

در پست‌های قبلی [OpenID چیست؟](#)

[استفاده از OpenID در وب سایت جهت احراز هویت کاربران با مفاهیم OpenID تاحدودی آشنا شدیم](#)

در این پست می‌خواهیم با یک مثال ساده نشان دهیم که سایت ما چگونه با استفاده از OpenID Authentication می‌تواند از اکانت گوگل استفاده کرده و کاربر در وب سایت ما شناسایی شود.

برای اینکار ابتدا

1- کتابخانه DotNetOpenAuth را از طریق [NuGet](#) به لیست رفرنس‌های پروژه وب خود اضافه نمایید

2- یک صفحه بنام Login.aspx یا هر نام دلخواهی را به پروژه خود اضافه نمایید. در نهایت کد Html صفحه شما به ید به صورت زیر باشد.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Login.aspx.cs" Inherits="OAuthWebTest.Login" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div id="loginform">
<div id="NotLoggedIn" runat="server">
<asp:ImageButton ID="ButtonLoginWithGoogle" runat="server" ImageUrl="Google.JPG"
OnCommand="ButtonLoginWithGoogle_Click"
CommandArgument="https://www.google.com/accounts/o8/id" />
<p />
<asp:Label runat="server" ID="LabelMessage" />
</div>
</div>
</form>
</body>
</html>
```

در کد Html بالا به خصوصیت CommandArgument از کنترل ImageButton دقت نمایید که دارای مقدار

"https://www.google.com/accounts/o8/id" می‌باشد. در واقع این آدرس باید برای اکانت گوگل جهت احراز هویت توسط OpenID

استفاده شود. (آدرس API گوگل برای استفاده از این سرویس)

3- در قسمت کد نویسی صفحه کدهای زیر را وارد نمایید.

```
using System;
using System.Web.UI;
using System.Web.UI.WebControls;
using DotNetOpenAuth.OpenId.RelyingParty;

namespace OAuthWebTest
{
    public partial class Login : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            OpenIdRelyingParty openIdRelyingParty = new OpenIdRelyingParty();
            var response = openIdRelyingParty.GetResponse();
            if (response == null) return;
            switch (response.Status)
            {
                case AuthenticationStatus.Authenticated:
                    NotLoggedIn.Visible = false;
                    Session["GoogleIdentifier"] = response.ClaimedIdentifier.ToString();
                    Response.Redirect("Default.aspx");
                    break;
                case AuthenticationStatus.Canceled:
                    LabelMessage.Text = "Cancelled.";
                    break;
            }
        }
    }
}
```

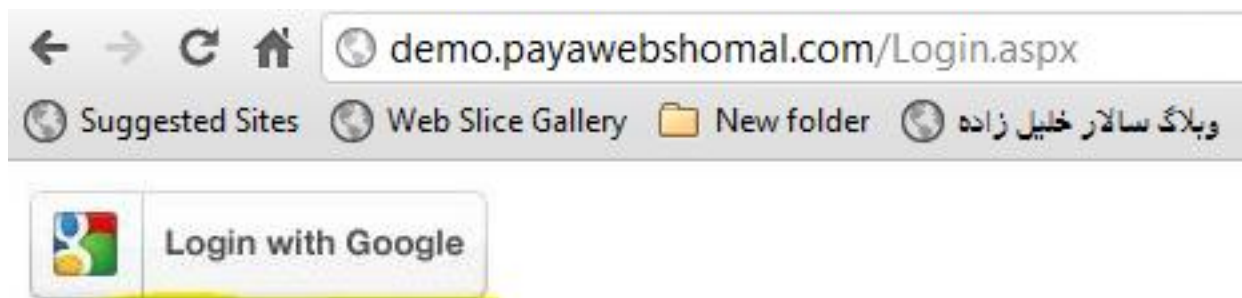
```

        case AuthenticationStatus.Failed:
            LabelMessage.Text = "Login Failed.";
            break;
    }
}

