



Quality is our Mantra

Q
X

10 Implementing Google account authentication in ASP.NET MVC

Aug Naveena Kunder | .NET , MVC , C# | 0 Comments | Return

Implementation of Google OAuth in MVC application:

Step 1: Create a Google OAuth application

To create it please follow the steps below:

- Go to <https://console.developers.google.com> , login with your Gmail id.
- Click on 'Select a Project' dropdown at left top of the page.
- Click on 'Create project' button,
- Enter project name and create project.
- Click Credentials on the left navigation.
- Click Create Credentials > OAuth ClientID, fill the form and submit.
- Client ID and Client Secret will be created on successful creation of application.
- This will be used in Authorization process.

Step 2: Configuring your Google application

- Login to <https://console.developers.google.com>
- Select your created application and click on edit.
- Enter the 'Authorized redirect URL'.
- Click on 'Save button'.

Now your application created and configured.

Step 3: Creating visual studio application

- Create empty Asp.Net MVC application and add controller.
- In 'Home View' create one action link like below
 - `@Html.ActionLink("Login Using Google", "")`
- In action method add redirect url, there user redirect to Google login URL,
 - URL: `https://accounts.google.com/o/oauth2/auth?response_type=code&redirect_uri=&scope=https://www.googleapis.com/auth/userinfo.email%20https://www.googleapis.com/auth/userinfo.pro`
 - `response_type` : The value of this field should always be: code
 - `redirect_uri` : The URI your users will be sent back to after authorization. This value must match one of the defined OAuth 2.0 Redirect URLs in your application configuration.

- Scope : A URL-encoded, space delimited list of member permissions your application is requesting on behalf of the user. If you do not specify a scope in your call, we will fall back to using the default member permissions you defined in your application configuration.
- client_id : The "ClientID" value generated when you registered your application.
- Add controller and create callback function and add below code
 - Here we need to collect the Authorization code and Access token.
 - Sample code:

```

try
{
    var url = Request.Url.Query;
    if (url != "")
    {
        string queryString = url.ToString();
        char[] delimiterChars = { '=' };
        string[] words = queryString.Split(delimiterChars);
        string code = words[1];

        if (code != null)
        {
            //get the access token
            HttpWebRequest webRequest =
(HttpWebRequest)WebRequest.Create("https://accounts.google.com/o/oauth2/token");
            webRequest.Method = "POST";
            Parameters = "code=" + code + "&client_id=" + client_id + "&client_secret=" + client_secret +
"&redirect_uri=" + redirect_url + "&grant_type=authorization_code";
            byte[] byteArray = Encoding.UTF8.GetBytes(Parameters);
            webRequest.ContentType = "application/x-www-form-urlencoded";
            webRequest.ContentLength = byteArray.Length;
            Stream postStream = webRequest.GetRequestStream();
            // Add the post data to the web request
            postStream.Write(byteArray, 0, byteArray.Length);
            postStream.Close();
            WebResponse response = webRequest.GetResponse();
            postStream = response.GetResponseStream();
            StreamReader reader = new StreamReader(postStream);
            string responseFromServer = reader.ReadToEnd();
            GoogleAccessToken serStatus = JsonConvert.DeserializeObject<GoogleAccessToken>
(responseFromServer);

            if (serStatus != null)
            {
                string accessToken = string.Empty;
                accessToken = serStatus.access_token;
                Session["Token"] = accessToken;
                if (!string.IsNullOrEmpty(accessToken))
                {
                    //call get user information function with access token as parameter
                }
            }
        }
    }
}

```

```

    }
    }
    }
}
catch (Exception ex)
{
    return RedirectToAction("Index", "Home");
}
}

```

- To get user information add below function

```

try
{
    HttpClient client = new HttpClient();
    var urlProfile = "https://www.googleapis.com/oauth2/v1/userinfo?access_token=" + access_token;
    client.CancelPendingRequests();
    HttpResponseMessage output = client.GetAsync(urlProfile).Result;
    if (output.IsSuccessStatusCode)
    {
        string outputData = output.Content.ReadAsStringAsync().Result;
        serStatus = JsonConvert.DeserializeObject<GoogleUserOutputData>(outputData);
    }
}
catch (Exception ex)
{
    //catching the exception
}
return View(serStatus);

