

برای استفاده از سیستم مدیریت کاربران و نقش‌های آنها به یک پیاده سازی از کلاس انتزاعی MembershipProvider نیاز داریم. SQL Membership Provider تو کار دات نت، انتخاب پیش فرض ماست ولی به دلیل طراحی در دات نت 2 و نیاز سنجی قدیمی اون و همچنین گره زدن برنامه با sql server (استفاده از stored procedure و ...) انتخاب مناسبی نیست. پیشنهاد خود مایکروسافت استفاده از [SimpleMembership](#) است که این پیاده سازی قابلیت‌های بیشتری از MembershipProvider پایه رو دارد. این قابلیت‌های بیشتر با استفاده از کلاس انتزاعی ExtendedMembershipProvider که خود از MembershipProvider مشتق شده است میسر شده است. برای این آموزش ما از SimpleMembership استفاده می‌کنیم اگر شما دوست ندارید از SimpleMembership استفاده کنید می‌تونید از Provider های دیگه ای استفاده کنید و حتی می‌تونید یک پروایدر سفارشی برای خودتون بنویسید.

[Custom Membership Providers](#)

[Codefirstmembership](#)

[EFmembership](#)

...

برای شروع یک پروژه ConsoleApplication تعریف کنید و رفرنس‌های زیر رو اضافه کنید.

```
System.Web.dll
System.Web.ApplicationServices.dll
```

خاصیت Copy Local دو کتابخانه زیر رو true ست کنید.

```
C:\Program Files (x86)\Microsoft ASP.NET\ASP.NET Web Pages\v2.0\Assemblies\WebMatrix.Data.dll
C:\Program Files (x86)\Microsoft ASP.NET\ASP.NET Web Pages\v2.0\Assemblies\WebMatrix.WebData.dll
```

- در صورتیکه یک پروژه Asp.Net MVC 4 به همراه تمپلت Internet Application بسازید بصورت خودکار SimpleMembership و رفرنس‌های آن به پروژه اضافه می‌شود.

یک فایل App.config با محتویات زیر به پروژه اضافه کنید و تنظیمات ConnectionString را مطابق با دیتابیس موجود در کامپیوتر خود تنظیم کنید:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>

  <connectionStrings>
    <add name="DefaultConnection"
      connectionString="Data Source=.;Initial Catalog=SimpleMembershipProviderDB;Integrated
Security=True;"
      providerName="System.Data.SqlClient"/>
  </connectionStrings>

  <system.web>
    <roleManager enabled="true" defaultProvider="SimpleRoleProvider">
      <providers>
        <clear/>
        <add name="SimpleRoleProvider" type="WebMatrix.WebData.SimpleRoleProvider, WebMatrix.WebData"/>
      </providers>
    </roleManager>
    <membership defaultProvider="SimpleMembershipProvider">
      <providers>
        <clear/>
        <add name="SimpleMembershipProvider" type="WebMatrix.WebData.SimpleMembershipProvider,
WebMatrix.WebData"/>
      </providers>
    </membership>
  </system.web>
</configuration>
```

```
</membership>
</system.web>
</configuration>
```

محتویات فایل Program.cs :

```
using System;
using System.Security.Principal;
using System.Web.Security;
using WebMatrix.WebData;

namespace MemberShipConsoleApplication
{
    class Program
    {
        static void Main(string[] args)
        {
            WebSecurity.InitializeDatabaseConnection("DefaultConnection",
                "UserProfile", "UserId", "UserName",
                autoCreateTables: true);

            AddUserAndRolSample();
            Login();
            if (System.Threading.Thread.CurrentPrincipal.Identity.IsAuthenticated)
                RunApp();
        }

        static void AddUserAndRolSample()
        {
            if (WebSecurity.UserExists("iman"))
                return;

            // No implements in SimpleMembershipProvider :
            // Membership.CreateUser("iman", "123");
            WebSecurity.CreateUserAndAccount("iman", "123");
            Roles.CreateRole("admin");
            Roles.CreateRole("User");
            Roles.AddUserToRole("iman", "admin");
        }

        static void Login()
        {
            for (int i = 0; i < 3; i++)
            {
                Console.Write("UserName: ");
                var userName = Console.ReadLine();
                Console.Write("Password: ");
                var password = Console.ReadLine();
                if (Membership.ValidateUser(userName, password))
                {
                    var user = Membership.GetUser(userName);
                    var identity = new GenericIdentity(user.UserName);
                    var principal = new RolePrincipal(identity);
                    System.Threading.Thread.CurrentPrincipal = principal;

                    Console.Clear();
                    return;
                }

                Console.WriteLine("User Name Or Password Not Valid.");
            }
        }

        static void RunApp()
        {
            Console.WriteLine("Welcome To MemberShip. User Name: {0}",
                System.Threading.Thread.CurrentPrincipal.Identity.Name);

            if (System.Threading.Thread.CurrentPrincipal.IsInRole("admin"))
                Console.WriteLine("Hello Admin User!");

            Console.Read();
        }
    }
}
```

```
}
```

در ابتدا با فراخوانی متد InitializeDatabaseConnection تنظیمات اولیه simpleMembership را مقدار دهی می‌کنیم. این متد حتما باید یکبار اجرا شود.

```
InitializeDatabaseConnection(string connectionStringName,
                             string userTableName,
                             string userIdColumn,
                             string userNameColumn,
                             bool autoCreateTables)
```

ملاحظه می‌کنید پارامتر آخر متد مربوط به ساخت جداول مورد نیاز این پروایدر است. در صورتی که بخواهیم در پروژه از EntityFramework استفاده کنیم می‌تونیم موجودیت‌های معادل جدول‌های مورد نیاز SimpleMembership رو در EF بسازیم و در این متد AutoCreateTables رو False مقدار دهی کنیم. برای بدست آوردن موجودیت‌های معادل جدول‌های این پروایدر با ابزار [Entity Framework Power Tools](#) و روش مهندسی معکوس ، تمام موجودیت‌های یک دیتابیس ساخته شده رو استخراج می‌کنیم. SimpleMembership از تمام خانواده microsoft sql پشتیبانی می‌کند (SQL Server, SQL Azure, SQL Server CE, SQL Server) - Express, LocalD برای سایر دیتابیس‌ها به علت تفاوت در دستورات ساخت تیل‌ها و ... میکروسافت تضمینی نداده ولی اگر خودمون جدول‌های مورد نیاز SimpleMembership رو ساخته باشیم احتمالا در سایر دیتابیس‌ها هم قابل استفاده است. در ادامه برنامه بالا یک کاربر و دو نقش تعریف کردیم و نقش admin رو به کاربر نسبت دادیم. در متد login در صورت معتبر بودن کاربر ، اون رو به ترد اصلی برنامه معرفی می‌کنیم. هر جا خواستیم می‌تونیم نقش‌های کاربر رو چک کنیم و نکته آخر با اولین چک کردن نقش یک کاربر تمام نقش‌های اون در حافظه سیستم کش می‌شود و تنها مرتبه اول با دیتابیس ارتباط برقرار می‌کند.