

## آموزش احراز هویت user identity بخش دوم

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



### آموزش احراز هویت user identity بخش دوم

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

در این بخش ادامه بخش اول را خواهیم داشت و می‌خواهیم به جای شناسه و پسورد hardcoded درون کلاس‌ها از دیتابیس استفاده کنیم. خب سریع شروع کنیم :

(۱) نصب identity با کد زیر:

```
PM > Install-Package Microsoft.AspNet.Identity.EntityFramework
```

(۲) در پوشه مد یک کلاس با نام ApplicationUser ایجاد کنید و محتوای آن را به صورت زیر قرار دهید. (می‌خواهیم فقط نام کشور را بیاوریم)

```
public class ApplicationUser : IdentityUser
{
    public string Country { get; set; }
}
```

(۳) در پوشه مدل کلاس ApplicationDbContext را ایجاد میکنیم و محتوای آن را مطابق زیر مقداردهی میکنیم.

```
public class ApplicationDbContext: IdentityDbContext
{
    public ApplicationDbContext()
        : base("DefaultConnection")
    {
    }
}
```

(۴) برای اتصال به پایگاه داده connection string خود را مطابق زیر به web.config اضافه کنید:

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

(۵) به کلاس Startup.cs بروید و آن را به صورت زیر بازنویسی کنید:

```
public class Startup
{
    public static Func< UserManagerFactory > { get; private set; }
    public void Configuration(IApplicationBuilder app)
    {
        app.UseCookieAuthentication(new CookieAuthenticationOptions
        {
            AuthenticationType = "ApplicationCookie",
            LoginPath = new PathString("/Account/Login")
        });

        // configure the user manager
        UserManagerFactory = () =>
        {
            var usermanager = new UserManager(
                new UserStore(new ApplicationDbContext()));
            // allow alphanumeric characters in username
            usermanager.UserValidator = new UserValidator(usermanager)
            {
                AllowOnlyAlphanumericUserNames = false
            };

            return usermanager;
        };
    }
}
```

(۶) به کنترلر Account بروید و تکه کد زیر را به ابتدای آن اضافه کنید:

```
private readonly UserManager userManager;

public AccountController()
: this(Startup.UserManagerFactory.Invoke())
{
}

public AccountController(UserManager userManager)
{
    this.userManager = userManager;
}
```

۷) تکه کد زیر را نیز به انتهای کنترلر Account اضافه میکنیم برای نابود سازی در انتهای request

```
protected override void Dispose(bool disposing)
{
    if (disposing && userManager != null)
    {
        userManager.Dispose();
    }
    base.Dispose(disposing);
}
```

۸) حال باید با استفاده از کد زیر متد login را بازنویسی کنیم:

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

    var user = await userManager.FindAsync(model.Email, model.Password);

    if (user != null)
    {
        var identity = await userManager.CreateIdentityAsync(
            user, DefaultAuthenticationTypes.ApplicationCookie);

        GetAuthenticationManager().SignIn(identity);

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

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

همچنین تابع زیر را به همین کنترلر اضافه میکنیم :

```
private IAuthenticationManager GetAuthenticationManager()
{
    var ctx = Request.GetOwinContext();
    return ctx.Authentication;
}
```

۹) حال میخواهیم عضویت را ایجاد کنیم. ابتدا یک ویو مدل برای آن مینویسیم، به models سپس viewmodels بروید یک کلاس به نام Register ایجاد کنید و محتوای آن را به صورت زیر پر کنید:

```
public class Register
{
    [Required]
    [DataType(DataType.EmailAddress)]
    [DisplayName("شناسه")]
    public string Email { get; set; }

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

    [Required]
    [DisplayName("کشور")]
    public string Country { get; set; }
}
```

۱۰ به کنترلر account رفته و دو اکشن زیر را به آن اضافه کنید:

```
[HttpGet]
public ActionResult Register()
{
    return View();
}

[HttpPost]
public async Task Register(models.ViewModels.Register model)
{
    if (!ModelState.IsValid)
    {
        return View();
    }

    var user = new ApplicationUser
    {
        UserName = model.Email,
        Country = model.Country
    };

    var result = await userManager.CreateAsync(user, model.Password);

    if (result.Succeeded)
    {
        await SignIn(user);
        return RedirectToAction("index", "home");
    }

    foreach (var error in result.Errors)
    {
        ModelState.AddModelError("", error);
    }

    return View();
}
```

۱۱ چون تابع SignIn هم در login و هم register استفاده شده است میتوانیم آن را refactor کرده و در یک تابع جداگانه بنویسیم.

```
private async Task SignIn(ApplicationUser user)
{
    var identity = await userManager.CreateIdentityAsync(
        user, DefaultAuthenticationTypes.ApplicationCookie);
    GetAuthenticationManager().SignIn(identity);
}
```

در مرحله ده از Refactore شده مرحله یازده استفاده شده اما در قسمت ۸ متد refactor ، login نشده پس آن را نیز به صورت زیر بازنویسی میکنیم:

