

هنگامی که یک پروژه جدید ASP.NET را در VS 2013 می‌سازید و متد احراز هویت آن را Individual User Accounts انتخاب می‌کنید، قالب پروژه، امکانات لازم را برای استفاده از تامین کنندگان ثالث، فراهم می‌کند، مثلاً مایکروسافت، گوگل، توییتر و فیسبوک. هنگامی که توسط یکی از این تامین کنندگان کاربری را احراز هویت کردید، می‌توانید اطلاعات بیشتری درخواست کنید. مثلاً عکس پروفایل کاربر یا لیست دوستان او. سپس اگر کاربر به اپلیکیشن شما سطح دسترسی کافی داده باشد می‌توانید این اطلاعات را دریافت کنید و تجربه کاربری قوی‌تر و بهتری ارائه کنید. در این پست خواهید دید که چطور می‌شود از تامین کننده Facebook اطلاعات بیشتری درخواست کرد. پیش فرض این پست بر این است که شما با احراز هویت فیسبوک و سیستم کلی تامین کننده‌ها آشنایی دارید. برای اطلاعات بیشتر درباره راه اندازی احراز هویت فیسبوک به [این لینک](#) مراجعه کنید. برای دریافت اطلاعات بیشتر از فیسبوک مراحل زیر را دنبال کنید. یک اپلیکیشن جدید ASP.NET MVC با تنظیمات Individual User Accounts بسازید. احراز هویت فیسبوک را توسط کلید هایی که از Facebook دریافت کرده اید فعال کنید. برای اطلاعات بیشتر در این باره می‌توانید به [این لینک](#) مراجعه کنید.

برای درخواست اطلاعات بیشتر از فیسبوک، فایل Startup.Auth.cs را مطابق لیست زیر ویرایش کنید.

```
List<string> scope = newList<string>() { "email", "user_about_me", "user_hometown", "friends_about_me",
"friends_photos" };
var x = newFacebookAuthenticationOptions();
x.Scope.Add("email");
x.Scope.Add("friends_about_me");
x.Scope.Add("friends_photos");
x.AppId = "636919159681109";
x.AppSecret = "f3c16511fe95e854cf5885c10f83f26f";
x.Provider = newFacebookAuthenticationProvider()
{
    OnAuthenticated = async context =>
    {
        //Get the access token from FB and store it in the database and
        //use FacebookC# SDK to get more information about the user
        context.Identity.AddClaim(
            new System.Security.Claims.Claim("FacebookAccessToken",
            context.AccessToken));
    }
};
x.SignInAsAuthenticationType = DefaultAuthenticationTypes.ExternalCookie;
app.UseFacebookAuthentication(x);
```

در خط 1 مشخص می‌کنیم که چه scope هایی از داده را می‌خواهیم درخواست کنیم. از خط 10 تا 17 رویداد OnAuthenticated را مدیریت می‌کنیم که از طرف Facebook OWIN authentication اجرا می‌شود. این متد هر بار که کاربری با فیسبوک خودش را احراز هویت می‌کند فراخوانی می‌شود. پس از آنکه کاربر احراز هویت شد و به اپلیکیشن سطح دسترسی لازم را اعطا کرد، تمام داده‌ها در FacebookContext ذخیره می‌شوند. خط 14 شناسه FacebookAccessToken را ذخیره می‌کند. ما این آبجکت را از فیسبوک دریافت کرده و از آن برای دریافت لیست دوستان کاربر استفاده می‌کنیم. نکته: در این مثال تمام داده‌ها بصورت Claims ذخیره می‌شوند، اما اگر بخواهید می‌توانید از ASP.NET Identity برای ذخیره آنها در دیتابیس استفاده کنید. در قدم بعدی لیست دوستان کاربر را از فیسبوک درخواست می‌کنیم. ابتدا فایل Views/Shared/_LoginPartial.cshtml را باز کنید و لینک زیر را به آن بیافزایید.

```
<li>
    @Html.ActionLink("FacebookInfo", "FacebookInfo", "Account")
</li>
```

هنگامی که کاربری وارد سایت می‌شود و این لینک را کلیک می‌کند، ما لیست دوستان او را از فیسبوک درخواست می‌کنیم و به‌مراه عکس‌های پروفایل شان آنها را لیست می‌کنیم.

