozási pontjaként való használatának módszereit. Bemutatja a globális hatókörű adatok tárolását és az alkalmazás eseményeinek kezelését. A projekt ekvivalens Visual Basic mintamodult is magában foglal.
18. fejezet	
CustomHandlers	Bemutatja az egyedi HTTP-kezelőket különálló szerelvényekben és ASHX-fájlokban is. A projekt C#- és Visual Basic kódot is tartalmaz.
19. fejezet	
WebServiceORama	Bemutat egy véletlenszerű idézeteket megjelenítő webszolgáltatást.
QuoteServiceVB	Bemutat egy véletlenszerű idézeteket megjelenítő Visual Basic webszolgáltatást.

Projekt	Leírás
20. fejezet	
DeployThis	Bemutatja, hogyan kell egy webhely telepítéséhez szükséges telepítőcsomagot előállítani.

Ezek a projektek a gyakorlati feladatok teljes megoldásaként állnak rendelkezésünkre (ha netán szükségünk lenne egy kis ihletre).

A példaprogramok eltávolítása

A következő lépések segítségével eltávolíthatja számítógépéről a példaprogramokat:

1. Nyissuk meg a Control Panel Add/Remove Programs menüpontját!
2. A Currently Installed Programs listából válasszuk a Microsoft ASP.NET 2.0 Step by Step pontot!
3. Kattintsunk a Remove gombra!
4. A példaprogramok eltávolításához kövessük a képernyőn megjelenő utasításokat!

Megjelenés előtti szoftver

A könyvet a Visual Studio 2005 CTP (**Common Technical Preview**, közösségi technikai előzetes) 2005. augusztusi verziójának segítségével lektoráltuk és teszteltük. Az augusztusi CTP volt az utolsó előzetes a Visual Studio 2005 végleges kiadásának megjelenése előtt. A könyv tartalma teljes mértékben kompatibilis a Visual Studio 2005 végleges kiadásával. A könyv tartalmának esetleges változásait és javításait a Microsoft összegyűjti, és Microsoft Knowledge Base cikkben közzéteszi. További információkat a Bevezető Terméktámogatás című részében találunk.

Hálózati melléklet

A hálózati melléklet lapja a könyvhöz kapcsolódó tartalmat, példaprogramokat és hivatkozásokat biztosít, többek között a Microsoft Press Technology Updates weblap hivatkozását is. (Ahogyan a könyvhöz kapcsolódó technológiák frissülnek, idővel a Microsoft Press Technology Updates weblap további információkkal bővül majd. A Visual Studio 2005 és a kapcsolódó technológiák újdonságairól olvashat, ha rendszeresen ellátogatunk a weblapra.) A könyv kísérő tartalom lapját a következő címen találjuk:

<http://www.microsoft.com/mspress/companion/0-7356-2201-9/>

Terméktámogatás

Mindent megtettünk a könyv és a CD-melléklet tartalmának pontosságáért. A könyv tartalmának változásai és javításai összegyűjtve egy Microsoft Knowledge Base cikkben jelennek meg. A könyv pillanatnyilag ismert hibaigazításainak listája a következő cikkben áll rendelkezésünkre:

<http://support.microsoft.com/KB/905042/>

A Microsoft Press könyveinek támogatása és a könyvek CD-mellékletei a következő webhelyen érhetők el:

<http://www.microsoft.com/learning/support/books/>

Kérdések és megjegyzések

Az Olvasó a könyvvel vagy a CD-melléklettel kapcsolatos megjegyzéseit, kérdéseit, ötleteit vagy a fenti webhely meglátogatása után is megválaszolatlan kérdéseit a Microsoft Press-hez e-mailben vagy postai úton juttathatja el.

Email: mspinput@microsoft.com

Postacím:

Microsoft Press

Attn: Step by Step Series Editor One Microsoft Way

Redmond, WA 98052-6399

A fenti címeken Microsoft-terméktámogatás nem áll rendelkezésre.