2007.

**Az IIS mint alkalmazáskiszolgáló**

**Turóczy Attila – Microsoft Student Consultant**

## Bevezetés

## A Microsoft által készített alkalmazás kiszolgáló az Internet Information Service. Ennek első verziója még 1996-ban jelent meg a WindowsNT 3.51 -höz az IIS 1.0. Kezdetben csak egy sima webkiszolgáló szerepét töltötte be. Majd az internetben rejlő üzleti lehetőségek miatt egyre nagyobb funkcionalitással látták el. Elterjedése a Windows 2000 Server megjelenése után kezdet rohamosan piacot nyerni. Majd a Microsoft asztali operációs rendszerébe is bekerült, így a Windows XP Professionalba már az IIS 5.1-el találkozhattunk. A Windows Server 2003-ba pedig az IIS 6.0-s verziója került bele. A Windows Vista -ba (Bussines-től) és a Windows Server 2008-ba pedig már egy teljesen újratervezett alkalmazás kiszolgálóval találkozhatunk, melynek moduláris felépítése az eddiginél sokkal nagyobb rugalmasságot, és skálázhatóságot, valamint biztonsági újításokat hordoz.

## Mi az alkalmazás kiszolgáló?

## Az alkalmazáskiszolgáló olyan alapvető technológia, amely kulcsinfrastruktúrát és szolgáltatásokat biztosít a rendszeren tárolt alkalmazások számára. Az alkalmazáskiszolgálók általában a következő szolgáltatásokat nyújtják:

* Erőforrás-készletezés (például adatbáziskapcsolat-készletezés és objektumkészletezés)
* Elosztott tranzakciók kezelése
* Aszinkron programkommunikáció, általában a message queuing szolgáltatás segítségével
* Az igény szerinti (Just-in-Time) objektumaktiválási modell
* Automatikus XLM alapú webszolgáltatási felületek az üzleti objektumok eléréséhez
* Feladatátvétel és az alkalmazások állapotfelmérésével kapcsolatos szolgáltatások
* Integrált biztonság

Ez mind-mind olyan általános tulajdonság, amellyel egy alkalmazás kiszolgálónak rendelkeznie kell, és ezeket a tulajdonságokat az IIS mind tudja. Ezenkívül a Windows Server 2003 operációs rendszerrel karöltve még az XML-webszolgáltatások, webalkalmazások és elosztott alkalmazások fejlesztését, bevezetését és futás közbeni felügyeletét segítő funkciókat is biztosít. Tehát az IIS egy széleskörű szolgáltatásokat nyújtó webkiszolgáló, amely infrastruktúrát biztosít a .NET és a már meglévő webalkalmazások és webszolgáltatások számára.

Dehogy mindezeket az állításokat be is bizonyítsuk, készítsünk egy elosztott webalkalmazást, melynek a webservicet és a web alkalmazását az IIS, fogja hostolni.

## Elosztott alkalmazás készítése, valamint publikálása IIS segítségével

Az egyszerűség kedvéért webalkalmazást készítünk, mely a kliens oldalon bekér két számot, majd ezt a két számot elküldi a WebService felé. Ezt a két számot a mi általunk készített webservice feldolgozza, kiértékeli és a két szám összegét visszaküldi a kliensnek ami megjeleníti az eredményt. A kliens ajax alapú lesz, hogy Post-Back ne történjen.

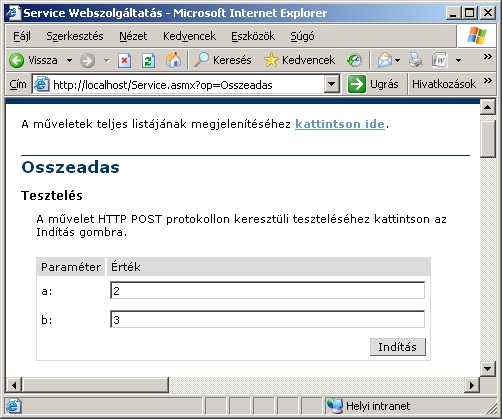
## WebService elkészítése

Indítsuk el a Visual Studio 2005-öt. Válasszuk ki a **File -> New ->Web Site** menüpontot. A megjelenő ablakon válasszuk ki az **ASP.NET WebService template** -t , majd adjunk neki egy nevet. Jelen esetben neveezzük el **OsszeadoService** –nek, majd kattintsunk az OK gombra. A Visual Studio legenerál számunkra egy alapértelmezett WebService templatet, majd ezt egészítsük ki a mi általunk elkészített összeadás webmetódussal.