تمام Claim ها را از **UserIdentity** بگیرید و آنها را در دیتابیس ذخیره کنید. در این قطعه کد ما تمام Claim هایی که توسط OWIN دریافت کرده ایم را می‌خوانیم، و شناسه FacebookAccessToken را در دیتابیس عضویت ASP.NET Identity ذخیره می‌کنیم.

```
//
// GET: /Account/LinkLoginCallback
public async Task<ActionResult> LinkLoginCallback()
{
    var loginInfo = await AuthenticationManager.GetExternalLoginInfoAsync(XsrfKey,
User.Identity.GetUserId());
    if (loginInfo == null)
    {
        return RedirectToAction("Manage", new { Message = ManageMessageId.Error });
    }
    var result = await UserManager.AddLoginAsync(User.Identity.GetUserId(), loginInfo.Login);
    if (result.Succeeded)
    {
        var currentUser = await UserManager.FindByIdAsync(User.Identity.GetUserId());
        //Add the Facebook Claim
        await StoreFacebookAuthToken(currentUser);
        return RedirectToAction("Manage");
    }
    return RedirectToAction("Manage", new { Message = ManageMessageId.Error });
}
```

خط 14-15 شناسه FacebookAccessToken را در دیتابیس ذخیره می‌کند.

StoreFacebookAuthToken تمام اختیارات (claim) های کاربر را از UserIdentity می‌گیرد و Access Token را در قالب یک User Claim در دیتابیس ذخیره می‌کند. اکشن LinkLoginCallback هنگامی فراخوانی می‌شود که کاربر وارد سایت شده و یک تامین کننده دیگر را می‌خواهد تنظیم کند.

اکشن ExternalLoginConfirmation هنگام اولین ورود شما توسط تامین کنندگان اجتماعی مانند فیسبوک فراخوانی می‌شود. در خط 26 پس از آنکه کاربر ایجاد شد ما یک FacebookAccessToken را بعنوان یک Claim برای کاربر ذخیره می‌کنیم.

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> ExternalLoginConfirmation(ExternalLoginConfirmationViewModel model,
string returnUrl)
{
    if (User.Identity.IsAuthenticated)
    {
        return RedirectToAction("Manage");
    }

    if (ModelState.IsValid)
    {
        // Get the information about the user from the external login provider
        var info = await AuthenticationManager.GetExternalLoginInfoAsync();
        if (info == null)
        {
            return View("ExternalLoginFailure");
        }
        var user = new ApplicationUser() { UserName = model.Email };
        var result = await UserManager.CreateAsync(user);
        if (result.Succeeded)
        {
            result = await UserManager.AddLoginAsync(user.Id, info.Login);
            if (result.Succeeded)
            {
                await StoreFacebookAuthToken(user);
                await SignInAsync(user, isPersistent: false);
                return RedirectToLocal(returnUrl);
            }
        }
        AddErrors(result);
    }

    ViewBag.ReturnUrl = returnUrl;
```

```
    return View(model);
}
```

اکشن **ExternalLoginCallback** هنگامی فراخوانی می‌شود که شما برای اولین بار یک کاربر را به یک تامین کننده اجتماعی اختصاص می‌دهید. در خط 17 شناسه دسترسی فیسبوک را بصورت یک claim برای کاربر ذخیره می‌کنیم.