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

    var user = await userManager.FindAsync(model.Email, model.Password);

    if (user != null)
    {
        await SignIn(user);
        return Redirect(GetRedirectUrl(model.ReturnUrl));
    }

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

۱۲) ویو register را ایجاد و محتوای آن را به صورت زیر قرار میدهیم:

```
@model user_identity.Models.ViewModels.Register
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>عضویت</h2>

@Html.ValidationSummary(false)

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

۱۳) حال برای نمایش نام کشور بعد از ورود باید تابع SignIn را بازنویسی کنیم:

```
var identity = await userManager.CreateIdentityAsync(
    user, DefaultAuthenticationTypes.ApplicationCookie);

identity.AddClaim(new Claim(ClaimTypes.Country, user.Country));

GetAuthenticationManager().SignIn(identity);
```

۱۴) کلاس AppUserClaimsIdentityFactory را در پوشه identity ایجاد و آن را به صورت زیر بازنویسی کنید:

```
public class AppUserClaimsIdentityFactory : ClaimsIdentityFactory
{
    public async override Task CreateAsync(UserManager manager,
        ApplicationUser user, string authenticationType)
    {
        var identity = await base.CreateAsync(manager, user, authenticationType);
        identity.AddClaim(new Claim(ClaimTypes.Country, user.Country));

        return identity;
    }
}
```

۱۵) به startup.cs میرویم و کد زیر را قبل از return userManager در تابع (public void Configuration(IApplicationBuilder app) قرار میدهیم:

```
// use out custom claims provider
userManager.ClaimsIdentityFactory = new AppUserClaimsIdentityFactory();
```

۱۶) حالا می‌خواهیم ویژگی‌های کاربر را زیاد کنیم مثلاً الان می‌خواهیم ویژگی سن را برای کاربر اضافه کنیم به ApplicationUser در پوشه Models بروید و آن را به صورت زیر تغییر دهید:

```
public class ApplicationUser : IdentityUser
{
    public string Country { get; set; }
    public int Age { get; set; }
}
```

۱۷) فایل Register.cs را در پوشه ViewModels -> models باز کنید و آن را به صورت زیر تغییر دهید تا ویژگی سن هم برای عضویت به سیستم اضافه کرد:

```
public class Register
{
    [Required]
    [DataType(DataType.EmailAddress)]
    [DisplayName("شناسه")]
    public string Email { get; set; }

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

    [Required]
    [DisplayName("کشور")]
    public string Country { get; set; }

    [Required]
    [DisplayName("سن")]
    public int Age { get; set; }
}
```

۱۸) سپس متد Register در کنترلر Account را ویرایش کنید و جایی که user در حال پر شدن است را تغییر و ویژگی سن را به آن اضافه کنید، پس نتیجه تابع به صورت زیر خواهد بود:

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

    var user = new ApplicationUser
    {
        UserName = model.Email,
        Country = model.Country,
        Age = model.Age,
    };

    var result = await userManager.CreateAsync(user, model.Password);

    if (result.Succeeded)
    {
        await SignIn(user);
        return RedirectToAction("index", "home");
    }

    foreach (var error in result.Errors)
    {
        ModelState.AddModelError("", error);
    }

    return View();
}
```

(۱۹) حالا اگر پروژه را اجرا کنید بعد از ساخت یک user با خطای زیر مواجه میشوید:

The model backing the 'ApplicationDbContext' context has changed since the database was created. Consider using Code First Migrations to update the database (<http://go.microsoft.com/fwlink/?LinkId=238269>).

EF متوجه شده است که کلاس ApplicationUser تغییر یافته است اما تغییرات در دیتابیس اعمال نشده است پس نیاز است با استفاده از migration دیتا بیس را بر اساس مدل خود به روز کنید:

پس در کنسول nugget package manager console کد زیر را وارد کنید تا migration خودکار فعال سازی شود:

```
PM> Enable-Migrations -EnableAutomaticMigrations
```

سپس برای اعمال تغییرات بر روی دیتا بیس کد زیر را وارد کنید:

```
PM> Update-Database
```

کار تمام است!

امیدوارم از این دو مقاله استفاده کرده باشید و آشنایی مناسبی با user identity با استفاده از Asp.net mvc پیدا کرده باشید.

برای دانلود نمونه کد از لینک زیر استفاده کنید. توجه کنید که پکیج ها پاک شده است و هنگام لود کدها باید به اینترنت وصل باشید تا پس از build اول پکیج ها restore شوند.

