**ASP.NET IDENTITY教程笔记**

课程链接: <https://app.pluralsight.com/library/courses/asp-dot-net-identity-fundamentals/table-of-contents>

主要特性：

双因子认证

外部认证

ASP.NET MVC结合使用

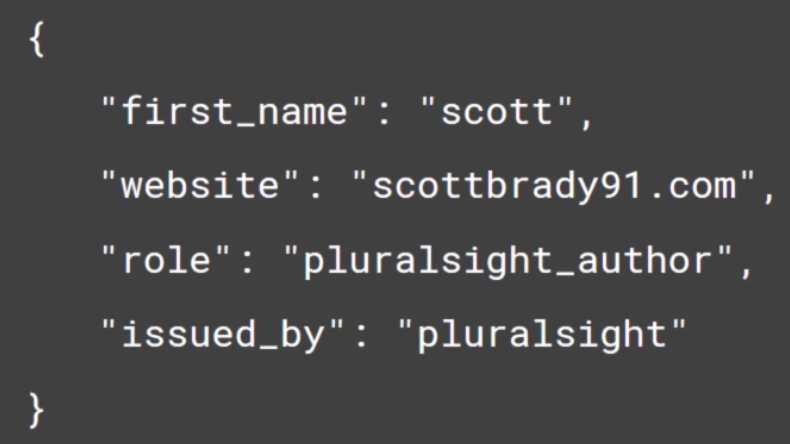
claims Based Identity

**Identity**: Specifuc to the User; Idenity不一定需要由自己的应用程序生成，比如：用户名、密码和Permission；外部比如：使用google、facebook或者微信登陆生成，形式：OpenID；

Identity的形式、内容：

也有可能是个人信息，比如: TELEPHONE NUMBER, EMAIL等; 或者组织内的Role

**ClaimBased Identity**



不同步骤：

1 Identity Management Layer, identity data 存储和进入的地方; 可以包括 注册方法、密码验证方法等;

2 Authentication; 解决“验证一个用户是谁”的问题;

3 Authorization, 解决“用户能做什么“的问题;

**ASP.NET Identity Components?**

**password hashing;user lockout; two factor authentication; external logins; tokenization; High Extensible**

**ASP.NET Identity 不是 Identity2; 不是Identity Framework；也不是IdentityServer；也不是General Purpose Database; 不是 Securitity Framework**

**History**

**2005 ASP.NET MEMBERSHIP, 不足：难以扩展，使用SHA1，已经不安全**

**2011 Universal Membership**

**2012 simple Membership**

**2013 ASP.NET Identity V1**

**2014 ASP.NET Identity V2  支持reset functionality, account lockout, and 2 factor authentication**

**2016 ASP.NET CORE Identity**

**Demo1--创建用户**

**1 新建 Console项目， 选择.NET 4.6**

**2 安装 Microsoft.AspNet.Identity.Core**

**3 安装 Microsoft.AspNet.Identity.EntityFramework**

**4 web.config, configuration节点下，添加新节点**

<connectionStrings>

    <add name="DefaultConnection" connectionString="Data Source=(LocalDb)\MSSQLLocalDB;Database=Pluralsight.AspNetIdentityDemo.Module1;trusted\_connection=yes;" providerName="System.Data.SqlClient"/>

  </connectionStrings>

**5 main函数代码如下**

 static void Main(string[] args)

        {

            var userStore = new UserStore<IdentityUser>();

            var userManager = new UserManager<IdentityUser>(userStore);

            var creationResult = userManager.Create(new IdentityUser("scott@scottbrad91.com"),"Password123!");

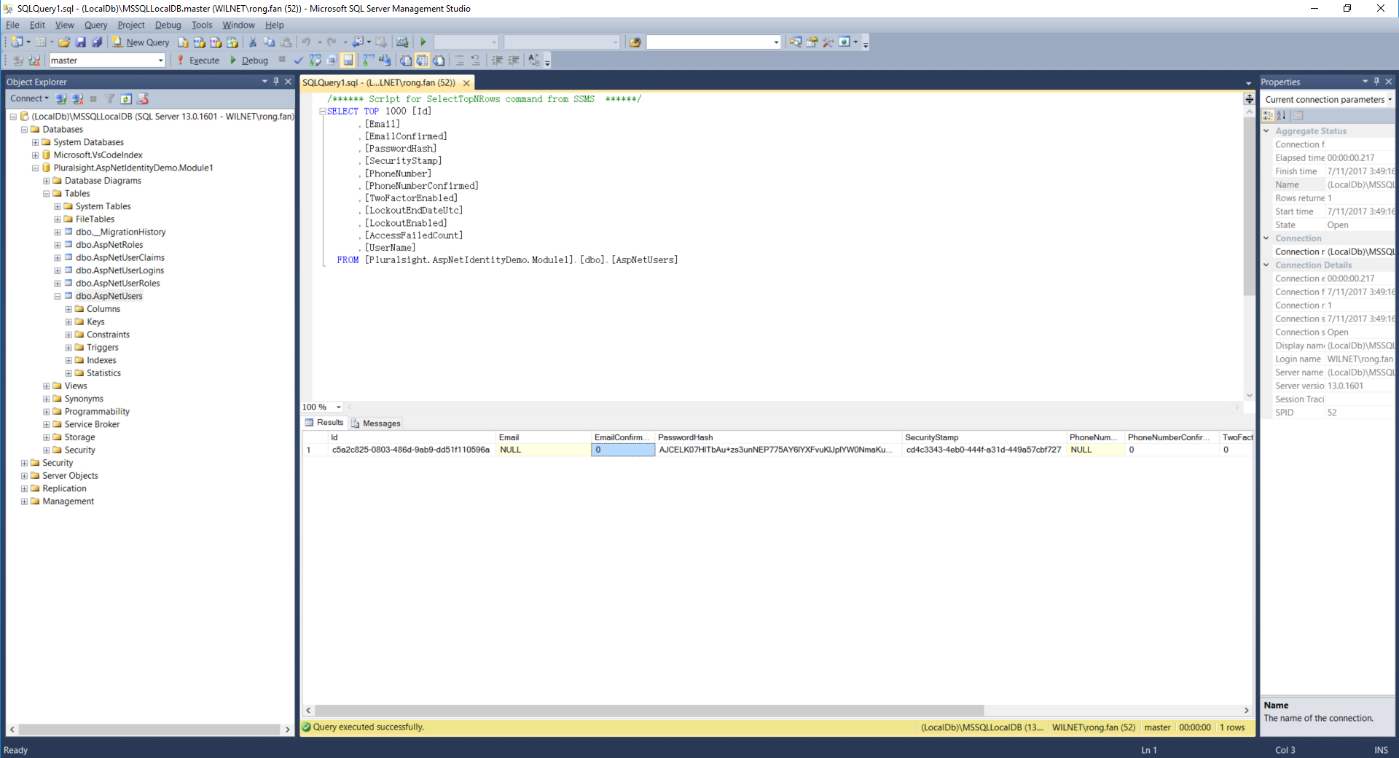
            Console.WriteLine("Created:{0}",creationResult.Succeeded);