```
//
// GET: /Account/ExternalLoginCallback
[AllowAnonymous]
public async Task<ActionResult> ExternalLoginCallback(string returnUrl)
{
    var loginInfo = await AuthenticationManager.GetExternalLoginInfoAsync();
    if (loginInfo == null)
    {
        return RedirectToAction("Login");
    }

    // Sign in the user with this external login provider if the user already has a login
    var user = await UserManager.FindAsync(loginInfo.Login);
    if (user != null)
    {
        //Save the FacebookToken in the database if not already there
        await StoreFacebookAuthToken(user);
        await SignInAsync(user, isPersistent: false);
        return RedirectToLocal(returnUrl);
    }
    else
    {
        // If the user does not have an account, then prompt the user to create an account
        ViewBag.ReturnUrl = returnUrl;
        ViewBag.LoginProvider = loginInfo.Login.LoginProvider;
        return View("ExternalLoginConfirmation", new ExternalLoginConfirmationViewModel { Email
= loginInfo.Email });
    }
}
```

در آخر شناسه FacebookAccessToken را در دیتابیس ASP.NET Identity ذخیره کنید.

```
private async Task StoreFacebookAuthToken(ApplicationUser user)
{
    var claimsIdentity = await
AuthenticationManager.GetExternalIdentityAsync(DefaultAuthenticationTypes.ExternalCookie);
    if (claimsIdentity != null)
    {
        // Retrieve the existing claims for the user and add the FacebookAccessTokenClaim
        var currentClaims = await UserManager.GetClaimsAsync(user.Id);
        var facebookAccessToken = claimsIdentity.FindAll("FacebookAccessToken").First();
        if (currentClaims.Count() <= 0 )
        {
            await UserManager.AddClaimAsync(user.Id, facebookAccessToken);
        }
    }
}
```

پکیج Facebook C# SDK را نصب کنید. <http://nuget.org/packages/Facebook>

فایل AccountViewModel.cs را باز کنید و کد زیر را اضافه کنید.

```
public class FacebookViewModel
{
    [Required]
    [Display(Name = "Friend's name")]
    public string Name { get; set; }

    public string ImageURL { get; set; }
}
```

کد زیر را به کنترلر Account اضافه کنید تا عکس‌های دوستان تان را دریافت کنید.

```
//GET: Account/FacebookInfo
[Authorize]
public async Task<ActionResult> FacebookInfo()
{
    var claimsforUser = await UserManager.GetClaimsAsync(User.Identity.GetUserId());
    var access_token = claimsforUser.FirstOrDefault(x => x.Type == "FacebookAccessToken").Value;
    var fb = new FacebookClient(access_token);
    dynamic myInfo = fb.Get("/me/friends");
    var friendsList = newList<FacebookViewModel>();
    foreach (dynamic friend in myInfo.data)
    {
        friendsList.Add(new FacebookViewModel()
        {
            Name = friend.name,
            ImageURL = @"https://graph.facebook.com/" + friend.id + "/picture?type=large"
        });
    }
    return View(friendsList);
}
```

در پوشه Views/Account یک نمای جدید با نام FacebookInfo.cshtml بسازید و کد Markup آن را مطابق لیست زیر تغییر دهید.

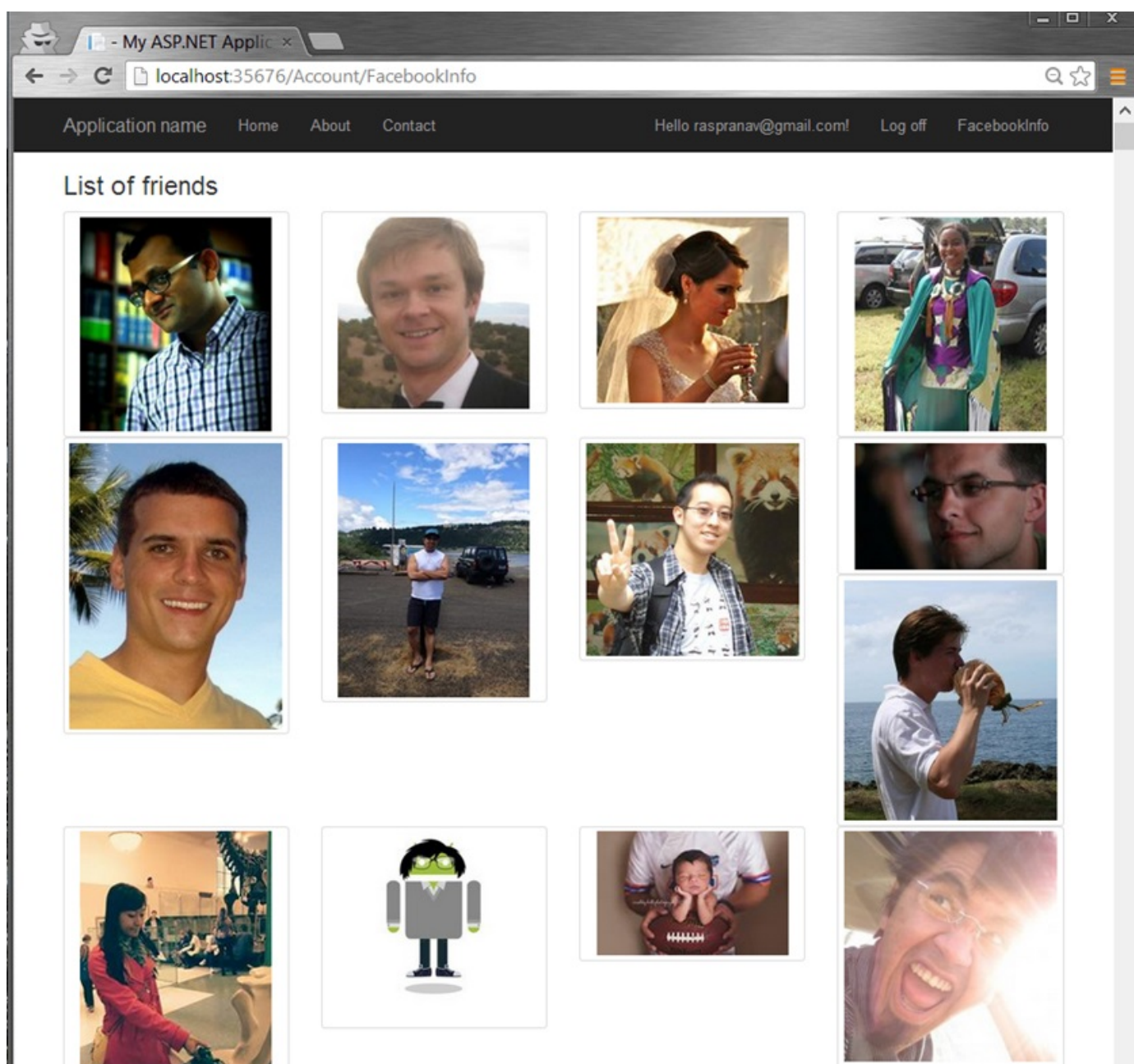
```
@model IList<WebApplication96.Models.FacebookViewModel>
@if (Model.Count > 0)
{
    <h3>List of friends</h3>
    <div class="row">
        @foreach (var friend in Model)
        {
            <div class="col-md-3">
                <a href="#" class="thumbnail">
                    <img src=@friend.ImageURL alt=@friend.Name />
                </a>
            </div>
        }
    </div>
}
```

در این مرحله، شما می‌توانید لیست دوستان خود را به همراه عکس‌های پروفایل شان دریافت کنید.

پروژه را اجرا کنید و توسط Facebook وارد سایت شوید. باید به سایت فیسبوک هدایت شوید تا احراز هویت کنید و دسترسی

لازم را به اپلیکیشن اعطا کنید. پس از آن مجدداً به سایت خودتان باید هدایت شوید.

حال هنگامی که روی لینک FacebookInfo کلیک می‌کنید باید صفحه ای مشابه تصویر زیر ببینید.



این یک مثال ساده از کار کردن با تامین کنندگان اجتماعی بود. همانطور که مشاهده می‌کنید، راحتی می‌توانید داده‌های بیشتری برای کاربر جاری درخواست کنید و تجربه کاربری و امکانات بسیار بهتری را در اپلیکیشن خود فراهم کنید.