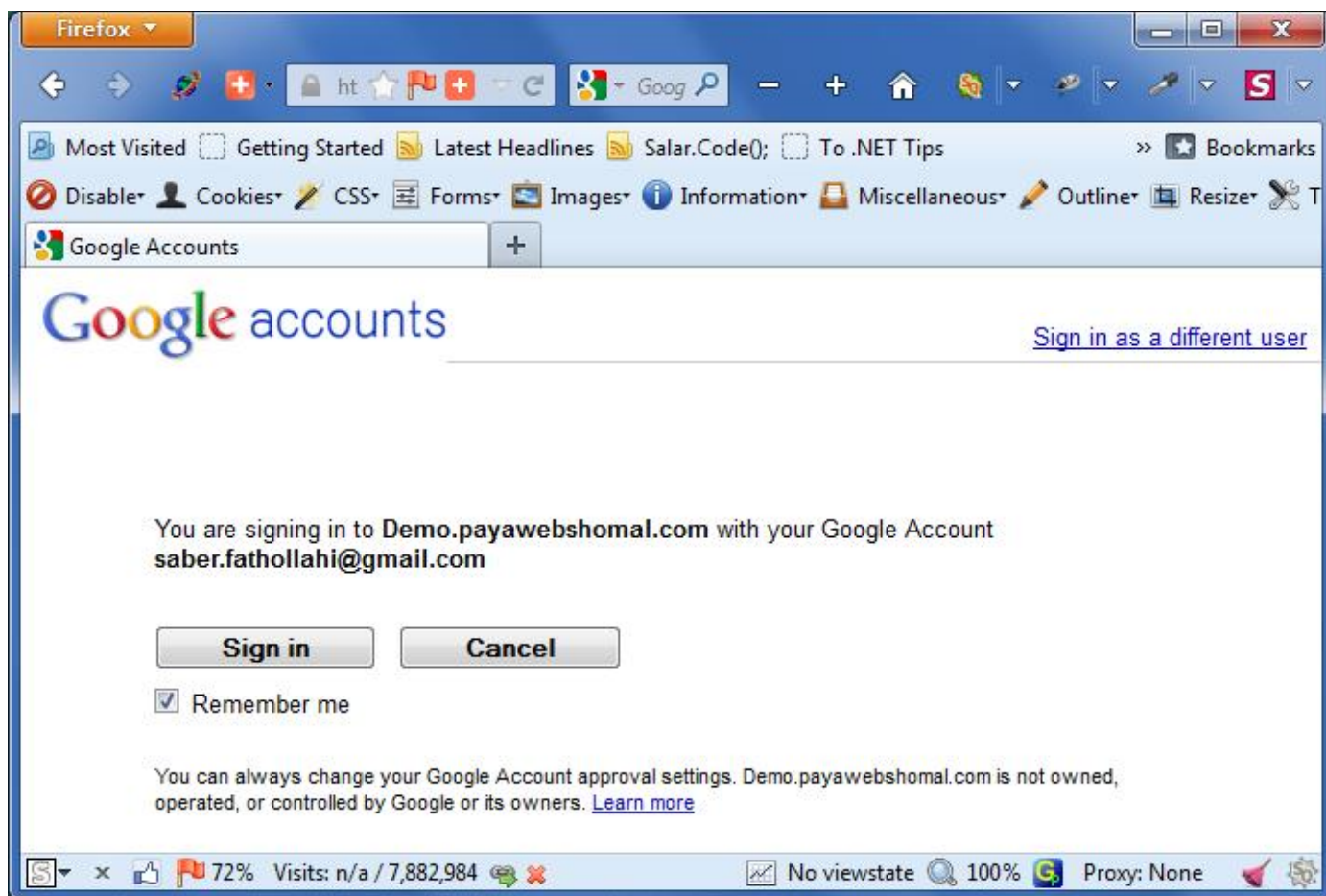
protected void ButtonLoginWithGoogle_Click(object src, CommandEventArgs e)
{
    string discoveryUri = e.CommandArgument.ToString();
    OpenIdRelyingParty openId = new OpenIdRelyingParty();
    var returnUrl = new UriBuilder(Request.Url) { Query = "" };
    var request = openId.CreateRequest(discoveryUri, returnUrl.Uri, returnUrl.Uri);
    request.RedirectToProvider();
}
}
}

```

همانگونه که مشاهده می‌کنید در رویداد کلیک دکمه لوگین ButtonLoginWithGoogle_Click ابتدا آدرس یکتا گوگل (مقدار CommandArgument اختصاص داده شده به کنترل ImageButton) به همراه آدرس صفحه جاری وب سایت توسط متد CreateRequest از شی openId ترکیب شده و یک درخواست (Request) ساخته شده و در نهایت برای سرویس دهنده گوگل ارسال می‌شود. با اجرای پروژه با تصویر زیر روبرو می‌شوید بروی کلید لوگین کلیک نمایید.



در واقع با کلیک روی دکمه مورد نظر تکه کدی که در بالا شرح داده شد اجرا شده و ما را به صفحه ای شبیه تصویر پایین هدایت می‌کند



در صورتی که کلید Sign In انتخاب شود شما به سایت (همین برنامه) اجازه داده اید که از اطلاعات حساب کاربری شما استفاده کند. پس از کلیک به برنامه اصلی (طبق آدرس بازگشت تعیین شده در رویداد `ButtonLoginWithGoogle_Click`) در رویداد `PageLoad` صفحه لاگین اطلاعات بازگشتی از سایت سرویس دهنده (در اینجا گوگل) چک می‌شود و با توجه به پاسخ مورد نظر عملیاتی انجام می‌شود، مثلاً در صورتی که شما تایید کرده باشید اطلاعات شناسایی شما توسط گوگل در کلیدی به نام `GoogleIdentifier` در سشن ذخیره شده و شما به صفحه اصلی سایت هدایت می‌شوید.

دموی پروژه در این [آدرس](#) قرار داده شده است.