            Console.ReadKey();

        }

**6 数据库查看结果**

用户创建成功，而且密码已经加密



**Demo2--给User添加Claim**

 static void Main(string[] args)

        {

            var userStore = new UserStore<IdentityUser>();

            var userManager = new UserManager<IdentityUser>(userStore);

            string userName = "scott@scottbrad91.com";

            string password = "Password123";

            //为了userID, 需要找到User

            IdentityUser user = userManager.FindByName(userName);

            //给用户添加Claims，后台自动使用EF存储到数据库

            IdentityResult claimResult = userManager.AddClaim(user.Id, new System.Security.Claims.Claim("given\_name","Scott"));

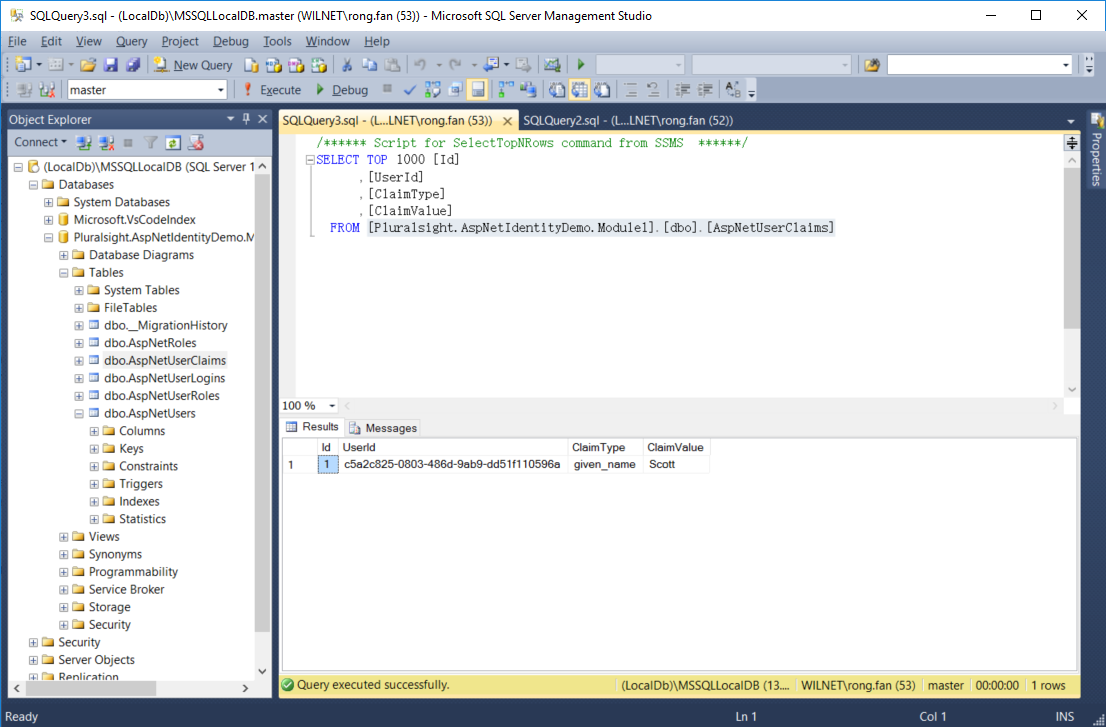
            Console.WriteLine("Created:{0}", claimResult.Succeeded);

            Console.ReadKey();

        }

**效果**

查看表[Pluralsight.AspNetIdentityDemo.Module1].[dbo].[AspNetUserClaims]，发现记录已经存在



**-------------------------------------------------**

**Demo3--验证密码**

static void Main(string[] args)

        {

            var userStore = new UserStore<IdentityUser>();

            var userManager = new UserManager<IdentityUser>(userStore);

            string userName = "scott@scottbrad91.com";

            string password = "Password123!";

            //为了userID, 需要找到User

            IdentityUser user = userManager.FindByName(userName);

            //verify user password

            bool isMatch = userManager.CheckPassword(user,password);

            Console.WriteLine("Created:{0}", isMatch);

            Console.ReadKey();

        }

**效果：成功**