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

## نظرات

نامنامایمیلایمیلشماره تماس :

موبایل

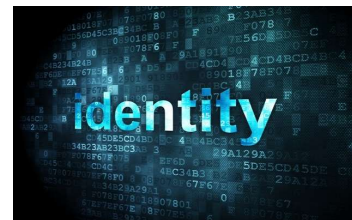
**نظر**

نظر

ارسال

مطالب محبوب

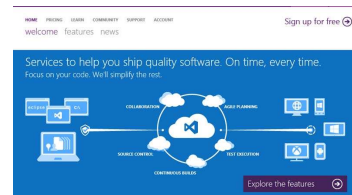
آموزش احراز هویت user identity بخش اول Contents\۱۱۴\%D۸%A۲%D۹%۸۵%D۹%۸۸%D۸%B۲%D۸%B۴-/) آموزش احراز هویت user identity بخش اول  
%D۸%A۷%D۸AD%D۸B۱%D۸A۷%D۸B۲-%D۹%۸۷%D۹%۸۸DB%۸C%D۸AA-user-identity-%D۸%A۸%D۸AE%D۸BF-  
(%D۸A۷%D۹%۸۸%D۹%۸۴



هاسٽ رايگان ويندوز (Contents\۱۱۶۵\%D۹%۸۷%D۸%۸۷%D۸%B۳%D۸%AA-%D۸%B۱%D۸%۸۷%DB%۸C%D۸%AF%D۸%۸۷%D۹%۸۶-/) (%D۹%۸۸%DB%۸C%D۹%۸۶%D۸%AF%D۹%۸۸%D۸%B۲

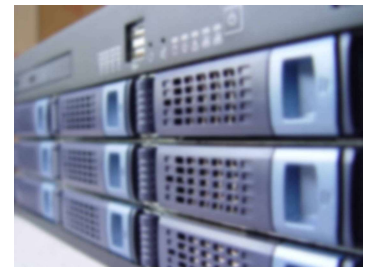
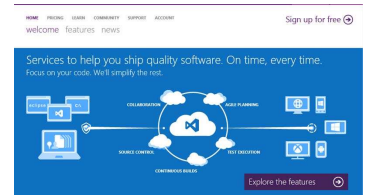


آموزش کار با team foundation server به زبان ساده برای تازه کارها (۱) Contents/۱۱۲۹/%DA%A۳%D۹%۸۵%D۹%۸۸%DA%B۳%DA%BF-/) (۱) %DA%A۹%DA%A۷%DA%B۱-%DA%A۸%AV-team-foundation-server-%DA%A۸%D۹%۸۷-%DA%B۳%DA%A۸%DA%A۷%D۹%۸۶-%DA%B۳%DA%A۷%DA%AF%D۹%۸۷-%DA%A۸%DA%B۱%DA%A۷%DB%۸C-%DA%AA%DA%A۷%DA%B۳%D۹%۸۷-(%DA%A۹%DA%A۷%DA%B۱%D۹%۸۷%DA%A۷-



آموزش کار با team foundation server به زبان ساده برای تازه کارها (۲) Contents/11۳۰/%DA%A۲%D۹%۸۵%D۹%۸۸%D۸%B۲%D۸%B۴-%۲) %DA%A۹%D۸%AV%D۸%B1-%D۸%A۸%D۸%AV-team-foundation-server-%DA%A۸%D۹%AV-%D۸%B۲%D۸%۸۸%D۸%AV%D۹%۸۶-%D۸%B۳%D۸%AV%D۸%AF%D۹%۸۷-%D۸%A۸%D۸%B1%D۸%AV%DB۸C-%D۸%AA%D۸%AV%D۸%B۲%D۹%۸۷-%D۸%A۹%D۸%AV%D۸%B1%D۹%۸۷%D۸%AV-%۲





آموزش برنامه نویسی (Contents/۱۱۲۴/%D۹%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/)

دامنه رایگان -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

تخفیف به مناسبت هفته وحدت (Contents/۱۲۷۷/%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-%D۹%۸۷%D۹%۸۱%D۸%AA%D۹%۸۷-%D۹%۸۸%D۸%AD%D۸%AF%D۸%AA



تخفیف 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-%D۸%A۹%D۸%B۱%D۸%A۷%D۹%۸۵%D۸%AA



تخفیف 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%A۷%D۸%B۳%D۸%A۸%D۸%AA-%D۹%۸۵%D۸%A۷%D۹%۸۷-%D۸%B۱%D۹%۸۵%D۸%B۶%D۸%A۷%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%A۷%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%A۷%D۸%B۳%D۸%AA-%D۸%B۳%D۸%B۱%D۹%۸۸%D۸%B۱-%D۹%۸۵%D۸%AC%D۸%A۷%D۸%B۲%DB%۸C-  
%D۸%B۳%D۸%B۱%D۹%۸۸%D۸%B۱-%D۸%A۷%D۸%AE%D۸%AA%D۸%B۵%D۸%A۷%D۸%B۵%DB%۸C-%D۹%۸۸-  
(%D۸%B۷%D۸%B۱%D۸%A۷%D۸%AD%DB%۸C-%D۸%B۳%D۸%A۷%DB%۸C%D۸%AA



## تگ های محبوب

طراحی سایت (Tag/طراحی-سایت)

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



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

پارس راد

(جوائز و-افتخارات/۱۱۳۵/Contents/) افتخارات

## دسترسی

(دامنه-ارزان/Contents/۱۲۱۴/ دامنه ارزان)

## مجوزها