سورس پروژه از این [آدرس](#) قابل دسترسی است.

توجه: در این [آدرس](#) شرح داده شده که چگونه دسترسی سایت‌های دیگر به اکانت گوگل خود را قطع کنید

در پست‌های آینده اتصال به تویتر و فیس‌بوک و سایت‌های دیگر توضیح داده خواهد شد.

نظرات خوانندگان

نویسنده: مجتبی چنانی
تاریخ: ۱۳۹۱/۰۵/۰۴ ۹:۰۰

با سلام و تشکر از پست خوبتون

در زمانی که ما از سیستم‌های ورود و ثبت کاربران شرکت‌های دیگر استفاده می‌کنیم آیا می‌توانیم لاگ گرفته یا اینکه برای خودمان یک صفحه داشته باشیم تا ورود و خروج‌های اکانت‌های درون وبسایتمان را بررسی کنیم یا اینکه خودمان باید این بخش را کدنویسی کنیم؟

یک سؤال دیگر این است که زمانی که از openid های شرکت‌های دیگر استفاده می‌کنیم فقط احراز هویت را از این سرویس‌ها دریافت می‌کنیم یا اینکه در همه صفحات و دیگر کارهای کاربر نظارت به صورت خودکار انجام می‌شود یا اینکه باز هم باید کدنویسی کنیم؟

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۴:۰۹

در صورتی که بخواهید لاگ بندازید باید خود کد نویسی کرده و در دیتابیس ذخیره کنید
بله می‌توان در هر صفحه با استفاده از کلیدی که در سشن کاربر ذخیره کرده اید (با توجه به اینکه سشن کاربران جداست) می‌توانید وجود کاربر را چک کنید

نویسنده: سینا علیزاده
تاریخ: ۱۳۹۲/۰۹/۰۳ ۱۳:۲۹

آقا عالی بود ممنون
میشه برای facebook و توییتر هم بزارید ، واقعا بهش نیاز دارم ممنون میشم
کارتون عالیه موفق باشید

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۰۳ ۱۳:۳۵

نگاهی هم به پروژه [DotNetAuth](#) داشته باشید.

نویسنده: دادخواه
تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۱:۳۱

سلام.

این سرویس چطوری برای اکانت‌های یاهو کار می‌کنه؟