```

Note: we need to create following model to Deserialize the json into object:

```

public class GoogleAccessToken
{
    public string access_token { get; set; }
    public string token_type { get; set; }
    public int expires_in { get; set; }
    public string id_token { get; set; }
    public string refresh_token { get; set; }
}

public class GoogleUserOutputData
{
    public string id { get; set; }
    public string name { get; set; }
    public string given_name { get; set; }
    public string email { get; set; }
    public string picture { get; set; }
}

```

- Finally, we will add LogOff action to logoff user.

```

public ActionResult LogOff()
{

```

```
//Logout from application
FormsAuthentication.SignOut();
return Redirect(Url.Action("Index", "Home"));

//Logout from google

return Redirect("https://www.google.com/accounts/Logout?continue=https://appengine.google.com/_ah/logout?continue=");

}
```

Conclusion:

We have discussed how to implement google oauth2 to secure our web application. The main advantage of google oauth2 is user no need to remember all of his/her account details, user can login using google credentials.

Related

Building a RESTful API using python and Django

Load testing using Visual Studio – 2013

Swift Mapping and Flat Mapping

Webinar -The curious of native versus the cross platform development mobile apps

What you Need? Python 2.6+
Django 1.5+ django-tastypie
Sqlite3 Django is Free and Open Source ...

suma.chandrashekar

In this blog post, I will introduce some of useful tips and techniques of using swift mapping and flat mapping

The curious case of native vs the cross platform development mobile apps Business is often

[Read More >](#)

[Read More >](#)

[Read More >](#)

[Read More >](#)

Share

Like 0

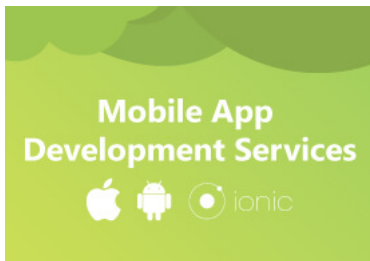
Tweet

Share

Post a Comment

Name*	Email*	WebSite
Comment* <div></div>		

Submit



Recent

Popular

How to pass Images from Report Body to Header/Footer in RDLC

Sandeep Suresh Vernekar 2/6/2019 3:35:00 PM

When using SetData and GetData functions to pass data from report body to report header/footer we can also pass the images

⚡ Angular — Protect your unsaved data

Pranesh Ganapathi 12/27/2018 11:26:00 AM

using CanDeactivate class route guard to prevent users from navigating with unsaved entries.

Introducing Azure DevOps Earlier (VSTS)

Nagaraj B Bhairaji 10/22/2018 12:48:00 PM

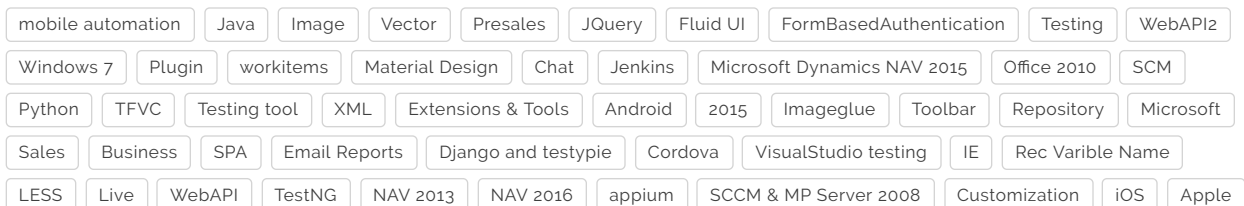
Microsoft recently announced Azure DevOps and an evolution of Visual Studio Team Services (VSTS) which is now called as Azure DevOps Services. All the VSTS users will be able to move or change to new ...

Create a simple Chatbot App using Botframework & Xamarin.Forms

Darshan S 8/23/2018 10:16:00 AM

We can integrate Chatbot API with Xamarin.Forms App using two methods, Embed Code with Web View or Native with DirectLine. In this blog, I shall explain how to integrate Chatbot using embed code with W...

Tags



Monthly Archive

February 2019 (1)

December 2018 (1)

October 2018 (1)

August 2018 (1)

Subscribe

Name

Email*

Subscribe

Home

 Corporate

 Services

 Software Services

 Mobility Solutions

 Microsoft Dynamics ERP & CRM

Products

 FlowWright

 e-Conformance


 Telemetry

 Mobile Apps

 Payroll-Addon

Services

 Software Development Solutions

 Staffing Solutions

 Become a Partner?

 Feedback

 Company Brochure

Contact Us

Name:

*

Email:

*

Message: