

آموزش احراز هویت user identity بخش اول

خانه (/) / مقالات (Contents/Category/۱۰۴/) / آموزش احراز هویت user identity بخش اول



آموزش احراز هویت user identity بخش اول

در این مقاله قصد داریم گام به گام مراحل ایجاد و کار با روش احراز هویت user identity را بررسی کنیم. با ما همراه باشید تا به سادگی یک برنامه بر این پایه ایجاد نمایید.

PM > Install-Package Microsoft.Owin.Host.SystemWeb

با سلام احراز هویت user identity به عقیده اکثر برنامه نویسان بسیار گیج کننده و پیچیده است اما نظر من متفاوت است و به نظر من این ساختار بسیار ساده جذاب و روان است و قطعاً می تواند کمک های بسیاری به شما بکند و ویژگی های بسیاری به برنامه شما بیفزاید.

با ما همراه باشید تا بعد از گذراندن ۳۳ گام ساده و واضح یک برنامه به پایه user identity داشته باشید.

در این مقاله قصد داریم به صورت ساده تنها به پیاده سازی ساختار بدون استفاده از دیتابیس پردازیم تا سادگی و روان بودن آموزش ملموس تر شود. در مقاله بعدی نیز به طور کامل پایگاه داده را وارد برنامه میکنیم تا آموزش کامل شود.

قبل از شروع باید بگم کسی که شروع به کار با asp.net user identity میکند حتماً به آشنایی با برنامه نویسی Asp.net mvc داره و پیش فرض ما در آموزش این هست و از توضیحات اضافه پرهیز میکنیم تا حجم مقاله خیلی زیاد نشود.

خب بیاید شروع کنیم:

(۱) ابتدا یک پروژه خالی mvc ایجاد کنید. یک پروژه empty با ساختار mvc

(۲) از طریق نوگت پکیج زیر را نصب:

(۳) همچنین چون میخواهیم بر پایه کوکی احراز هویت کنیم پکیج زیر را نیز نصب نمایید:

```
PM > Install-Package Microsoft.Owin.Security.Cookies
```

(۴) بر روی پروژه راست کلیک و گزینه Owin Startup Class add-> را میزنیم.

(۵) کلاس را با نام Startup ایجاد کنید.

(۶) کد زیر را داخل متد configuration کلاس مرحله ۵ قرار میدهیم:

```
app.UseCookieAuthentication(new CookieAuthenticationOptions
{
    AuthenticationType = "ApplicationCookie",
    LoginPath = new PathString("/Account/Login")
});
```

۷) UseCookieAuthentication به برنامه میگوید میخواهیم بر پایه cookie احراز هویت را انجام دهیم.

۸) LoginPath مسیری که اگر کاربر غیر مجاز بود به آنجا Redirect گردد. (میتوانید برای خود شخصی سازی نمایید).

۹) یک پوشه به نام Filter ایجاد کنید و یک کلاس به نام FilterConfig به آن اضافه نمایید (فیلترها برای انجام اعمالی قبل از اجرای اکشن ها استفاده میشوند و ما میخواهیم از فیلتر برای احراز هویت بهره ببریم). سپس متد زیر را درون کلاس قرار دهید:

```
public static void RegisterGlobalFilters(GlobalFilterCollection filters)
{
    filters.Add(new HandleErrorAttribute());
    filters.Add(new AuthorizeAttribute());
}
```

۱۰) در global.asax.cs تکه کد زیر را در انتهای تابع Application_Start () قرار میدهیم تا فیلتر اعمال گردد:

```
FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
```

۱۱) کنترلر Home را ایجاد کنید.

۱۲) View متد index را اضافه میکنیم.

۱۳) کنترلر Account را برای احراز هویت کاربران ایجاد کنید و در بالای کنترلر از AllowAnonymous ، attribute استفاده کنید تا این بخش برای احراز هویت همه قابل دسترس باشد. پس کنترلر ما به صورت زیر خواهد شد:

```
[AllowAnonymous]
public class AccountController : Controller
{
    public ActionResult Login()
    {
        return View();
    }
}
```

۱۴) View مربوط به متد مربوط به متد Login را ایجاد میکنیم.

۱۵) حال پروژه را اجرا کنید میبینید که به جاي Home/Index يا / به Account/Login ارسال میشود.

۱۶) كلا هر كنترلري كه بسازيد بر پايه احراز هویت خواهد بود. هنگامیکه بخواهید بعضی از كنترلرهای یا اکشن های شما نیاز به احراز هویت نداشته باشند کافی است [AllowAnonymous] را به بالاي كنترلر یا اکشن مورد نظر اضافه کنید تا اجازه دسترسی داده شود.

۱۷) در پوشه Models یک پوشه به نام ViewModels ایجاد و سپس یک کلاس به نام Login ایجاد میکنیم و آن را به صورت زیر در می آوریم:

```
public class Login
{
    [Required]
    [DataType(DataType.EmailAddress)]
    public string Email { get; set; }

    [Required]
    [DataType(DataType.Password)]
    public string Password { get; set; }

    [HiddenInput]
    public string returnUrl { get; set; }
}
```

۱۸) حال اکشن Login را به صورت زیر تغییر دهید:

```
[HttpGet]
public ActionResult LogIn(string returnUrl)
{
    var model = new Models.ViewModels.Login
    {
        ReturnUrl = returnUrl
    };

    return View(model);
}
```

۱۹) حال دو تابع زیر را به کنترلر Account اضافه میکنیم که اولی برای post لاگین است و دومی برای Redirect کردن به صفحه اولیه.

```
[HttpPost]
public ActionResult Login(models.ViewModels.Login model)
{
    if (!ModelState.IsValid)
    {
        return View();
    }

    // Don't do this in production!
    if (model.Email == "admin@admin.com" && model.Password == "password")
    {
        var identity = new ClaimsIdentity(new[] {
            new Claim(ClaimTypes.Name, "Ben"),
            new Claim(ClaimTypes.Email, "a@b.com"),
            new Claim(ClaimTypes.Country, "England")
        },
            "ApplicationCookie");

        var ctx = Request.GetOwinContext();
        var authManager = ctx.Authentication;

        authManager.SignIn(identity);

        return Redirect(GetRedirectUrl(model.ReturnUrl));
    }

    // user authN failed
    ModelState.AddModelError("", "Invalid email or password");
    return View();
}

private string GetRedirectUrl(string returnUrl)
{
    if (string.IsNullOrEmpty(returnUrl) || !Url.IsLocalUrl(returnUrl))
    {
        return Url.Action("index", "home");
    }

    return returnUrl;
}
```

۲۰) به Login.cshtml بروید و کد آن را به صورت زیر تغییر دهید:

```
@model user_identity.Models.ViewModels.Login
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
@section header{
    <title>ورود</title>
}

<h2>ورود</h2>
<h2>Log In</h2>

@Html.ValidationSummary(true)

@using (Html.BeginForm())
{
    @Html.EditorForModel()
    <p>
        <button type="submit">ورود</button>
    </p>
}
```

۲۱) حال اگر با (admin@admin.com/password) ورود کنید به صفحه خانه بازگشت داده می‌شوید برای اینکه از ورود اطمینان حاصل کنید تکه کد زیر را به view اصلی home اضافه کنید تا نام کاربر را نمایش دهد.

```
<p>
    Hello @User.Identity.Name
</p>
```

۲۲) حال برای خروج کاربر تابع زیر را به کنترلر Account اضافه کنید.

```
public ActionResult Logout()
{
    var ctx = Request.GetOwinContext();
    var authManager = ctx.Authentication;

    authManager.SignOut("ApplicationCookie");
    return RedirectToAction("Index", "Home");
}
```

۲۳) سپس تکه کد زیر را به View اصلی home اضافه کنید تا دکمه خروج ایجاد شود:

```
<p>  
    <a href="@Url.Action("logout", "Account")">خروج</a>  
</p>
```

۲۴) در حال حاضر ما تنها امکان دسترسی به نام کاربر جاری را داریم که نمایش دادیم اما برای دسترسی به بقیه موارد اطلاعات کاربر نیاز است تا اکشن index در Home را به صورت زیر بازنویسی کنیم:

```
var claimsIdentity = User.Identity as ClaimsIdentity;  
ViewBag.Country = claimsIdentity.FindFirst(ClaimTypes.Country).Value;
```

که در آن user.Identity را به ClaimsIdentity تبدیل نموده ایم تا با استفاده از خواص آن به مقادیری که قبلاً مقدار داده ایم دسترسی داشته باشیم که در اینجا نما کشور است.

۲۵) حال برای نمایش نام کشور کاربر جاری تکه کد زیر را به index ، view کنترلر Home اضافه میکنیم :

```
<p>  
    چطور ه @ViewBag.Country هوای @User.Identity.Name. سلام  
</p>
```

۲۶) راه بهتر برای کار بالا به این صورت است: ابتدا یک پوشه به نام Identity ایجاد کنید و درون آن یک کلاس با نام AppUser ایجاد کنید و محتوای آن را به صورت زیر قرار دهید:


```

public class AppUser : ClaimsPrincipal
{
    public AppUser(ClaimsPrincipal principal)
    : base(principal)
    {
    }

    public string Name
    {
        get
        {
            return this.FindFirst(ClaimTypes.Name).Value;
        }
    }

    public string Country
    {
        get
        {
            return this.FindFirst(ClaimTypes.Country).Value;
        }
    }
}

```

این کلاس از ClaimsPrincipal ارث میبرد و با این کلاس دو خاصیت نام و کشور را ایجاد که بعداً برای ما قابل دسترسی باشد.

۲۷) سپس یک کنترلر جدید با نام BaseController ایجاد میکنیم و درون آن یک ویژگی CurrentUser که از نوع کلاس AppUser مرحله قبل است ایجاد میکنیم.

```

public abstract class BaseController : Controller
{
    public AppUser CurrentUser
    {
        get
        {
            return new AppUser(this.User as ClaimsPrincipal);
        }
    }
}

```

سپس در کنترلرهای HomeController و AllowAnonymousController از این کنترلر جدید ارث میبریم در واقع همه کنترلرهای باید همیشه از این کنترلر ارث ببرند تا اطلاعات کاربر درون آن باشد.

۲۸) حال باید متد index درون کنترلر home را به صورت زیر در آوریم

```
public ActionResult Index()
{
    ViewBag.Country = CurrentUser.Country;

    return View();
}
```

که در آن Property ، CurrentUser همان کنترلر base است که در همه جا وجود دارد و قبلا آن را با کلاس AppUser پر نمودیم

۲۹) می‌خواهیم کار را بهتر کنیم و دیگر از viewbag هم استفاده نکنیم. خوب ابتدا در پوشه identity یک کلاس به نام AppViewPage بسازید و سپس آن را به صورت زیر بازنویسی کنید:

```
public abstract class AppViewPage : WebViewPage
{
    protected AppUser CurrentUser
    {
        get
        {
            return new AppUser(this.User as ClaimsPrincipal);
        }
    }
}

public abstract class AppViewPage : AppViewPage
{
}
```

۳۰) حالا به web.config داخل پوشه view بروید و تگ page در درون تگ را به شکل زیر بازنویسی مجدد نمایید:

```
<pages pageBaseType="user_identity.Identity.AppViewPage">
```

۳۱) حال به view اصلی home بروید محتوای زیر را جایگزین محتوایی کنید که در مرحله ۲۵ اضافه کرده بودیم:

```
<p>
    چگونه @CurrentUser.Country هوای @CurrentUser.Name سلام
</p>
```

۳۲) حال کدهای اضافه را از اکشن index در home پاک کرده به شکل زیر در آورید:

```
public ActionResult Index()
{
    return View();
}
```

۳۳) کار تمام است. برنامه را اجرا کنید و از احراز هویت بر پایه user identity لذت ببرید.

به نظرم حالا شما هم، هم عقیده من شده اید که این ساختار نه تنها پیچیده نیست اتفاقاً خیلی ساده و جذاب میباشد.

حالا بهتره بعد از یه استراحت کوچیک بریم به سراغ مقاله بعدی ما که پیاده سازی user identity به همراه دیتا بیس خواهد بود. مطمئن باشید اون هم مثل این بخش خیلی ساده هست و فقط نیاز به حضور ذهنی شما دارد.

کلیدواژه: آموزش برنامه نویسی (/Tag/آموزش-برنامه-نویسی) , آموزش (/Tag/آموزش) , user identity (/Tag/user-identity) , برنامه نویسی (/Tag/برنامه-نویسی)
👁 ۱۵۰۷۰ بازدید

نظرات

نام

نام

ایمیل

ایمیل

شماره تماس

موبایل

نظر

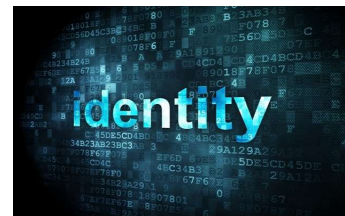
نظر

ارسال

مطالب محبوب

آموزش احراز هویت user identity بخش اول

Contents/۱۱۴۱/%D۸%A۲%D۹%۸۵%D۹%۸۸%D۸%B۲%D۸%B۴-/
 %D۸%A۷%D۸%AD%D۸%B۱%D۸%A۷%D۸%B۲-%D۹%۸۷%D۹%۸۸%DB%۸C%D۸%AA-user-
 (identity-%D۸%A۸%D۸%AE%D۸%B۴-%D۸%A۷%D۹%۸۸%D۹%۸۴

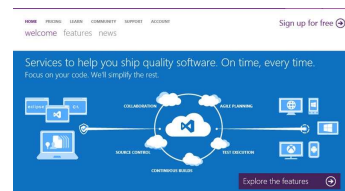


Contents/۱۱۶۵/%D۹%۸۷%D۸%A۷%D۸%B۳%D۸%AA-/
 %D۸%B۱%D۸%A۷%DB%۸C%DA%AF%D۸%A۷%D۹%۸۶-
 (%D۹%۸۸%DB%۸C%D۹%۸۶%D۸%AF%D۹%۸۸%D۸%B۲



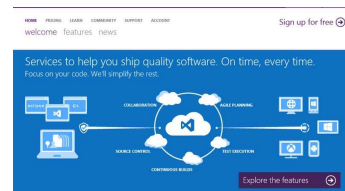
آموزش کار با team foundation server به زبان ساده برای تازه کارها (۱)

Contents/۱۱۲۹/%D۸%A۲%D۹%۸۵%D۹%۸۸%D۸%B۲%D۸%B۴-%DA%A۹%D۸%A۷%D۸%B۱-/
 %D۸%A۸%D۸%A۷-team-foundation-server-%D۸%A۸%D۹%۸۷-
 %D۸%B۲%D۸%A۸%D۸%A۷%D۹%۸۶-%D۸%B۳%D۸%A۷%D۸%AF%D۹%۸۷-
 %D۸%A۸%D۸%B۱%D۸%A۷%DB%۸C-%D۸%AA%D۸%A۷%D۸%B۲%D۹%۸۷-
 (%DA%A۹%D۸%A۷%D۸%B۱%D۹%۸۷%D۸%A۷-۱



آموزش کار با team foundation server به زبان ساده برای تازه کارها (۲)

Contents/۱۱۳۰/%D۸%A۲%D۹%۸۵%D۹%۸۸%D۸%B۲%D۸%B۴-%DA%A۹%D۸%A۷%D۸%B۱-/
 %D۸%A۸%D۸%A۷-team-foundation-server-%D۸%A۸%D۹%۸۷-
 %D۸%B۲%D۸%A۸%D۸%A۷%D۹%۸۶-%D۸%B۳%D۸%A۷%D۸%AF%D۹%۸۷-
 %D۸%A۸%D۸%B۱%D۸%A۷%DB%۸C-%D۸%AA%D۸%A۷%D۸%B۲%D۹%۸۷-
 (%DA%A۹%D۸%A۷%D۸%B۱%D۹%۸۷%D۸%A۷-۲

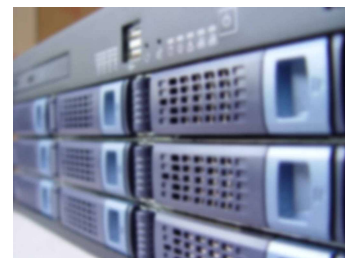


آموزش برنامه نویسی (Contents/۱۱۲۴/%D۸%A۲%D۹%۸۵%D۹%۸۸%D۸%B۲%D۸%B۴-/)

%D۸%A۸%D۸%B۱%D۹%۸۶%D۸%A۷%D۹%۸۵%D۹%۸۷-
 (%D۹%۸۶%D۹%۸۸%DB%۸C%D۸%B۳%DB%۸C



هاست (Contents/۱۱۱۳/%D۹%۸۷%D۸%A۷%D۸%B۳%D۸%AA-/)



ir (/Contents/۱۲۰۷/%D۸%AF%D۸%A۷%D۹%۸۵%D۹%۸۶%D۹%۸۷-
 (%D۸%B۱%D۸%A۷%DB%۸C%DA%AF%D۸%A۷%D۹%۸۶-ir



آخرین اخبار

Contents/1۲۷۷/%D۸%AA%D۸%AE%D۹%۸1%DB%۸C%D۹%۸1-%D۸%AA%D۹%۸۷-/
%D۹%۸۵%D۹%۸۶%D۸%۸۷%D۸%B۳%D۸%AA%D۸%AA-
(%D۹%۸۷%D۹%۸1%D۸%AA%D۹%۸۷-%D۹%۸۸%D۸%AD%D۸%AF%D۸%AA