|  |
| --- |
| [WebMethod]  public int Osszeadas(int a, int b)  {  return a + b;  } |

Most, hogy készek vagyunk a webszolgáltatásunkkal, leforditjuk (F6) és már publikálhatjuk is az IIS-re.  
  
*Megjegyzés: A Visual Studio 2005 rendelkezik egy sajátfejlesztő web és alkalmazás kiszolgálóval direkt fejlesztési célra. Ezt a külvilág nem érheti el. Tehát akár a Visual Studio 2005 segítségével is kipróbálhattuk volna az alkalmazásunkat .*

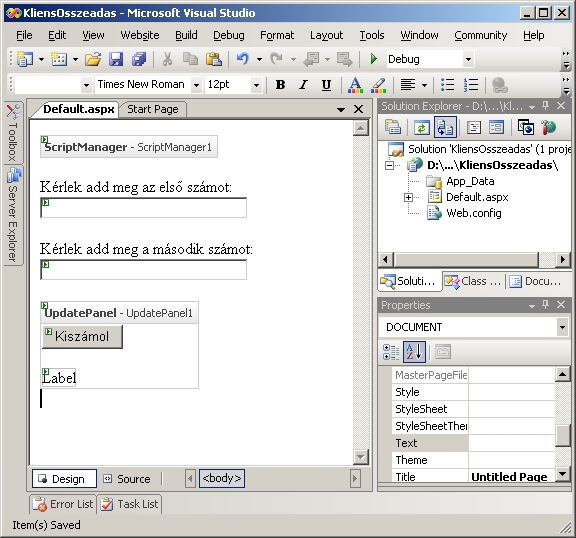
## WebService publikálása IIS kiszolgálóra

 A publikálás ezen egyszerű esetben történhet egy egyszerű fájl másolással is. De a Visual Studio 2005 rendelkezik egy külön erre a célra fejlesztett felülettel is, valamint akár telepítő készletett is készíthetünk web alkalmazásunkhoz. Most válaszuk a Visual Studio nyújtotta publikációs felületet. Kattintsunk a **WebSite** menü **Copy Web Site** menüpontjára . A megjelenő ablakban kattintsunk a Connect gombra, majd a megjelenő ablakba válasszuk ki, hogy helyi IIS kiszolgálóra szeretnénk publikálni, ezt követően válasszuk ki, hogy melyik Virtual Directoryba kívánjuk publikálni , majd kattintsunk az OK gombra. A Visual Studio publikáló eszköze most kapcsolódott a helyi IIS kiszolgálóra, ezt követően elég átmásolnunk a Service.asmx fájlt és az App\_Code mappát. Ha ezzel készen vagyunk, nyissuk meg kedvenc böngészőnket és a címsorba írjuk a service –ünk elérési útvonalát. Jelen esetben nekünk elég a következő útvonal is <http://localhost/Service.asmx> . Most egy automatikus XLM alapú web szolgáltatási felületet látunk magunk előtt melynek segítségével tesztelhetjük, és megtekinthetjük a web szolgáltatásunkat. Próbáljuk is ki. Adjunk meg két számot, legyen most a 2 és a 3. Ezt követően kattintsunk az Intiást gombra. És a következő eredményt kapjuk:

|  |
| --- |
| <<?xml version="1.0" encoding="utf-8" ?>  <i<int xmlns="**http://tempuri.org/**">**5**</int> |

Mint látjuk az eredményt egy XML dokumentumban kapjuk meg, és ezt küldi majd az alkalmazásunk felé is.