تخفيف به مناسبت دهه کرامت (Contents/۱۲۷۱/%D۸%AA%D۸%AE%D۹%۸۱%DB%۸C%D۹%۸۱-/%D۸%A۸%D۹%۸۷-%D۹%۸۵%D۹%۸۶%D۸%A۷%D۸%B۳%D۸%A۸%D۸%AA-(%D۸%AF%D۹%۸۷%D۹%۸۷-%DA%A۹%D۸%B۱%D۸%A۷%D۹%۸۵%D۸%AA



تخفيف به مناسبت ماه رمضان (Contents/۱۲۶۹/%D۸%AA%D۸%AE%D۹%۸۱%DB%۸C%D۹%۸۱-/%D۸%A۸%D۹%۸۷-%D۹%۸۵%D۹%۸۶%D۸%AV%D۸%B۳%D۸%A۸%D۸%AA-(%D۹%۸۵%D۸%AV%D۹%۸۷-%D۸%B۱%D۹%۸۵%D۸%B۶%D۸%AV%D۹%۸۶



تخفیف ویژه دامنه Contents/۱۲۶۱/%D۸%AA%D۸%AE%D۹%۸۱%DB%۸C%D۹%۸۱-/) (%D۹%۸۸%DB%۸C%DA%۹۸%D۹%۸۷-%D۸%AF%D۸%AV%D۹%۸۵%D۹%۸۶%D۹%۸۷



تخفیف هاست، سرور مجازی، سرور اختصاصی و طراحی سایت
 Contents/۱۲۲۴/%D۸%AA%D۸%AE%D۹%۸۱%DB%۸C%D۹%۸۱-/
 %D۹%۸۷%D۸%AV%D۸%B۳%D۸%AA-%D۸%B۳%D۸%B۱%D۹%۸۸%D۸%B۱-
 %D۹%۸۵%D۸%AC%D۸%AV%D۸%B۲%DB%۸C-%D۸%B۳%D۸%B۱%D۹%۸۸%D۸%B۱-
 %D۸%AV%D۸%AE%D۸%AA%D۸%B۵%D۸%AV%D۸%B۵%DB%۸C-%D۹%۸۸-
 (%D۸%B۷%D۸%B۱%D۸%AV%D۸%AD%DB%۸C-%D۸%B۳%D۸%AV%DB%۸C%D۸%AA



Content/۱۲۱۴/%D۸%AF%D۸%AV%D۹%۸۵%D۹%۸۶%D۹%۸۷-/) دامنہ ارزان
(%D۸%AV%D۸%B۱%D۸%B۲%D۸%AV%D۹%۸۶



گواهینامه رایگان ssl
(/Contents/۱۲۰۸/%DA%AF%D۹%۸۸%D۸%AV%D۹%۸۷%DB%۸C%D۹%۸۶%D۸%AV%D۹%۸۵%D۹%۸۷-%D۸%B۱%D۸%AV%DB%۸C%DA%AF%D۸%AV%D۹%۸۶-ssl



تگ های محبوب

هاست (Tag/هاست)

سایت (Tag/سایت)

طراحی سایت (Tag/طراحی-سایت)

آموزش (Tag/آموزش)

آموزش برنامه نویسی (/Tag/آموزش-برنامه-نویسی)


سایت ساز (Tag/سایت-ساز)

برنامه نویسی (Tag/برنامه-نویسی)

(TFS (/Tag/TFS

۰۲۱۵۶۸۸۷۲۶۹ 



 تهران، رباط کریم،
خیابان نیروی انتظامی،
رو به روی انتظام ششم،
پارس راد

پارس راد

(درباره-پارس-راد/1127/Page) درباره ما

(تماس-با-ما/1115/page) تماس با ما

(قوانین-و-مقررات/1163/page) قوانین و مقررات

(جوائز-و-افتخارات/1135/Contents) افتخارات

دسترسی

(104/Contents/Category) مقالات

(105/Contents/Category) اخبار

(106/Contents/Category) مطالب

(107/Contents/Category) خدمات

(109/contents/category) مشتریان

(ir-دامنه-رایگان/1207/Contents) دامنه رایگان ir

(دامنه-ارزان/1214/Contents) دامنه ارزان

مجوزها





دشمن ملی ثبت (رسانه های دیجیتال)
samandehi.ir



(<https://www.instagram.com/parsradco/>)



(<https://t.me/parsradv>)



(<https://facebook.com/parsradco>)