## Webkliens alkalmazás elkészítése

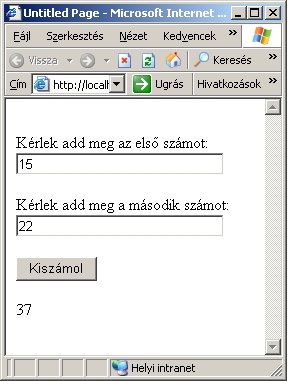
 Most, hogy elkészítettük a webszolgáltatásunkat írjunk egy kliens alkalmazást. A Web kliens alkalmazás elkészítéséhez indítsuk el a Visual Studio 2005-öt. Válasszuk ki a **File -> New ->Web Site** menüpontot. A megjelenő ablakon válasszuk ki az **ASP.NET AJAX-Enabled WebSite** templatet , majd adjunk neki egy nevet. Jelen esetben nevezzük el KliensOsszeadas –nak majd kattintsunk az OK gombra. Miután a Visual Studio elkészítette a templatet, kattintsunk Design nézetre, majd húzzunk a WebFormra két TextBox-ot egy Buttont és egy Label controlt. Majd húzzunk fel egy Ajax-os UpdatePanelt is a Formra. Ebbe az UpdatePanelbe helyezzük bele a Button és a Label controlt. Ugyanis ez a két control fog aszinkron módon működni. Ha ezzel meg vagyunk a Formunk nagyábból a következő alakot fogja mutatni. Ugyan kész a formunk kinézete de még tartalom, nem áll mögötte. De még mielőtt belekezdenénk kódunk megírásába, a webszolgálatatásunkat megkell hívnunk a Solutionünkből. Kattintsunk a Solution Explorerre (jobb oldalt). Majd Jobb egérgomb és kattintsunk **Add Web Reference** menüpontra. A Megjelenő ablak címsorábaírjuk be a webszolgáltatásunk URL-jét, ami jelen esetben a <http://localhost/Service.asmx> , ha sikerrel jártunk akkor a tesztelésnél megszokott kép fogad minket és ugyanúgy kipróbálhatjuk a webszolgáltatásunkat. Mielőtt bezárnánk az ablakot a Web Reference name-hez irjuk be azt a nevet amivel a Web szolgáltatásunkra hivatkozni fogunk, ez a név most legyen a WebSzolg, majd kattintsunk az **Add Reference** gombra. Most már sikeresen hozzáadtuk az alkalmazásunkhoz a webszolgáltatásunk referenciáját.

Most már nincs más hátra, csak a kód megírása. Kattintsunk kétszer a Buttonra, ekkor a Visual Studio beköti az alapértelmezett esemény vezérlőt, és megírhatjuk a kódunkat. Ami a következőképpen fog kinézni:

|  |
| --- |
| protected void Button1\_Click(object sender, EventArgs e)  {  WebSzolg.Service szolg = new WebSzolg.Service();  int a = Convert.ToInt32(TextBox1.Text.ToString());  int b = Convert.ToInt32(TextBox2.Text.ToString());  int eredmeny = szolg.Osszeadas(a, b);  Label1.Text = eredmeny.ToString();  } |

A kód tulajdonképpen bekér két számot, és a két számot elküldi a webszolgáltatásnak, ami összeadja és visszaküldi az alkalmazásnak. Anélkül, hogy Post-Back történne a webformon.

## Tesztelés

Most, hogy kész a kliens alkalmazásunk, próbáljuk ki. Kattintsunk a Debug menü Start Debugging menüpontjára (F5). Ekkora a Visual Studio belső webszerverén fog futni a kliens. Természetesen akár publikálhatnánk is egy IIS kiszolgálóra az eredmény ugyan az.

A két szövegmezőben adjunk meg két számot. Majd kattintsunk a Kiszámol Buttonra, és láthatjuk, hogy megjelenik az eredmény, anélkül, hogy a formon post-back történt volna. É s maga az összead függvény pedig az IIS –en hostolt webszolgáltatás számolta ki. Ezen rövid és egyszerű példa is ékes bizonyítéka annak, hogy az IIS és az ASP.NET egy alkalmazás kiszolgáló. Természetesen az IIS akár a PHP-val is együtt tud működni.

## Összegzés

Tehát az IIS egy széleskörű szolgáltatásokat nyújtó webkiszolgáló, amely infrastruktúrát biztosít a .NET és a már meglévő webalkalmazások és webszolgáltatások számára.